LIGHT AIRCRAFT LIFT, DRAG, AND MOMENT PREDICTION - A REVIEW AND ANALYSIS

Frederick O. Smetana, Delbert C. Summey,
Neill S. Smith, and Ronald K. Carden

1. Report No. NASA CR-2523	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle LIGHT AIRCRAFT LIFT, DRAG, AND MOMENT PREDICTION - A REVIEW AND ANALYSIS		5. Report Date May 1975
		6. Performing Organization Code
7. Author(s) Frederick O. Smetana, Delbert C. Summey, Neill S. Smith, and Ronald K. Carden		8. Performing Organization Report No.
		10. Work Unit No. 760-60-01-0900
9. Performing Organization Name and Address North Carolina State University Raleigh, North Carolina 27607		11. Contract or Grant No. NGR 34-002-179
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Topical report.

16. Abstract

The historical development of analytical methods for predicting the lift, drag, and pitching moment of complete light aircraft configurations in cruising flight is reviewed. Theoretical methods, based in part on techniques described in the literature and in part on original work, are then developed in detail. These methods form the basis for understanding the computer programs presented in the remainder of the work. Programs are given to: (1) compute the lift, drag, and moment of conventional airfoils, (2) extend these two-dimensional characteristics to three dimensions for moderate-to-high aspect ratio unswept wings, (3) plot complete configurations, (4) convert the fuselage geometric data to the correct input format, (5) compute the fuselage lift and drag, (6) compute the lift and moment of symmetrical airfoils to M = 1.0 by a simplified semi-empirical procedure, and (7) compute, in closed form, the pressure distribution over a prolate spheroid at $\alpha = 0$. Comparisons of the predictions with experiment indicate excellent lift and drag agreement for conventional airfoils and wings for $C_L < 0.8$. Limited comparisons of body-alone drag characteristics yield reasonable agreement. Also included are discussions for interference effects and techniques for summing the results of 1-5 above to obtain predictions for complete configurations.

17. Key Words (Suggested by Author(s)) Aerodynamic characteristics, prediction techniques, lift estimation, drag estimation, computer programs, numerical methods	18. Distribution Statement Unclassified - Unrestricted New Subject Category 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 492	22. Price* $12.00

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued) - Appendices

LIST OF TABLES

GENERAL INTRODUCTION

The successful prediction of the performance of a new or modified aircraft depends as much on the availability of an accurate estimate of the configuration's lift and drag characteristics as on any one thing. Despite the importance of this task, the procedure used in the light aircraft industry and that taught in most universities has remained essentially a semi-empirical correlation of wind tunnel and flight test data plus a collection of useful rules of thumb. The major airframe manufacturers and their cognizant governmental laboratories have for some time sought both to reduce the time needed to develop these predictions and to increase their accuracy and reliability through the use of large-scale digital computers. Employing long-known, highly rigorous analytical computation methods which become too involved when applied to complete aircraft for one to perform manually, these groups have, within the last three-to-six years, achieved some remarkable successes in predicting the aerodynamic characteristics of complex geometric shapes.

It is the intention of the present work

- to review analytical and experimental developments in aerodynamics of the past 32 years, in particular those of the National Aeronautics and Space Administration,

- to identify those of special pertinance to the design of light aircraft and

- to develop from these easy-to-use design procedures.

Of necessity these procedures will involve digital computer programs. This approach follows that employed in earlier works in this series. Reference 1, for example, provides detailed computer programs for the prediction of point and path performance assuming that the lift, drag, and thrust characteristics are known. References 2 and 3 give programs for the calculation of stability derivatives and aircraft motions given the vehicle's geometric and inertial characteristics. Thus with these and the present work the reader can specify the aircraft geometry, mass distribution, and thrust applied to the air and expect to obtain the vehicle's performance and its handling qualities. He can then vary the geometry, etc. in a systematic fashion and find the shape giving the most satisfactory combination of performance and handling qualities.

While the availability of these programs will certainly be of great assistance in the overall design task, it should be noted that many areas of aircraft configuration design have not been treated in detail or have not been programmed for computer solution in the work to date. These include large excursions in the motions about an equilibrium position, performance in the horizontal plane, takeoff and landing, aerodynamic characteristics at high angles of attack and/or with deflected flaps, flight in turbulent air, calculation of stick and rudder forces and deflections, propeller slipstream effects, adequate representations of thrust horsepower and fuel

flow, and the effects of specific stability augmentation systems. It is the authors' ultimate intention to treat all of these problems in the manner of the programs included in the present work. They would, however, be pleased to receive suggestions from readers and users of the work as to the priority with which the problems should be attacked.

The present work departs from the practice of previous works in this series in that the computer programs presented are usually modifications (generally simplifications) of elaborate programs in use at government facilities rather than original efforts. This was done to take advantage of the rather substantial effort which went into the preparation of these programs. Each program which was used has shown good agreement with experiment in at least a limited number of cases. Such a practice also has a number of disadvantages:

1. The available documentation is usually very sketchy and frequently inconsistent with the program statements and/or logic. As a result it is very difficult to determine in detail the method on which the program is based and the validity and/or applicability of the methods.

2. The programs usually contain many more options than are needed for the present purposes. It is often difficult to unravel the program to the point that these unneeded options can be removed successfully.

3. The programs are usually written to take advantage of the characteristics of a particular machine which limits their transferability to other machines.

4. In every case the programs are written for very large machines. Smaller machines generally have insufficient storage capacity even to compile the programs. In order to use them on smaller machines one must devise a means of splitting a program into several parts or employing a form of virtual storage.

The present work represents an effort at overcoming these disadvantages. It begins with a review of the literature on the estimation of lift and drag characteristics of wings, wing-bodies, and complete aircraft configurations. Among those treated in this discussion are a group of government reports which describe computer programs for performing various portions of this estimation task in a rapid but accurate manner. Several of these programs appeared to offer a sufficient reduction in the cost of estimating the aerodynamic characteristics of new or modified designs that it seemed desirable to adapt them for use with light aircraft, the computer capabilities of this industry, and as an instructional device for fledgling designers. For these reasons, those portions of the programs dealing with the effects of flap deflections have been removed. The modified programs are therefore more applicable to the higher speed portions of the flight profile. Studies are currently underway of means for including the computation of these effects with reasonable additional computer requirements.

In the next section of the work the theoretical bases of the recommended programs are discussed starting from first principles. It should be emphasized that the methods described are not always exactly those used by the computer programs. The approach to the problem is <u>usually</u> the same but the details are frequently quite different. This has been done because, as noted above, the details of the methods actually used are obscure, at least to the present authors, and because a different treatment was regarded as being easier for those approaching the area for the first time to understand.

Following this discussion is a review of the changes in the programs, instructions for their use, and some sample results. Included also are appendices providing locally-written computer programs found useful for producing analytical check cases, simple approximate solutions to more general computations, or extensions of the range of the major programs to other speed regimes.

The present work is intended to serve several needs. Its primary function is to provide the practicing light aircraft designer with a powerful tool for reducing the engineering labor needed to develop a new airplane or revise an existing one. Hopefully, it is written at such a level and in sufficient depth that the user will be able to gain an understanding of the limitations imposed on the attainable accuracy by the choice of physical and mathematical models as well as an appreciation for the new capabilities provided by the programs and instructions for their use. By keeping the mathematical sophistication required for comprehension to a minimum and by emphasizing physical descriptions of the means by which flows over aircraft are represented, it is hoped that undergraduate aeronautical engineering students will also find the work both helpful and illuminating. It seems unfortunate that because of time limitations, a lack of technical maturity on the student's part, and a reluctance on many educators' part to depart from traditional practice, flight vehicle design is still taught largely as a semi-empirical art rather than as the near-science which it has lately become. Perhaps with the aid of these more powerful less time-consuming tools the student can now successfully complete more realistic design problems during his undergraduate education.

LITERATURE REVIEW AND THEORETICAL

BASIS OF COMPUTER PROGRAMS

LITERATURE REVIEW

INTRODUCTION

Given the task of creating an entirely new airplane, the designer will usually seek to devise first a wing geometry and, ultimately, a whole airplane geometry that

1. provides the required lift

2. has suitable stall characteristics

3. has minimum drag for good performance

4. has good stability and control characteristics

5. meets structural requirements

6. is easy to build.

He will usually select a configuration that satisfies the last two objectives reasonably well and then attempt to determine how well the configuration meets the other objectives. He recognizes that he need not calculate the aerodynamic forces acting on the vehicle with great accuracy in order to determine the flying qualities. On the other hand, if he is to predict the craft's performance with reasonable accuracy, he must know the lift and drag as precisely as possible.

From the viewpoint of designers active during the early years of this century the analysis process was very ill-defined. One did not then even know how much wing he should provide or what shape to make it in order to insure that his aircraft would fly. Being able to estimate how fast or how far his craft might go seemed a matter of secondary concern to the more urgent problem of how much lift is associated with a particular geometry. A systematic study of this problem would seem to begin with consideration of the lift developed by a slice or section out of the wing. Modeling the problem in this fashion has the advantage that one need consider only flow in two dimensions rather than in three, a great mathematical simplification. Further it would seem reasonable to assume that the fluid is inviscid if for no other reason than to take advantage of the extensive analytical studies (particularly those of Helmholtz (Ref. 4) and Kirchhoff (Ref. 5)) that had been carried out for this case during the nineteenth century. These studies had been successful at explaining several experimental facts and present far less mathematical difficulty than one would encounter working the more general equations for the flow of a viscous fluid formulated by Navier and by Stokes about 1840. A good account of much of this work may be found in Lamb (Ref. 6).

The immensity of the problem facing engineers in 1900 trying to devise a rational means of calculating wing lift can be better appreciated when

6

one realizes that in the contemporary view lift was the force reacting to
the change in the momentum of the airstream striking the inclined lower
surface of wing. Such a force would be proportional to $\sin^2 \alpha$ where α is
the angle by which the lower surface is inclined to the wind. If one were
to assume that a wing is flying at fifty miles an hour with $\alpha = 6°$, then
it could develop about 0.0635 pounds of lift per square foot of surface,
according to this theory. Since it was then impossible to build a wing
lighter than this weight, many scientists confidently predicted that man
would never fly. More preceptive individuals noted however that the
flight of gliders could not be explained by such small values of lift
and therefore something must be wrong with the theory.

THE AIRFOIL IN INVISCID FLOW

Lord Rayleigh had shown in 1878 that the swerving flight of a "cut"
tennis ball could be explained at least in general terms by comparing it to
the case of a cylinder placed in an inviscid uniform stream. By superposing
a circulatory flow upon the cylinder, the cylinder developed a force normal
to the direction of the uniform stream, directly proportional to the strength
of the circulatory flow. This result along with the earlier work of Helmholtz
and Kirchhoff was known to the German mathematician M. W. Kutta who was
interested in why cambered airfoils produce lift at $\alpha = 0$. In a 1902 paper
(Ref. 7) he studied a thin airfoil formed by a circular arc. He concluded
that the only reasonable assumption one could make in view of what was
known physically was that the flow velocity over the upper surface was
equal to that over the lower surface at the trailing edge. The flow would
therefore leave the surface smoothly at finite velocity. He was willing
to accept the idea of an infinite velocity at the sharp leading edge, a
situation studied by Helmholtz, in order to obtain an approximate solution
for the lift. Von Kármán (Ref. 8) gives a highly readable account of this
early work.

Joukowski (Ref. 9), working independently along somewhat parallel
lines, was able to obtain exact solutions for a certain class of airfoils
in inviscid flow. He first showed that when a cylindrical body of arbitrary
cross-section moves with velocity, V, in a fluid whose density is ρ and
there is a circulation of the magnitude, Γ, around the body, a force is
produced equal to the product $\rho V \Gamma$ per unit length of the cylinder. The
direction of the force is normal both to the velocity, V, and the axis of
the cylinder. Joukowski also assumed the flow to leave the airfoil smoothly
at the trailing edge. By means of this hypothesis the whole problem of
lift becomes purely mathematical: one has only to determine the amount of
circulation so that for zero vertex angle at the trailing edge the velocity
of the flow leaving the upper surface is equal to the flow leaving the lower
surface. If the tangents to the upper and lower surfaces form a finite
angle, the trailing edge is a stagnation point.

Joukowski then found a transformation, $\zeta = z + C^2/z$, by which a circle
in the z-plane becomes an airfoil in the ζ-plane. See the following sketch.

$$\zeta = \xi + i\eta$$
$$z = x + iy$$
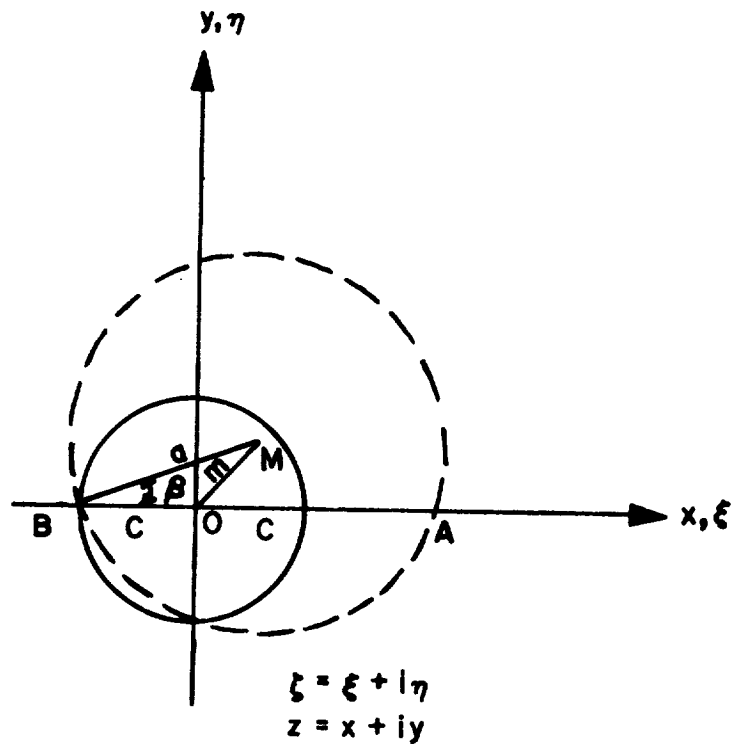
According to the transformation, a point represented by $z = x + iy$ in the x,y plane is moved to a different location in the ξ,η plane. In the process, all the other points in the plane are moved in such a way that the figure of a circle in the x,y plane becomes an airfoil in the ξ,η plane. Under Joukowski's transform the shape of the airfoil may be changed to a considerable extent by moving the center of the circle (originally at O) to some other location M while keeping the point at which the circle crosses the x-axis in the left half-plane at B. To see how this happens it is instructive to carry out a sample calculation.

The general equation of a circle is of course

$$(x - x_1)^2 + (y - y_1)^2 = a^2$$

where
$$x_1 = a \cos \beta - C = \frac{a^2 - C^2 - m^2}{2C}$$

$$y_1 = a \sin \beta$$

in the notation of the sketch. For simplicity one may assume that in this calculation $\beta = 0$. Then

$$[x - (a - C)]^2 + y^2 = a^2$$

or
$$y = \sqrt{2(a - C)x + 2aC - C^2 - x^2} \quad .$$

8

Now
$$\zeta = x + iy + \frac{C^2}{x + iy}$$

$$= x + iy + C^2 \left(\frac{x - iy}{x^2 + y^2} \right)$$

$$= x \left(1 + \frac{C^2}{x^2 + y^2} \right) + iy \left(1 - \frac{C^2}{x^2 + y^2} \right).$$

Substituting values for $x^2 + y^2$ and y into this expresion yields

$$\zeta = x \left(1 + \frac{C^2}{2aC - C^2 + 2x(a - C)} \right)$$

$$+ i \left(1 - \frac{C^2}{2aC - C^2 + 2x(a - C)} \right) \sqrt{2(a - C)x + 2aC - C^2 - x^2}.$$

Since a and C are arbitrary numbers, choosing x completely specifies the value of ζ. For example, let C = 1 and a = 1.1. For this special case the previous equation becomes

$$\zeta = x \left(1 + \frac{1}{1.2 + 0.2x} \right) + i \left(1 - \frac{1}{1.2 + 0.2x} \right) \sqrt{1.2 + 0.2x - x^2}.$$

The equation is easily evaluated and the results presented in tabular form. The table below may be extended to determine the shape of the resulting figure more accurately, if desired.

x	ξ	η
1.2	2.035	0.
1.	1.707	+.1855
1.	1.707	-.1855
0.5	0.885	+.236
0.5	0.885	-.236
0.	0.	+.182
0.	0.	-.182
-1.	-2.	0.

Even from this limited set of numbers, however, it is apparent that for these values of a and C the circle <u>maps</u> into a symmetrical airfoil-like figure of high thickness-to-chord ratio. Moving M to the right increases airfoil thickness while moving M in the y-direction adds camber to the airfoil. Note that the airfoil chord is approximately 4C. Note also that point A becomes the leading edge of the mean camber line and point B the trailing edge of the airfoil under the transformation. When the angle of attack is changed, the flow strikes the airfoil from a different direction. To represent this situation, the strength of the circulation must be changed so that as far as the flow over the cylinder in the x-y plane is concerned the forward stagnation point has moved to some new location obtained by rotating the line MA through an angle α, α being positive when A moves down (y becomes negative). The location of the rear stagnation point must, for reasons pointed out in the next chapter, remain fixed during this operation.

Since the transformation is conformal, the fluid velocity and pressure which exist at any point on the surface of the cylinder can be related quantitatively, as indicated below, to those which exist at the corresponding point on the airfoil. Integration of these pressures in the direction normal to the free stream velocity then gives the airfoil lift (which is also the same as the lift produced by the generating cylinder).

For the cylinder, the surface velocity components are given by

$$u = V[\cos \alpha(1 - \cos 2\theta) + \sin \alpha \sin 2\theta] + \frac{\Gamma y}{2\pi a^2}$$

$$v = V[\cos \alpha \sin 2\theta - \sin \alpha(1 - \cos 2\theta)] - \frac{\Gamma x}{2\pi a^2}$$

while the surface pressures are given by

$$P_{CIRCLE} = P_{STAGNATION} - \frac{\rho}{2}\left(u^2 + v^2\right) \quad .$$

Here θ is the angular location of the point of interest on the surface measured from the negative x-axis. Hence $x = a \cos \theta$ and $y = a \sin \theta$. One may use these values in the procedure outlined above to find that location on the airfoil corresponding to θ. The velocity on the airfoil surface is simply the velocity at the equivalent point on the circle times $|dz/d\zeta|$. From the transform

$$\left|\frac{dz}{d\zeta}\right| = \left|\frac{1}{d\zeta/dz}\right| = \left|\frac{1}{1 - C^2/z^2}\right| = \left|\frac{z}{z - C^2/z}\right|$$

thus the airfoil surface pressure is given by

$$P_{AIRFOIL} = P_{STAGNATION} - \frac{\rho}{2}\left(u^2 + v^2\right)\left|\frac{z}{z - C^2/z}\right|^2 \quad .$$

It is interesting to note that while the theory places no limit on the magnitude of Γ, a value greater than $\Gamma = 2\pi Va$ means that the front and rear stagnation points have come together and are moving away from the circle along the ray $\theta = -\pi/2$, clearly a physically impossible situation since it would mean a strong cyclonic flow was present about the airfoil. In actual cases Γ seldom exceeds $\pi Va/4$.

The Joukowski transform technique was a great step forward in analyzing the lift of airfoils. It gives the correct variation of lift with angle of attack and predicts lift values which are very close to measured values at the same angles of attack. Unfortunately the Joukowski transform techniques also had a number of disadvantages:

1. It is an inverse technique, that is, one does not know beforehand precisely what the airfoil will look like. As a result it is difficult to use the technique to estimate the characteristics of a given airfoil.

2. It leads always to an airfoil with a cusp at the trailing edge. This is impractical structurally.

3. It leads to airfoils which have their minimum pressure point very far forward. Consequently, they have thick boundary layers, and therefore higher drag and lower maximum lift values than airfoils with the minimum pressure point further aft.

4. Being an inviscid theory, it cannot be used to estimate either lift characteristics near stall or drag values.

5. It is tedious to determine the ordinates of the airfoil accurately.

These deficiencies were soon recognized and many investigators set about devising more general transforms which could be used to represent a greater variety of airfoils, in particular those with finite trailing edge angles. Kármán and Trefftz (Ref. 10), von Mises (Ref. 11), Müller (Ref. 12), and Theodorsen (Ref. 13) were among the leaders in this effort, which by 1932 had reached the point where one could determine the lift characteristics of a great variety of airfoils. The great effort required to complete a calculation, however, discouraged thoughts of a further generalization in the transform technique. The following outline of Theodorsen's method will indicate the labor required.

The transform or mapping function is built up in two stages; in the first the airfoil profile in the z-plane is mapped into a contour in the ζ'-plane through the use of the Joukowski transformation

$$z = \zeta' + \frac{C^2}{\zeta'} \quad .$$

It is desirable that the contour in the ζ'-plane be as close to a circle as possible; for this reason the axes in the z-plane should be chosen with a view toward producing that result. This means that the airfoil should be distributed as near like an ellipse as possible with respect to the axes in the z-plane.

The second stage consists of <u>finding</u> a mapping function which will transform the near-circle in the ζ'-plane to an exact circle in the ζ-plane. Theodorsen used the transform

$$\zeta' = \zeta \exp\left(\sum_{n=1}^{\infty} \frac{C_n}{\zeta^n}\right) \quad ,$$

where the coefficients C_n, complex in general, have to be determined.

A point on the near or pseudo circle in the ζ'-plane is given by

$$\zeta' = C\, e^{\psi(\theta)}\, e^{i\theta} \quad .$$

The factor $e^{\psi(\theta)}$ determines how much the contour in the ζ'-plane departs from that of a circle. The relationship between points on the airfoil and points

11

on the pseudo-circle is given by the equations

$$x = 2C \cosh \psi \cos \theta$$

$$y = 2C \sinh \psi \sin \theta .$$

These two equations can be put into the form

$$2 \sin^2 \theta = P + \left[P^2 + \left(\frac{y}{C} \right)^2 \right]^{1/2}$$

$$2 \sin^2 \psi(\theta) = -P + \left[P^2 + \left(\frac{y}{C} \right)^2 \right]^{1/2} ,$$

where

$$P = 1 - \left(\frac{x}{2C} \right)^2 - \left(\frac{y}{2C} \right)^2 ,$$

which establishes the function $\psi(\theta)$.

Theodorsen describes a point on the exact circle in the ζ-plane by the equation

$$\zeta = Ce^{\psi_0}e^{i\phi} = Re^{i\phi}$$

where ψ_0 is a constant not yet determined. R is, of course, also a constant. The relationship between points on the pseudo-circle and those on the exact circle is given by

$$\frac{Ce^{\psi(\theta)}e^{i\theta}}{Ce^{\psi_0}e^{i\theta}} = \frac{\zeta'}{\zeta} = \exp \left(\sum_{n=1}^{\infty} \frac{C_n}{\zeta^n} \right) .$$

Setting

$$\frac{C_n}{\zeta^n} = \frac{A_n + iB_n}{\zeta^n} = \frac{A_n + iB_n}{R^n} e^{-in\phi}$$

and equating real and imaginary parts one obtains

$$\psi - \psi_0 = \sum_{n=1}^{\infty} \frac{1}{R^n} (A_n \cos n\phi + B_n \sin n\phi)$$

$$\theta - \phi = \sum_{n=1}^{\infty} \frac{1}{R^n} (B_n \cos n\phi - A_n \sin n\phi) .$$

From these it follows that

12

$$\psi_0 = \frac{1}{2\pi} \int_0^{2\pi} \psi(\phi) d\phi$$

$$\frac{A_n}{R^n} = \frac{1}{\pi} \int_0^{2\pi} \psi(\phi) \cos n\phi \, d\phi$$

$$\frac{B_n}{R^n} = \frac{1}{\pi} \int_0^{2\pi} \psi(\phi) \sin n\phi \, d\phi \quad .$$

The foregoing equations define ψ_0, A_n, and B_n in terms of $\psi(\phi)$ or, equivalently, $\theta(\phi)$. Since $\psi(\theta)$ is usually not easily extracted and when it has been it is not a simple form, the evaluation of the various coefficients is best handled numerically or by a combination of graphical constructions, approximations, and iterations. Theodorsen's original method followed the second course. The original paper may be consulted for details. To use the method today one would employ numerical techniques.

Once the process has been completed by whichever means are employed, one then has the pressures and velocities at each point on the airfoil surface in terms of those at the equivalent point on the exact circle.

Analyses of the lift characteristics of various Joukowski airfoils in the meantime revealed that the thickness contributed little to the lift. It therefore seemed to some that if airfoils for which one had difficulty finding appropriate conformal transforms could be characterized by their mean camber lines only, then perhaps one would have a relatively simple, yet direct method of evaluating the lift and pressure distribution of arbitrary airfoils. Such an approach is obviously most appropriate when the actual airfoils are thin. These ideas were developed in the early 1920's by Munk (Ref. 14), Birnbaum (Ref. 15), and Glauert (Ref. 16).

In Glauert's conception the airfoil is replaced by its mean camber line which he assumed, never lies very far from the chord line. For this reason he felt justified in making the approximation that the velocities over the airfoil could be represented by a continuous distribution of vortices (or a sheet of vorticity)* lying along the chord line. The variation in vorticity with chord location is not known initially. The velocity induced at point x' on the chord of the airfoil due to the vortex sheet is given by

$$v(x') = \int_0^c \frac{\gamma dx}{2\pi(x - x')} \quad ,$$

---

* The reader unfamiliar with the theoretical basis of the concept is referred to the next section of the present work or to Reference 17 for complete mathematical details.

where γ is the vortex strength per unit length. This induced velocity is actually calculated for a point on the chord but, according to Glauert's approximation, may be taken to be the same as the induced velocity at the corresponding point of the airfoil itself.* Since the resultant of the free stream velocity and the induced velocity adjacent to the airfoil must be parallel to the surface at each point of the airfoil and since the flow angularities are small, one may write this statement as

$$\alpha + \frac{v}{V} = \frac{dy}{dx} ,$$

where dy/dx is the slope of the mean camber line at x'. It will be seen that these two equations are sufficient to provide a complete solution of the problem in terms of the shape of the curved line which represents the airfoil. The solution is obtained as $\gamma(x)$. Then according to Joukowski's theorem

$$L = \int_0^c \rho V \gamma \, dx$$

$$M = \int_0^c \rho V \gamma x \, dx \quad .$$

The method Glauert employed to find $\gamma(x)$ is instructive because most subsequent calculative procedures use refinements of the same idea. Glauert first changed the independent variable x to θ according to the transformation

$$x = \frac{c}{2} (1 - \cos \theta) \quad .$$

He assumed that he could represent γ by a sine series in θ:

$$\gamma = 2V \left\{ A_0 \cot \theta/2 + \sum_{n=1}^{\infty} A_n \sin n\theta \right\} \quad .$$

Hence

$$\gamma dx = cV \left\{ A_0(1 + \cos \theta) + \sum_{n=1}^{\infty} A_n \sin n\theta \sin \theta \right\} d\theta \quad ,$$

then

$$v(x') = \frac{V}{\pi} \int_0^\pi \frac{A_0(1 + \cos \theta) + \frac{1}{2} \sum_{n=1}^{\infty} A_n \{\cos (n - 1)\theta - \cos (n + 1)\theta\}}{\cos \theta' - \cos \theta} d\theta$$

---

* Karamcheti (Ref. 17) presents a very detailed discussion of the relation of this approximation to the exact formulation.

14

$$= V \left\{ - A_0 + \frac{1}{2} \sum_{n=1}^{\infty} A_n \frac{\sin (n + 1)\theta' - \sin (n - 1)\theta'}{\sin \theta'} \right\}$$

$$= V \left\{ - A_0 + \sum_{n=1}^{\infty} A_n \cos n\theta' \right\} .$$

Substituting of this result in the second of the two original equations gives

$$\frac{dy}{dx} = \alpha - A_0 + \sum_{n=1}^{\infty} A_n \cos n\theta' .$$

According to the theory of Fourier series the coefficients A_n are determined from the shape of the airfoil by evaluating the integrals

$$\alpha - A_0 = \frac{1}{\pi} \int_0^\pi \frac{dy}{dx} d\theta$$

$$A_n = \frac{2}{\pi} \int_0^\pi \frac{dy}{dx} \cos n\theta \, d\theta$$

where dy/dx now is the slope of the surface at any x between 0 and c as a function of θ. The integral can, of course, be evaluated piecewise if the functional form changes as θ goes from 0 to π.

Glauert showed that one need find only A_0, A_1, and A_2 in order to determine C_L and C_{M_O}. Note that

$$\frac{L}{\frac{1}{2}\rho V^2 c} = C_L = \frac{2}{\rho c V^2} \int_0^\pi \rho c V^2 \left\{ A_0 (1 + \cos \theta) + \sum_{n=1}^{\infty} A_n \sin n\theta \sin \theta \right\} d\theta$$

$$= 2\pi (\dot{A}_0 + \tfrac{1}{2} A_1)$$

and similarly that

$$C_{M_O} = \frac{\pi}{2} (A_0 + A_1 - \tfrac{1}{2} A_2) = \frac{\pi}{4} (A_2 - A_1) - \tfrac{1}{4} C_L .$$

He also showed that

$$A_0 + \tfrac{1}{2} A_1 - \alpha = - \frac{1}{\pi} \int_0^\pi \frac{dy}{dx} (1 - \cos \theta) d\theta = \frac{2}{\pi} \int_0^\pi \frac{y}{c} \frac{d\theta}{1 + \cos \theta} ,$$

thus

$$C_L = 2\pi \left(\alpha + \frac{2}{\pi} \int_0^\pi \frac{y}{c} \frac{d\theta}{1 + \cos \theta} \right) .$$

15

Since Glauert also showed that

$$\int_0^\pi \frac{y}{c} \cos \theta \, d\theta = \frac{\pi}{4} (\alpha - A_0 - \tfrac{1}{2}A_2) \, ,$$

$$C_{M_O} = 2 \left(\int_0^\pi \frac{y}{c} \cos \theta \, d\theta - \tfrac{1}{2} \int_0^\pi \frac{y}{c} \frac{d\theta}{1 + \cos \theta} \right) - \tfrac{1}{4}C_L \, .$$

Values of C_L and C_{M_O} computed by this method Glauert found to be "in close agreement with experimental determinations of these quantities." The method can be seen to be considerably simpler to use than the transform technique. During the 1930's when designers sought to reduce wing drag by eliminating external bracing, they were forced by structural considerations to abandon the very thin airfoils they had been using until that time. They found that in order to predict the lift and moment characteristics of the newer and thicker airfoil sections that were then becoming the vogue, more elaborate analytical methods or extensive wind tunnel testing were necessary. One of these analytical methods took the following tack. Since the sum of solutions to the Laplace equation (the equation describing inviscid, incompressible flow) is also a solution, one can describe a thick, cambered airfoil at angle of attack by superimposing solutions for a curved line (Glauert's method), a flat plate at angle of attack (also represented by a vortex sheet), and a thick symmetrical airfoil at $\alpha = 0$ (using a distribution of sources along the chord line). An example of such a built-up solution is given by Karamcheti (Ref. 17). Reference 18 provides an exposition of both the thin airfoil and Theodorsen approaches and indicates how these techniques were used to guide the very significant series of experimental investigations carried out during the 1930's by the NACA.

These investigations sought to measure in considerable detail aerodynamic characteristics of several general families of airfoils. Since these data were obtained in well-calibrated wind tunnels at flight Reynolds numbers and presented valid drag data as well, designers came to regard NACA TR-824, "A Summary of Airfoil Data," (Ref. 19) and its forerunners as their primary data source. It has only been within the last 20 years or so that interest in improved analytical methods has been rekindled. This revival perhaps can be attributed to the simultaneous occurance of

1. recent, sharp escalation in the cost of making models and conducting tests,

2. the desire to optimize certain aspects of airfoil behavior and to investigate the characteristics of unconventional airfoils,

3. the appearance of the large digital computer which made it possible to consider the use of what had previously been rather laborious methods on a routine basis.

16

One of the first and most widely used methods of the current revival is that described by J. Weber (Refs. 20, 21). In the earlier of the two papers she treated the case of a symmetrical two-dimensional airfoil at angle of attack. By transforming an airfoil into a slit, she was able to show that the source distribution which she used to represent the thickness at zero lift can be placed along the chord line rather than on the surface with little error, provided the airfoil is no thicker than about 10% of the chord. With that assumption and the superposition of a vortex distribution on a flat plate at angle of attack, Weber obtained the equation

$$V(x,z) = \frac{V_o}{\sqrt{1 + (\frac{dz}{dx})^2}} \left\{ \cos \alpha \left[1 + \frac{1}{\pi} \int_0^1 \frac{dz}{dx} \frac{dx'}{x - x'} \right] \right.$$

$$\left. \pm \sin \alpha \sqrt{\frac{1 - x}{x}} \left[1 + \frac{1}{\pi} \int_0^1 \left(\frac{dz}{dx} - \frac{2z(x')}{1 - (1 - 2x')^2} \right) \frac{dx'}{x - x'} \right] \right\} .$$

The positive sign holds for the upper surface, the negative sign for the lower surface. $V(x,z)$ is the velocity along the airfoil surface. The pressure coefficients along the surface are given by

$$C_p = 1 - \left(\frac{V(x,z)}{V_o} \right)^2 .$$

The most attractive feature of the method is Weber's technique for finding a numerical value of $V(x,z)$. She begins by making the following definitions

$$S^{(1)}(x) = \frac{1}{\pi} \int_0^1 \frac{dz}{dx'} \frac{dx'}{x - x'}$$

$$S^{(2)}(x) = \frac{dz}{dx}$$

$$S^{(3)}(x) = \frac{1}{\pi} \int_0^1 \left[\frac{dz}{dx} - \frac{2z(x')}{1 - (1 - 2x')^2} \right] \frac{dx'}{x - x'} .$$

She then evaluates these quantities at specific points, x_ν, along the chord using the representation

$$S^{(1)}(x_\nu) = \sum_{\mu=1}^{N-1} S_{\mu\nu}^{(1)} z_\mu$$

$$S^{(2)}(x_\nu) = \sum_{\mu=1}^{N-1} S_{\mu\nu}^{(2)} z_\mu$$

$$S^{(3)}(x_\nu) = \sum_{\mu=1}^{N-1} S_{\mu\nu}^{(3)} z_\mu + S_{N\nu}^{(3)} \sqrt{\frac{\rho}{2c}} \quad .$$

The coefficients $S_{\mu\nu}^{(1)}$, $S_{\mu\nu}^{(2)}$, $S_{\mu\nu}^{(3)}$ are independent of airfoil shape and Weber gives tables of their values. N is the number of points used to approximate the airfoil (she gives tables for 8, 16, and 32 points). Point #1 is always at the trailing edge. ρ is the leading edge radius and c is the chord length. Weber also gives a table for finding x_ν corresponding to a given value of ν. These are the chordwise stations at which the pressure is calculated. z_μ is the airfoil ordinate corresponding to the chord location given by

$$x_\mu = \tfrac{1}{2}\left(1 + \cos\frac{\mu\pi}{N}\right)$$

where $1 < \mu < N-1$. It will be seen that with the aid of the tables of universal coefficients $S_{\mu\nu}^{(1)}$, $S_{\mu\nu}^{(2)}$, $S_{\mu\nu}^{(3)}$ the pressure computation is carried out very easily using a desk calculator.

A significant feature of Weber's method is her retention of the factor $1/\sqrt{1 + (dz/dx)^2}$ which materially improves the accuracy of the pressure computation near the leading edge. (See Appendix F.)

Weber extended her approach to treat cambered airfoils in a second paper (Ref. 21). She showed that two additional terms are required in the expression for pressure to account for camber:

$$c_p = 1 - \frac{\{\cos\alpha\,[1 + S^{(1)}(x) \pm S^{(4)}(x)] \pm \sin\alpha\,\sqrt{(1-x)/x}\,[1 + S^{(3)}(x)]\}^2}{1 + [S^{(2)}(x) \pm S^{(5)}(x)]^2}$$

where

$$S^{(5)}(x) = \sum_{\mu=1}^{N-1} S_{\mu\nu}^{(5)} z_{c_\mu} = \left(\frac{dz_c}{dx}\right)_{x_\nu}$$

$$S^{(4)}(x) = \sum_{\mu=1}^{N-1} S_{\mu\nu}^{(4)} z_{c_\mu} = \frac{\Gamma(x_\nu)}{2V_o} \quad .$$

The subscript "c" refers to the camber line. Weber gives tables for $S_{\mu\nu}^{(4)}$ and $S_{\mu\nu}^{(5)}$ and also provides some second order corrections to aid in predicting the pressure in the nose regions more accurately.

18

Comparisons between Weber's results and exact theory for Joukowski airfoils indicate that her method predicts pressures which are low by about 1%. Maximum camber must be less than about 4% of chord and thickness less than 10% of chord to obtain results of this accuracy, however.

The success of Weber's approach and its obvious adaptability to computer solution (see for example Reference 22) seems to have served as a spur to the development of more exact airfoil representation schemes which are practical only if carried out by digital computer. The method of Hess and Smith (Ref. 23) is among the best known of these developments. In this method the non-lifting airfoil surface is replaced by a source sheet with strength $\sigma(s)$ where s is the distance measured along the airfoil surface. The sum of the velocity induced by the source sheet and the free stream velocity is forced to satisfy the condition that its component normal to the airfoil surface at each value of s is zero. This statement is written mathematically as a Fredholm integral equation of the second kind:

$$2\pi\ \sigma(s)\ +\ \oint \sigma(s')\ \ell n\ r(s,s')ds'\ =\ F(s)$$

where $r(s,s')$ is the distance between the point of interest, s, and any other point on the surface, s'; $\sigma(s')$ represent the source strength at points other than s; $\sigma(s)$ is the source strength at s; and $F(s)$ represents the component of the free stream velocity normal to the surface at s. The left side of the equation then represents the component of the velocity induced by the source sheet which is normal to the surface. Note that for a given airfoil in a stream of known speed the unknown quantity is $\sigma(s')$ which occurs under the integral sign.

To solve this equation Hess and Smith make several approximations:

1. the contour of the airfoil can be represented by N straight line segments,

2. $\sigma(s')$ is constant over each segment,

3. the integral is evaluated at only one point—generally the mid-point—of each segment.

This leads to a system of N simultaneous linear equations which can be solved to find σ on each segment. For good accuracy, N must be large, particularly in regions of high curvature. Knowing σ one can then find the tangential component of velocity from which one can compute the surface pressure.

To treat the lifting airfoil, Hess and Smith in effect superpose a vortex sheet of suitable strength so that the total flow satisfies the local tangency condition as well as the Kutta condition at the trailing edge.

Martensen (Ref. 24) chose a different approach. He represented the airfoil by a vortex sheet on its surface. By requiring that the strength

19

of the vortex sheet be identical to the velocity distribution on the
surface of the airfoil, Martensen was able to show that in the interior of
a closed vortex sheet the velocity is everywhere zero. Thus on the
inner side of the vortex sheet the net tangential velocity which is a
sum of that due to free stream and that due to the vortex sheet is zero, or

$$\frac{\Gamma(s)}{2} - \frac{1}{2\pi} \frac{\partial}{\partial n} \oint \Gamma(s') \ln r(s,s')ds' = V_\infty \left(\frac{dx}{ds} \cos \alpha + \frac{dy}{ds} \sin \alpha \right).$$

This equation has almost exactly the same form as that formulated by
Hess and Smith.

To solve the equation Martensen chose, as did Hess and Smith, to
replace the integral by a summation. As a result he also ended up solving
a system of simultaneous equations. An equation expressing the Kutta
condition is required to complete the formulation. Martensen's method
while giving the velocity distribution on the surface as the solution
to the system of equations does not give good results for very thin
airfoils. The reason is that when the upper and lower surface control
points are very close together, the vortices located there induce strong
tangential velocities on each other. While this induced velocity
actually decays very rapidly for points in the neighborhood of the control
point, the method of approximating the integral which Martensen used
assumes it to be a constant. Jacob (Ref. 25) used a different limiting
approximation which improves the results but at the cost of restricting
one's freedom in distributing the control points on the airfoil surface.

If an inviscid fluid flow is.everywhere parallel to the surface of
a closed body then the surface of the body can be represented by a
streamline on which the stream function, ψ, is constant. Oellers (Ref. 26)
used this idea to write

$$\psi = V_\infty y(s) \cos \alpha - V_\infty x(s) \sin \alpha - \frac{1}{2\pi} \oint \Gamma(s') \ln r(s,s')ds' .$$

To solve this equation for ψ and $\Gamma(s')$, the integral is approximated by
a summation of the type used by Hess and Smith.

Chen (Ref. 27) made a very detailed comparison of the three foregoing
methods. He found that when applied to airfoils for which analytical
expressions for the pressure distribution are known, the Hess-Smith method
always gives the correct value of circulation generated by the airfoil.
On the other hand the computed surface velocity was found to be very
sensitive to the coordinates of the control points. A tiny error in the
input coordinates can produce a wavy behavior of large amplitude in the
computed surface velocity, more so than is reasonable physically. Chen
also tried approximating Martensen's integral in the same way Hess and
Smith approximated theirs. He found that the resulting circulation was
smaller than that found by Hess and Smith because the integration is
carried out along straight line segments rather than curves. This also
leads to some difficulties in the numerical computation because the

20

matrix of coefficients is ill-conditioned. Even after curvature effects are taken into account, the circulation computed by the Martensen-Jacob method, although larger, is still slightly smaller than that obtained by the Hess-Smith method. On the other hand, because it is a vortex sheet and tangential velocities which are considered by Martensen and Jacob, the computed results are not very sensitive to inaccuracies in the values of the input coordinates.

Chen prefers Oellers' method, primarily because it leads to fewer computational difficulties. Since it is an integral representation, no surface slopes must be computed, a process which always causes some loss in accuracy. Secondly, because the kernel of the integral equation is simpler in this formulation, the computing time required is generally less.

Several improvements in the transform approach to predicting airfoil characteristics have also appeared in recent years. Lighthill (Ref. 28) chose to specify the desired velocity distribution about the airfoil in closed form. Sato (Ref. 29) extended this approach to permit a velocity distribution of any kind to be specified. As worked out by Sato, the velocity distribution is assumed in such a way that front and rear stagnation points can be treated separately. A well-behaved function $g(\theta)$ takes up the velocity distribution everywhere with the exception of the stagnation points and three constants which are imbedded. The constants are determined by $g(\theta)$, the fact that the airfoil is a closed curve, and the fact that the flow field at infinity is uniform. A set of initial values must be given to the three constants in order to obtain $g(\theta)$ from the specified velocity distribution. This $g(\theta)$ is then used to obtain a new set of values for the constants which will give a closed curve as the airfoil geometry. The process is repeated iteratively until the before and after constant values match. In this way Sato's method always guarantees an airfoil geometry giving the desired velocity distribution. Because of the repetitive nature of many of its steps and the need for piecewise integration, it is best done on a digital computer.

THE AIRFOIL IN VISCOUS FLOW

It was of course recognized that all of these approaches would give somewhat optimistic predictions of airfoil lift and no prediction at all of airfoil drag. It was therefore just a matter of time until efforts would be made to attempt to account for the effects of viscosity at least so far as the lift produced by an airfoil is concerned. Powell (Ref. 30) was one of the first to attack the problem in a fairly rigorous fashion. He modified the airfoil geometry in two ways to account for the effects of boundary layer displacement of the inviscid flow. To the airfoil thickness distribution (symmetrical about the mean camber line) he added the total displacement thickness evenly distributed between upper and lower surfaces. He recognized, however, that the displacement thickness is not the same in the two surfaces, being thicker

on the upper surface. He chose to account for this fact by reflexing the trailing edge, adding $\frac{1}{2}(\delta^*_{upper} - \delta^*_{lower})$ to the ordinates of the mean camber line. Because the airfoil did not then physically close at $x = c$, he chose to set the upper and lower pseudo-surface velocities equal at $x = c$ as a replacement for the Kutta condition. He then employed Weber's method to predict the surface pressures. Powell assumed in his computation that $\delta^*(x)$ was available a priori. He also discussed the problem of "closing" the pseudo-airfoil in the wake as a means of finding reasonable surface slopes at $x = c$.

Apparently Powell's paper served as a source of inspiration for the work reported in Reference 31. Although the sketchy nature of the discussion in the report makes it difficult to ascertain precisely the heritage of the approach used or even its particulars, detailed examination of the computer program indicates that the authors (of the Lockheed Georgia Company) actually employed a combination of methods (vortex distribution on surface of a cambered airfoil plus a vortex distribution on a symmetrical airfoil) along with the idea discussed by Powell of modifying camber and thickness separately to account for boundary layer thickness. This represents somewhat of a departure from an earlier version of the computer program (Ref. 32) which is said to be based on Van Dyke's inviscid method (Ref. 33) and earlier British work on viscous corrections which was also considered by Powell. Van Dyke offered a way of treating thicker airfoils by transferring the surface tangency condition to the chord line with a Taylor series expansion. In other respects his approach is equivalent to Weber's.

The significant feature of the Lockheed program is its provision for arriving at the pseudo-airfoil shape in an iterative fashion. The program uses the inviscid pressure distribution and a fairly crude boundary layer computation to obtain the initial estimate of the displacement thickness. The displacement thickness is then used to get a new inviscid pressure distribution. The process is continued for five iterations until the pressure distribution used to compute the displacement thickness is virtually the same as that which one gets after adding the displacement thickness to the airfoil geometry. Computations of skin friction, transition of the boundary layer from laminar to turbulent flow, and location of the separation point are made at the same time. Program output is therefore airfoil lift, drag, and pitching moment as a function of angle of attack. It can also account for variations in free stream Mach number.

The program, which forms the basis of the modified version given herewith, was found, when compared with experimental results, (see for example Reference 52) to:

1.   over-estimate the lift curve slope by 5% to 8%,

2.   give very accurate estimates for the surface pressures except in the neighborhood of large suction peaks or incipicent separation,

22

3. give rather poor estimates of section drag, errors of 30% to 50% being common at low-to-moderate angles of attack; in particular the integration of the inviscid pressures in the drag direction is frequently not zero.

Despite these difficulties, the success of the program and its inherent rigor makes it apparent that with improvements in the boundary layer computation method and in the numerical procedures used an accurate and reliable means of estimating all the aerodynamics characteristics of airfoils at subsonic speeds is at hand. Reference 31 is the basis for the presentation in the next chapter of a theory for the prediction of lift, drag, and moment characteristics of two-dimensional airfoils.

COMPRESSIBILITY

To complete the discussion of developments in theoretical means for predicting the inviscid characteristics of light plane airfoils it is necessary to mention the effect of changes in Mach number. Although non-jet powered aircraft are not likely to reach speeds such that a local sonic point will exist on the airfoil, many do experience sufficiently high speeds to distort the M = 0 pressure distribution significantly. Thus it is desirable to chronicle the efforts which have been made in describing these effects.

Despite the folklore that the speed of sound presented an impregnable barrier to the velocity of flight vehicles, it was recognized quite early by many aeronautical scientists that artillery shells, for example, frequently exceed this speed. One should therefore be able to develop an expression for the pressure forces on a body for situations where the compressibility of the air is not negligible. Studies later showed that by allowing the density to vary in the equations describing the motion of an inviscid fluid but retaining the idea that the airfoil was thin and therefore did not disturb the flow greatly, it was possible to describe the flow over airfoils by the equation

$$\cdot \left(1 - M_\infty^2\right)\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$$

along with suitable boundary conditions. This equation, however, can be written as

$$\frac{\partial^2(\beta\phi)}{\partial x^2} + \frac{\partial^2(\beta\phi)}{\partial(\beta y)^2} = 0$$

where $\beta^2 = 1 - M_\infty^2$, a constant. The form then is that of the Laplace equation with which hydrodynamicists and early aerodynamicists were already familiar. Consistent with the small disturbance idea is the representation of the pressure coefficient at an arbitrary point on the airfoil surface by

$$C_P = \frac{P_L - P_\infty}{\frac{\rho_\infty}{2} U_\infty^2} = \frac{1}{\frac{\rho_\infty}{2} U_\infty^2} \left[P_{STAG} - \frac{\rho_\infty}{2} (U_\infty + u')^2 - P_{STAG} + \frac{\rho_\infty}{2} U_\infty^2 \right]$$

$$= \frac{2}{\rho_\infty U_\infty^2} \left[-\frac{\rho_\infty}{2} (U_\infty^2 + 2u'U_\infty + u'^2) + \frac{\rho_\infty}{2} U_\infty^2 \right]$$

$$= \frac{2}{\rho_\infty U_\infty^2} \left[-\rho_\infty U_\infty u' - \cancelto{0}{\frac{\rho_\infty}{2} u'^2} \right]$$

$$\cong -\frac{2u'}{U_\infty} \quad .$$

But $u' = \beta \frac{\partial \phi}{\partial x}$ and thus

$$C_P = -\frac{2\beta}{U_\infty} \frac{\partial \phi}{\partial x} \quad .$$

For the case where $\beta = 1$ (incompressible flow),

$$C_P = -\frac{2}{U_\infty} \frac{\partial \phi}{\partial x} \quad .$$

Thus,

$$(C_P)_{COMPRESSIBLE} = \frac{C_{P_{INCOMP}}}{\sqrt{1 - M_\infty^2}} \quad .$$

This simple relation was advanced about 1928 by Glauert and by Prandtl independently. It provides a good prediction of experimental results for local Mach numbers over the airfoil less than critical. It fails at higher Mach numbers because for rigor, the equation describing the flow must then contain an additional, non-linear term. This also makes it impossible to compare exactly the same airfoil at two different Mach numbers. Nevertheless, the fact that the Prandtl-Glauert formula permits one to find the pressure distribution over airfoils with reasonable accuracy for all Mach numbers less than that where the flow first becomes sonic merely by knowing the M = 0 distribution led to a very serious search for parameters which can be used to determine when the flow over an airfoil at one Mach number is similar to that over a second airfoil at another Mach number and hence will have the same pressure distribution.

One of the most successful efforts in this direction was the formula

$$C_P = \frac{C_{P_{INC}}}{\sqrt{1 - M_\infty^2} + \frac{M_\infty^2}{\sqrt{1 - M_\infty^2} + 1} \left(\frac{C_{P_{INC}}}{2} \right)}$$

24

proposed by Kármán and Tsien in 1939. (Quoted in Reference 56.) Although not strictly applicable to the flow over the same airfoil at different Mach numbers, it has been used in this way with good results. A good discussion of the theoretical basis for comparing flows over bodies at one Mach number with flows over the same or related bodies at different Mach numbers is given in Chapter 10 of the text by Liepmann and Roshko (Ref. 57).

EXTENSION TO THREE DIMENSIONS

The conceptual basis for expanding an airfoil section laterally into a finite wing can be traced to an 1894 paper by F. W. Lancaster. He elaborated these views in a book (Ref. 34) published in 1907. The mathematical expression of these ideas in a convenient form, however, seems to have originated with Prandtl (Ref. 35). He took a very simplified view of the wing, arguing that because the lift curve slope of all airfoils is nearly the same and because one cannot in any event determine viscous effects from an inviscid theory why not, then, for purposes of determining the effect of planform geometry or twist, represent the wing by just a line located at about the airfoil aerodynamic center. By placing a circulation about this line whose strength varies with the angle of attack of the wing, one can obtain a linear lift curve (C_L versus α) of the correct magnitude. Since such a vortex must either close or extend to infinity, Prandtl assumed that the vortex leaves each wing tip and extends, parallel to the fuselage, to some point very far downstream at which it closes; this can be considered to be at infinity for all practical purposes. Such a flow pattern is then consistent with the vortices observed leaving the tips of lifting wings. By superposing a series of vortices of different strengths and spans but assuming that they all "roll up" into one on leaving the tips, one can represent a rather arbitrary spanwise lift distribution.

Glauert (Ref. 16b) points out that the flow induced by this vortex system is normal to the span and to the direction of the aircraft's motion and is directed downwards in general. This downward flow velocity, w, is small compared with the flight velocity, U, but has the effect of reducing the angle with which the wing meets the oncoming flow, α. The reduction in angle of attack is given by w/U. Since w varies over the span, the induced angle of attack also varies over the span. Further, since the lift is defined as the force normal to the flow direction, the presence of an induced flow angle w/U causes the lift force to tilt backwards, giving a component in the direction of the drag force. This induced drag,

$$D_i = \frac{w}{U} L$$

is an **inviscid** effect resulting from the finite extent of the wing span. It may be noted that the work done on the fluid by the induced drag appears as the kinetic energy of the trailing vortices.

In the lifting line theory, the characteristics of a monoplane airfoil are determined by first finding w and hence the effective angle of attack at each point along the span, then finding the corresponding two-dimensional lift and drag, and finally, integrating across the span. The first step in this process is determining w in terms of the strength of the trailing vortices. Between the points y and y + dy on the span, the circulation Γ can be assumed to fall by an amount $-(d\Gamma/dy)dy$ and hence a trailing vortex of this strength springs from the element of span dy. There is therefore a sheet of trailing vortices extending across the span and the normal induced velocity, w, at any point y_1 on the span contains contributions from all the trailing vortices in this sheet. At y_1, therefore,

$$w(y_1) = \int_{-b/2}^{-b/2} \frac{-\frac{d\Gamma}{dy}\, dy}{4\pi(y - y_1)} \quad .$$

It can also be shown that the circulation around a section of any wing (airfoil) is

$$\Gamma = \tfrac{1}{2} C_L cU = \tfrac{1}{2} C_{L_\alpha} (\alpha - w/U)\, U \quad .$$

Although C_{L_α} actually varies slightly with airfoil geometry it is usually taken to be a constant. This equation, in conjunction with the preceding one, makes it possible to determine the circulation and w for any wing in terms of the local values of c and α. Note, however, that the first of these two equations is an integral equation because one of the unknowns, Γ, appears under the integral sign. This fact is responsible for much of the difficulty incurred in solving the wing lift problem because, in contrast to differential equations, few techniques exist for solving integral equations.

The technique which Glauert (Ref. 16b) suggested for solving the two equations proceeds as follows:

Call
$$y = -\frac{b}{2} \cos \theta$$

and
$$\Gamma = 2bU \sum_{n=1}^{\infty} A_n \sin n\theta \quad .$$

Then
$$w(\theta_1) = \frac{U}{\pi} \int_0^\pi \frac{\Sigma n A_n \cos n\theta}{\cos \theta - \cos \theta_1}\, d\theta = U \sum n A_n \frac{\sin n\theta_1}{\sin \theta_1}$$

from which it may be seen that at any point along the span

$$w \sin \theta = U \sum n A_n \sin n\theta \quad .$$

The second equation connecting Γ and w becomes in this notation

$$2bU \sum A_n \sin n\theta = \tfrac{1}{2} C_{L_\alpha} cU \left\{ \alpha - \frac{\Sigma n A_n \sin n\theta}{\sin \theta} \right\} \ .$$

Letting $\mu = (C_{L_\alpha} c)/(4b)$ one can write this as

$$\sum A_n \sin n\theta \sin \theta = \mu\alpha \sin \theta - \mu \sum n A_n \sin n\theta$$

or

$$\sum A_n \sin n\theta \, (\mu n + \sin \theta) = \mu\alpha \sin \theta \ .$$

In general it is to be expected that μ and α are functions of θ. The problem now is determining the values of the coefficients A_n which will satisfy the foregoing equation at every value of θ along the span of the particular wing in question.

In passing one may note that since

$$L = C_L \frac{\rho}{2} S U^2 = \int_{-b/2}^{b/2} b^2 \rho U^2 \, (\Sigma A_n \sin n\theta) \sin \theta \, d\theta$$

$$= \pi b^2 \frac{\rho}{2} U^2 A_1 \ ,$$

then

$$A_1 = \frac{C_L S}{\pi b^2} = \frac{C_L}{\pi AR} \ .$$

From this result it appears that the wing lift is determined by the value of A_1 and that the other coefficients in the series for the circulation, Γ, modify the shape of the spanwise lift distribution without altering the total lift.

The general procedure for obtaining the coefficients is to write as many equations as coefficients one desires to evaluate, each equation for a different value of θ, and solve the resulting system for the coefficient values. For example, the system

$$A_1\sin \theta_1(\mu_1 + \sin \theta_1) + A_3\sin 3\theta_1(3\mu_1 + \sin \theta_1) + A_5\sin 5\theta_1(5\mu_1 + \sin \theta_1)$$

$$+ A_7\sin 7\theta_1(7\mu_1 + \sin \theta_1) = \mu_1\alpha_1\sin \theta_1$$

$$A_1\sin \theta_2(\mu_2 + \sin \theta_2) + A_3\sin 3\theta_2 \, (3\mu_2 + \sin \theta_2) + A_5\sin 5\theta_2(5\mu_2 + \sin \theta_2)$$

$$+ A_7\sin 7\theta_2(7\mu_2 + \sin \theta_2) = \mu_2\alpha_2 \sin \theta_2$$

$$A_1\sin\theta_3(\mu_3 + \sin\theta_3) + A_3\sin 3\theta_3(3\mu_3 + \sin\theta_3) + A_5\sin 5\theta_3(5\mu_3 + \sin\theta_3)$$
$$+ A_7\sin 7\theta_3(7\mu_3 + \sin\theta_3) = \mu_3\alpha_3 \sin\theta_3$$

$$A_1\sin\theta_4(\mu_4 + \sin\theta_4) + A_3\sin 3\theta_4(3\mu_4 + \sin\theta_4) + A_5\sin 5\theta_4(5\mu_4 + \sin\theta_4)$$
$$+ A_7\sin 7\theta_4(7\mu_4 + \sin\theta_4) = \mu_4\alpha_4 \sin\theta_4$$

can easily be solved for A_1, A_3, A_5, and A_7 by algebraic techniques since the values of θ_1, θ_2, θ_3, θ_4, μ_1, μ_2, μ_3, μ_4, etc. are known from the wing geometry. The value of A_1 becomes independent of n only when n is large.

Only odd coefficients occur in the series because the wing is assumed to be symmetric about its mid-span point.

The induced drag is found easily once the A_n's are known. Since

$$D_i = \int_{-b/2}^{b/2} \frac{w}{U} L\, dy = \int_{-b/2}^{b/2} \rho w\Gamma\, dy = \int_{-b/2}^{b/2} \rho U^2 b^2 (\sum nA_n \sin n\theta)(\sum A_n \sin n\theta)d\theta,$$

$$D_i = \pi b^2 \frac{\rho}{2} U^2 \sum nA_n^2 .$$

Since A_1 is independent of the planform shape, it follows that the induced drag will be a minimum when the other coefficients in the series are zero. The circulation for such a condition is then represented by

$$\Gamma = 2bU \frac{C_L}{\pi AR} \sin\theta = 2bU \frac{C_L}{\pi AR} \sqrt{1 - \cos^2\theta} = 2bU \frac{C_L}{\pi AR} \sqrt{1 - 4y^2/b^2}$$

or the equation of an ellipse. An elliptical span-wise distribution of circulation (lift) can be obtained in practice by an elliptical variation of chord in the spanwise direction or by combinations of taper and twist.

During preliminary design one of the things one seeks to establish, at least approximately, is the relationship between aircraft attitude and lift developed. This entails finding the effect of the finite wing span on the slope of the lift curve. For infinite aspect ratio of course it is about 2π per radian for all airfoils. The lifting-line theory, however, indicates that for an elliptical lift distribution—the most efficient type—there is a reduction in the effective angle of attack of $C_L/\pi AR$. To develop the same lift, the geometric angle of attack must then be $\alpha_{2D} + C_L/\pi AR$. The geometric angle of attack for finite-span wings can also be written

$$\alpha_{2D}\left(1 + \frac{C_{L\alpha 2D}}{\pi AR}\right) \simeq \alpha_{2D}\left(1 + \frac{2}{AR}\right) = \alpha_{3D} .$$

28

Since $C_{L_{\alpha 2D}}\alpha_{2D} = C_{L_{\alpha 3D}}\alpha_{3D}$ when the finite span wing develops the same lift as the infinite span wing, the three-dimensional lift-curve slope is

$$C_{L_{\alpha 3D}} = \frac{2\pi}{1 + \frac{2}{AR}} \quad \text{per radian.}$$

The conclusion drawn from lifting line theory that an elliptical spanwise aerodynamic loading leads to minimum induced drag has been of particular interest to the designers of large aircraft. Since the performance gains resulting from minimum induced drag can be significant for such aircraft, designers have sought to devise methods capable of treating complex planforms more accurately and accounting for inviscid, non-planar effects (wing fences, end plates, engine pylons, etc.) in determining the lift distribution. The latter area has been of more than academic interest since the appearance of jet transports with pylon-mounted engines and boundary layer control fences. The computer program described by Lundry (Ref. 55) uses a transform technique to map the non-planar configuration into a type of lifting line and then computes the distribution of twist and camber necessary to minimize the induced drag.

Many other significant features of the aerodynamic characteristics of wings have been deduced using the lifting-line approach. This theory has its limits, however. Obviously it does not treat well the case where, because of sweep, there is a substantial spanwise flow component, nor is a single lifting line an adequate representation of a wing when the ratio of span to chord is not large. These deficiencies were recognized quite early, but because of the complexity of the generalization from lifting line to lifting surface and the use of high aspect ratio unswept wings until after World War II, solution techniques were long in developing. In 1950 Multhopp (Ref. 60) employed a generalization of the scheme above to make one of the first successful attacks on the problem.

The complexity one must contend with is easily seen in the expression for the local angle of attack (Ref. 43) at a point on the wing

$$\alpha(x_1,y_1) = -\frac{1}{8\pi} \fint_{-b/2}^{b/2} \frac{dy}{(y_1 - y)^2} \int_{x=x_{\ell.e.}}^{x=x_{t.e.}} \Delta C_p(x,y) \left[1 + \frac{x_1 - x}{\sqrt{(x_1-x)^2 + (y_1-y)^2}} \right] dx$$

where ΔC_p is the unknown load distribution and the bar through the integral sign denotes the principal value.

As in the lifting line theory, the unknown loading function is usually approximated by a series:

$$C_p(\xi,\eta) = \frac{2b}{c(\eta)} \sum_{r=0}^{N-1} a_r(\eta) h_r(X)$$

$$\text{with} \quad h_r(X) = \frac{2}{\pi} \frac{\cos[\frac{\psi}{2}(2r+1)]}{\sin(\psi/2)} \;; \quad \xi = \frac{2x}{b} \;; \quad \eta = \frac{2y}{b} \;; \quad X = \frac{x - x_{\ell.e.}}{c} = \tfrac{1}{2}(1 - \cos\psi);$$

$$\psi = \cos^{-1}(1 - 2X) \qquad\qquad \theta = \cos^{-1}(\eta)$$

$$a_r(\eta) = \frac{2}{m+1} \sum_{n=1}^{m} a_{rn} \sum_{\mu=1}^{m} \sin\mu\theta \sin\frac{\mu n\pi}{m+1}$$

This approach in effect replaces the unknown $\Delta C_p(x,y)$ by mN unknown coefficients A_{rn}. The problem is then to calculate A_{rn} by satisfying the boundary condition $\alpha(x_1,y_1)$ at suitable points distributed over the planform. The main numerical difficulty lies in determining the double integral due to each term in the respective loading.

Reference 43 discusses three methods of carrying out the integration which are of comparable accuracy and difficulty. The details of one of these, including the computer program (in Algol 60), is given in Reference 61. In the later work, the theory has been extended to treat slowly oscillating wings. Reference 41 makes some detailed appraisals of the accuracy of this and other methods developed in Europe. A similar approach for the non-oscillating case with the computer program given in FORTRAN is presented in Reference 39. Further details are discussed in Reference 44. A FORTRAN program of slightly different approach is given by Lamar in Reference 59. The effects on accuracy of certain assumptions for the form of the pressure distribution, the number of points at which the boundary conditions are satisfied, and the location of these points is discussed in Reference 38. Wagner, in Reference 47, gives a good summary of the present state of development of true lifting surface theory. He is particularly careful to distinguish this approach from the vortex lattice or other "finite element" approaches.

Because one seems compelled to employ a large system of simultaneous equations to approximate the lifting surface integral equation satisfactorily and because the choice of points at which the boundary condition is satisfied, the method of integration, and the complexity of the planform all effect to the accuracy of the results, investigators quickly began to search for alternate lifting surface methods. Although called by a variety of names, the most popular alternate is an extension of the idealized single horseshoe vortex representation of a wing given by Glauert (Ref. 16b). By dividing the wing surface into a finite number of flat rectangular panels, placing such a horseshoe vortex on each, and summing the contribution of all the vortices to the flow over a control point in each panel, one obtains a system of relatively simple equations which, when solved for the individual

vortex strengths, has been found to give remarkably good estimates of the pressure distribution over the wing. Reference 37 describes such an approach—called here a vortex lattice—and supplies a FORTRAN program for computing the lift and moment distribution and overall lift and moment characteristics of rather complex wings. This particular program can accept a maximum number of horseshoe vortices on the left side of the plane of symmetry of 120. Within this limit, the number of horseshoe vortices in any chordwise row may vary from 1 to 20 and the number of chordwise rows may vary from 1 to 50. It can treat wings with dihedral and/or sweep.

Reference 40 describes another perhaps more restricted computer-based approach while Reference 42 is a systematic mathematical study of the characteristics of the method and various solution techniques. Current work at the Boeing Company (Ref. 58) seems intended to reduce computation time and increase accuracy by using overlapping (both spanwise and chordwise) continuous distributions of vorticity over a set of panels on a paneled wing. The basic distributions are independent and each satisfies all the boundary conditions required of the final solution. Boundary conditions are satisfied in a least square error sense. Excellent results have been obtained thus far and consideration is now being given to including an automatic paneling routine in the program so that the user need only specify the wing geometry and the accuracy with which he wishes to calculate the downwash in order for the program to select, on an iterative basis, sufficient panels to satisfy this requirement.

All of these lifting surface and vortex-lattice theories suffer from twin faults: they are applicable only to planar wings, wings without thickness which lie entirely in the x-y plane, and they do not include the effects of viscosity. One interesting way of circumventing these problems for unswept, moderate-to-high aspect ratio wings is given in Reference 36. There, two dimensional data—obtained either from wind tunnel test or theoretical calculations which include the effects of thickness and viscosity—are extended to three-dimensions by using lifting-line theory to determine the effective local angle of attack at each point along the span, looking up the two-dimensional characteristics corresponding to that local angle of attack, and integrating the results in the spanwise direction. This method forms the basis of the discussion in the next chapter on extending the theory for predicting two-dimensional aerodynamic characteristics to treat complete wings.

TREATMENT OF VISCOUS EFFECTS: DRAG

Despite the fact that much of the aerodynamic behavior of an airplane can be deduced by considering air to be an inviscid fluid, one very important characteristic, its resistance to continued motion, arises directly from the viscosity of the air and necessitates the installation of a power plant and a store of fuel to operate that power plant. Unfortunately, adequate theoretical descriptions of this characteristic are very much more difficult to provide than are descriptions of the aircraft's lifting behavior. For this reason early designers relied almost exclusively on correlations of experimentally-determined drag with body shape and surface condition during the

preliminary design phase and on wind tunnel and flight tests of the actual
configuration during the final stages of development. This procedure is still
widely used. Reference 53 is an up-to-date compilation of the most widely
accepted correlations along with procedures for using them to estimate th
drag of complete subsonic aircraft. A similar approach is employed in
Reference 62.

Analytical determination of the drag of bodies evolved from the work of
Prandtl, who in 1904 proposed that the effects of viscosity could be considered
to be confined to a thin layer of fluid immediately adjacent to the body sur-
face (*i.e.*, a boundary layer). Such an assumption permits a considerable
simplification to be made in the equations describing fluid motion in this
region. Outside this region one can use the classical inviscid analysis.
Other investigators then began to develop methods for solving the boundary
layer equations, first for simple configurations such as flat plates and
later for curved two-dimensional bodies. One of the more versatile techniques
has been programmed for computer solution (Ref. 63). Although the technique
treats compressible flows with heat transfer, the flow will be taken here to
be incompressible and non-heat-conducting in order to describe the approach
as simply as possible.

If one begins with the conventional momentum integral equation (equation
44 in the next chapter),

$$\frac{d\theta}{dx} + \frac{du_e}{dx}\left(\frac{2\theta + \delta^*}{u_e}\right) = \frac{\nu}{u_e^2}\left(\frac{\partial u}{\partial y}\right)_{wall} \quad ,$$

and makes the following definitions

$$\ell \equiv \frac{\theta}{u_e}\left(\frac{\partial u}{\partial y}\right)_w \quad ,$$

$$-n \equiv \frac{\theta^2}{\nu}\frac{du_e}{dx} \quad ,$$

$$H \equiv \frac{\delta^*}{\theta} \quad ,$$

then this equation can be written

$$-u_e\frac{d}{dx}\left(\frac{n}{\frac{du_e}{dx}}\right) = 2[n(H + 2) + \ell] = N$$

Correlation of N against n for a number of exact theoretical solutions
of the boundary layer equations indicated to Cohen and Reshotko (Ref. 64)
that one could take

32

$$N = A + Bn$$

where A and B are constants for flows with zero or favorable pressure gradients. Under these circumstances the equation can be integrated to yield

$$n = -Au_e^{-B} \frac{du_e}{dx} \int_0^X u_e^{B-1} dx$$

A = 0.44 and B = 5.5. Then,

$$\theta = \left(\frac{-n\nu}{\frac{du_e}{dx}} \right)^{1/2} .$$

Since exact theoretical solutions of the boundary layer equations give a unique correlation between n and ℓ, this can be used to find $(\partial u/\partial y)_w$. Then one may use

$$A + Bn = 2 \left[n \left(\frac{\delta^*}{\theta} + 2 \right) + \ell \right]$$

to find δ^*.

The solution for the turbulent boundary layer case is obtained from a variant of the momentum integral analysis with experimental skin friction correlations. Transition is determined from a variant of the Schlichting Ulrich (sixth order polynominal representation of the velocity distribution) laminar boundary layer stability analysis.

Comparisons between predicted and measured values of δ^* and θ on an NACA 0012 airfoil were quite good, except in the immediate neighborhood of the transition point. It is to be expected, therefore, that predicted values of skin friction drag would also be quite good. To find the total drag on the airfoil, however, it would be necessary to find the change in the pressure distribution over the airfoil resulting from the presence of the boundary layer displacement thickness and add this to the skin friction drag or integrate completely across the wake to find the overall change in the momentum of the flow caused by the passage of the airfoil. Schlichting (Ref. 65) describes a method, based on the latter idea, which was developed by Squire and Young in 1938. The drag force per unit length of span represented by the momentum defect in the wake far downstream of the airfoil is

$$\frac{D}{b} = \int_{y=-\infty}^{y=+\infty} \rho u \, (u_\infty - u) \, dy$$

from which

$$C_D = \frac{D}{b \, \frac{\rho}{2} \, cu_\infty^2} = \frac{2}{c} \int_{y=-\infty}^{y=+\infty} \frac{u}{u_\infty} \left(1 - \frac{u}{u_\infty} \right) dy = \frac{2\theta_\infty}{c} .$$

The problem, then, is to evaluate θ_∞ in terms of the boundary layer characteristics over the airfoil.

The momentum integral equation of boundary layer theory is of course also valid in the wake behind a body. In the wake, however, the shearing stress at the wall is zero so that the equation for this circumstance becomes

$$\frac{d\theta}{dx} + \frac{du_e}{dx}\left(\frac{2\theta + \delta^*}{u_e}\right) = 0 \ .$$

x now denotes the distance from the trailing edge of the body measured along the wake centerline. The foregoing equation can also be written as

$$\frac{1}{\theta}\frac{d\theta}{dx} = -\left(\frac{2 + H}{u_e}\right)\frac{du_e}{dx} = -\left(2 + H\right)\frac{u_\infty}{u_e}\frac{d(u_e/u_\infty)}{dx}$$

$$= -\left(2 + H\right)\frac{d}{dx}\left(\ln \frac{u_e}{u_\infty}\right) \ .$$

Integration by parts of this equation along x from the trailing edge of the body (station 1) to a station very far downstream in the wake results in

$$\ln \theta \Big|_\infty^1 = -(H + 2)\ln \frac{u_e}{u_\infty}\Big|_\infty^1 + \int_\infty^1 \ln \frac{u_e}{u_\infty}\frac{dH}{dx}\ dx \ .$$

At the downstream station $u_e = u_\infty$ and

$$\frac{\delta^*}{\theta} = \frac{\displaystyle\int\left(1 - \frac{u_e}{u_\infty}\right)dy}{\displaystyle\int\frac{u_e}{u_\infty}\left(1 - \frac{u_e}{u_\infty}\right)dy} = 1$$

thus,

$$\ln (\theta_1/\theta_\infty) + (H_1 + 2)\ln (u_{e_1}/u_\infty) = \int_{H=1}^{H=H_1} \ln \frac{u_e}{u_\infty}\ dH$$

or

$$\theta_\infty = \theta_1 \ln\left(\frac{u_{e_1}}{u_\infty}\right)^{H_1 + 2} \exp\left(\int_1^{H_1} \ln \frac{u_\infty}{u_e}\ dH\right) \ .$$

Now, if the integral on the right hand side and u_{e_1}/u_∞ can be evaluated, one has the required explicit relationship between θ_∞ and θ_1. From an analysis of experimental data H. B. Squire proposed that one could assume

$$\frac{\ln (u_\infty/u_e)}{H - 1} = \frac{\ln (u_\infty/u_{e_1})}{H_1 - 1} = \text{constant} \ .$$

34

Hence,

$$\ln (u_\infty/u_e) = \text{constant } (H - 1)$$

$$\int_1^{H_1} \ln \frac{u_\infty}{u_e} \, dH = \int_1^{H_1} (H - 1) \left[\frac{\ln (u_\infty/u_e)}{H_1 - 1} \right] dH = \frac{\ln (u_\infty/u_{e_1})}{H_1 - 1} \left(\frac{H^2}{2} - H \right)_1^{H_1}$$

$$= \left(\ln \frac{u_\infty}{u_{e_1}} \right) \left(\frac{H_1 - 1}{2} \right) \quad ;$$

with this result

$$\theta_\infty = \theta_1 \left(\frac{u_{e_1}}{u_\infty} \right)^{H_1 + 2} \left(\frac{u_\infty}{u_{e_1}} \right)^{\frac{H_1 - 1}{2}} = \theta_1 \left(\frac{u_{e_1}}{u_\infty} \right)^{\frac{H_1 + 5}{2}} .$$

A typical value of H_1 is 1.4. For this value

$$\theta_\infty = \theta_1 \left(\frac{u_{e_1}}{u_\infty} \right)^{3.2}$$

and thus

$$C_D = \frac{2\theta_1}{c} \left(\frac{u_{e_1}}{u_\infty} \right)^{3.2} .$$

To find C_D, then, one must know the value of the potential velocity (velocity outside the boundary layer) at the trailing edge and the value of the momentum thickness at the same place. One attempt to be more explicit is reported by Schlichting. Using relations for a flat plate H. B. Helmbold obtained

$$C_D = \frac{0.074}{R_e^{\frac{1}{2}}} \left\{ \int_{x_u/c}^1 \left(\frac{u_e}{u_\infty} \right)^{3.5} d\left(\frac{x}{c} \right) + 62.5 \, R_e^{\frac{1}{4}} \left(\frac{\theta_t}{c} \right)^{5/4} \left(\frac{u_{e_t}}{u_\infty} \right)^{3.75} \right\}^{0.8}$$

for the drag due to the one surface of a wing. The subscript "t" refers to the point of transition from laminar to turbulent flow.

The method of Squire and Young can be extended fairly simply to axisymmetric bodies and was so done by Young in 1939. (ARC R&M 1947)

Cebeci, Mosinskis, and Smith (Ref. 54) studied the possibility of improving the estimation of the drag of two-dimensional and axisymmetric bodies by improving the laminar and turbulent boundary layer methods used as inputs to the Squire and Young method. They also sought the effect of better identification of the location of the transition region. They concluded that the total drag coefficients of two-dimensional bodies such as airfoils can with such improved techniques be calculated very accurately for $\alpha \leq 6°$. For higher angles of attack "use of the Squire-Young formula introduces an error into the drag calculations. . .since the Squire-Young formula is applicable only to a symmetrical wake." It would seem, however, that this

restriction could be removed without excessive difficulty. They also found that by improving the theoretical turbulent boundary layer methods they could match 57 experimental values with an rms error of 2.9%

The total-drag coefficient of axisymmetric bodies, they found, can be calculated less accurately than the total drag coefficient of two-dimensional bodies. "The calculations show a great sensitivity to the choice of tail end location on the body and to the use of inviscid pressure distributions in the drag calculations." For more general three-dimensional bodies such as aircraft fuselages there are unfortunately no quasi-rigorous analytical methods now available and one must resort to techniques based on rather gross approximations or to correlations of experimental results.

The reader has no doubt observed by this time that a detailed discussion of ways to calculate the drag of bodies ultimately comes to a consideration of methods for solving the boundary layer equations. Even the simplest case of steady, two-dimensional, incompressible, laminar flow involves a non-linear partial differential equation for which a general closed-form solution is impossible. This is the reason for the proliferation of solution techniques one sees in the literature. Some of these involve a great deal of insight into the problem and others employ rather sophisticated mathematical techniques. For these reasons it seems appropriate not to discuss the various methods in detail here but rather to direct the reader to Reference 65 which is probably the best single source of information on the rationale behind the various solution techniques.

FUSELAGE CONTRIBUTIONS TO LIFT, DRAG, AND MOMENT

The isolated fuselage is generally a body with a plane of symmetry rather than an axis of symmetry. This situation effectively precludes accurate calculations of its lift and drag by relatively simple, closed-form methods. Thus, until recently, it was the practice to rely on the guidance provided by a few classical approximate theoretical treatments and determine the detailed lift and drag characteristics experimentally.

Sir Horace Lamb (Ref. 6) for example was able to find an exact expression for the potential about an ellipsoid with three unequal axes. The problem was treated somewhat more completely by Munk (Ref. 66) who was interested in its application to determining the aerodynamic characteristics of airship hulls. Timman (Ref. 49) extended the analysis to include flows with velocity components along two axes simultaneously. He then calculated the streamline patterns for such a case. From these one can get the inviscid pressure distribution over the surface. This could then serve as the basis of a boundary layer calculation. Timman in fact had previously developed a boundary layer computation method for such a body and Reference 49 was intended to supply the potential field needed to begin the calculation.

A 1941 paper by Hans Multhopp (Ref. 51) provided a quantum jump in theoretical understanding of the fuselage contribution to airframe lift and moment. The essence of his arguments are contained in the following exerpts taken from a translation of that paper.

36

One notoriously neglected phase in the aerodynamics of aircraft is that of the fuselage. This is due, in the first instance, to the fact that the fuselage considered by itself is a comparatively simple structure the effects of which are apparently readily perceived. But its real effects come into evidence only in combination with other parts of the aircraft, especially with the wing; hence it becomes necessary to evolve a fuselage theory which includes this mutual interference.

The search for mathematically exact solutions for such interference problems is exceedingly bothersome throughout, as it would entail the development of a three-dimensional potential theory with very arbitrary boundary coditions; a problem to which hardly more than a few proofs of existance could be adduced.

****

For the present task the performance mechanics are, in general, excluded, since drag problems usually must be left to experimental research.

****

Before proceeding to the analysis of the interference of the fuselage with the other parts of the airplane, a brief discussion of the phenomena observed on the fuselage, in the absence of all other airplane parts, is necessary.

On analyzing the conditions in frictionless parallel flow the first result is the total absence of resultant forces on the fuselage; the pressure distribution over the body merely affords free moments. These free moments are of some significance since they are proportional to the angle of attack of the fuselage and hence enter the stability quantities. The sign of these moments is such that the stability about the normal axis is lowered by the action of these free moments. On an axially symmetrical fuselage the free moment in flow along the fuselage axis is, of course, zero; on unsymmetrical fuselage forms or by appendages the axis for zero moment can be located at any other place. The free moment is produced by negative pressure on the upper side of the bow and on the lower side of the stern and positive pressure at the lower side of the bow and on the upper side of the stern. (See the figure at the top of the next page.)

The free moments can be computed in various ways. If time and patience are no object, a field of singularities substituting for the fuselage may be built up by means of potential theory methods.* But for the task in hand Munk's method is much more suitable. He simply determined the asympototic value for very slender fuselage forms and then added a correction factor dependent on the slenderness ratio, which he obtained by a comparison with the values of easily and accurately computable forms.

---

*   This is, in fact, just what is done in Reference 23; the computer makes it possible to be impatient and still accomplish the task quickly and accurately.
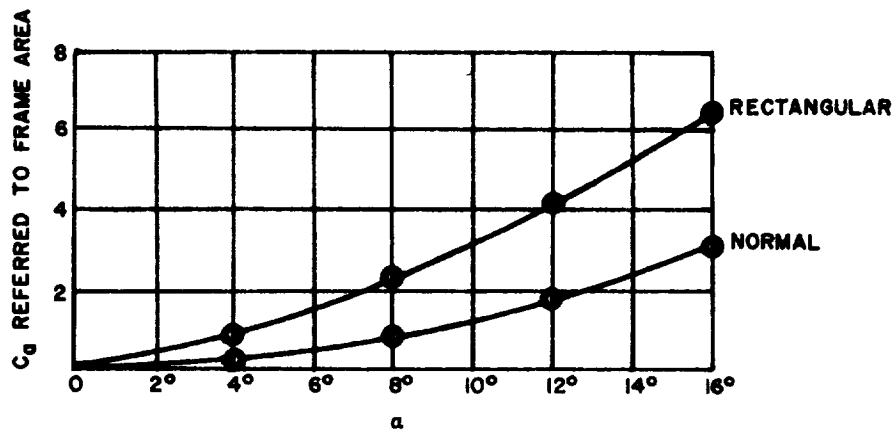
According to Munk, the unstable moment of a very slender body of revolution is

$$\frac{1}{q} \frac{dM}{d\alpha} = -2 \text{ vol.} \text{ ,}$$

the effect of finite fuselage length being accounted for by the correction factor $(K_2 - K_1)$ which depends on the slenderness ratio ℓ/D. (See the accompanying sketch.) In this representation K_2 is the air volume in ratio
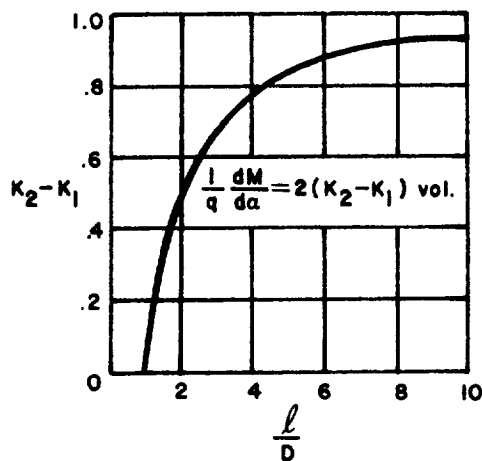
to the fuselage volume by transverse motion of the fuselage and K_1 *that by longitudinal fuselage motion. For other than axially symmetrical surfaces it is*

$$\frac{1}{q}\frac{dM_R}{d\alpha} = -\frac{\pi}{2}(K_2 - K_1)\int_0^{\ell} b_R^2\, dx.\ {}^{**}$$

This is all that the consideration of the potential theory supplies concerning a single fuselage. But the actual behavior of the fuselage is not described by the potential flow alone. As soon as the flow past the fuselage ceases to be perfectly symmetrical, boundary-layer material accumulates more on one side than on the other and the flow conditions are altered. This results in additional forces at the fuselage and so becomes an appreciable factor in the moment balance of the fuselage. The point of application of the induced frictional lift or cross force is of course proportionally far aft at the fuselage.

As the dependence of frictional lift on angle of attack is strongly suggestive of a very similar course on wings of very small aspect ratio, its correlation suggests itself. For a wing of very small aspect ratio we get approximately for $\alpha = 0$:

$$\frac{1}{q}\frac{dA}{d\alpha} = \frac{\pi b^2}{2}\ {}^{***},$$

---

*   The air volume depends upon the local air velocity which is found from an exact solution to the flow over an ellipsoid. (See Reference 66, Division C.) The formula for computing K_1 is given (Ref. 66) as

$$K_1 = \frac{2\left(\dfrac{1-e^2}{e^3}\right)\left[\frac{1}{2}\log\dfrac{1+e}{1-e} - e\right]}{2 - 2\left(\dfrac{1-e^2}{e^3}\right)\left[\frac{1}{2}\log\dfrac{1+e}{1-e} - e\right]}$$

where $e = \sqrt{1 - (a^2/b^2)}$ and a = the half axis and b = the largest radius of the ellipsoid.

**   The notation is that of the original paper. While somewhat different from present usage in the United States, its meaning, when taken in context, should be reasonably clear.

***   Assume a circular streamtube with diameter b equal to the wing span. The lifting force which the fluid applies to the wing results in a deflection of the streamtube according to Newton's Second Law of Motion:

$$L = \left[\rho V\left(\frac{\pi b^2}{4}\right)\right] V \sin \varepsilon$$

where ε is the deflection angle of the streamtube. For small aspect ratios the wing chord is comparatively long. Thus the flow inclination seen by most of the wing is ε. Hence for such situations one can take

a result readily derivable by means of certain momentum considerations which is in good agreement with the available test data for such wings. However, the conventional fuselage has no sharp sides; hence a temporarily unknown measure that denotes the width of the separating boundary layer substitutes for the width b. In place of it we correlate the lift to the maximum fuselage width b_R and introduce a form factor f, the exact determination of which might be a profitable field of experimental research; presumably it depends, above everything else, on the cross-sectional form of the fuselage, on its solidity, and on the location of appendages. Hence we put

$$\frac{1}{q} \frac{dA_R}{d\alpha} = \frac{\pi}{2} f b_R^2 .$$

The foregoing appraisal of the moments of the fuselage in free stream fails, because the flow pattern of the wings causes a very substantial variation of the flow at the fuselage. To begin with, the previously described frictional lift of the fuselage is not likely to exist, since the wing orientates the flow along the wing chord and even far aft of it with the result that no appreciable flow component transverse to the fuselage exists in the real zone of formation of the frictional lift. Hence there is some justification in assuming that the theoretically anticipated moments will afterward actually occur.

First of all the fuselage with wing differs from the fuselage alone in that the fuselage takes up a very substantial proportion of the lifting

---

***continued $\qquad\qquad\qquad$ $\alpha \approx \epsilon$.

For small angles of attack the lift may then be written

$$L = \frac{\pi}{2} b^2 \frac{\rho V^2}{2} \alpha$$

from which

$$\frac{1}{q} \frac{dL}{d\alpha} = \frac{\pi}{2} b^2 .$$

Note that Multhopp uses the symbol A to designate lift. This result can also be obtained from the general expression for the dependence of lift-curve slope on aspect ratio (sweep angle and Mach number = zero):

$$C_{L_\alpha} = \frac{2\pi AR}{2 + \sqrt{4 + AR^2}}$$

when AR→0 this expression becomes

$$C_{L_\alpha} = \frac{2\pi AR}{4} = \frac{2\pi b^2}{4S} = \frac{\pi}{2} \frac{b^2}{S} .$$

Then $\qquad\qquad\qquad$ $L = C_{L_\alpha} \alpha q S$

and $\qquad\qquad\qquad$ $\frac{1}{q} \frac{dL}{d\alpha} = \frac{\pi}{2} b^2 .$

40

forces ordinarily carried by the wing section in its place. The point of application of the aerodynamic forces at the fuselage directly due to the circulation of the wing, is located at the same place as on the substitute wing section; separating this air force distribution for the moment leaves only a free moment which is solved from a simple momentum consideration.

Next, the fuselage is assumed to be sufficiently long, so that, after fixing a reference plane at right angles to the flow direction that meets the fuselage at distance x from the nose, the integral over the pressure distribution of the fuselage portion ahead of the reference plane equals the vertical momentum passing through this area in unit time. Then, with β^* as the angle in yaw in the reference plane, that is, the angle which the flow would form with the fuselage axis if the fuselage were non-existent, and b_R as the fuselage width at this point, the lift of the thus segregated fuselage portion is:

$$\int_0^x \frac{dA_R}{dx}\, dx = \rho V^2 \beta\, \frac{\pi}{4}\, b_R^2 \ .$$

For, if the fuselage is long enough, the flow at right angles to the fuselage axis may be approximated to two-dimensional, and for the flow at right angles to an elliptic cylinder the comprised air volume, that is, the integral

$$\int \rho (v_n - v_{n_\infty})\, df = \rho v_{n_\infty}\, \frac{\pi}{4}\, b_R^2 = \rho v \beta\, \frac{\pi}{4}\, b_R^2$$

is (v_n and v_{n_∞} being the components at right angles to the cylinder axis) independent of the axes ratio of the ellipse. (Note the sketch below.)
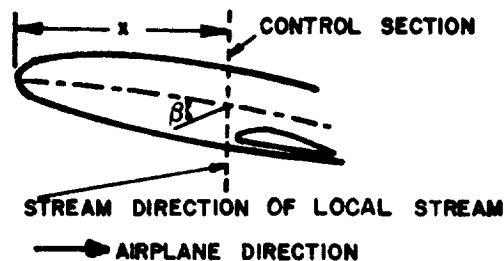
Since this formula holds true even for a cylinder degenerated to a flat plate, its approximate use for all cross-section forms appears justified.

---

* β is the angle the flow direction makes with the fuselage axis. If these vectors lie in the x-z plane then β is the angle of attack.

Differentiation with respect to x then affords:

$$\frac{1}{q}\frac{dA}{dx} = \frac{\pi}{2}\frac{d}{dx}\left(\beta\, b_R^2\right)$$

(The air load distribution along the typical fuselage is shown in the figure below)

■ **FROM FUSELAGE MOMENTS**

□ **FROM FUSELAGE LIFT**

-- **TOTAL VARIATION**
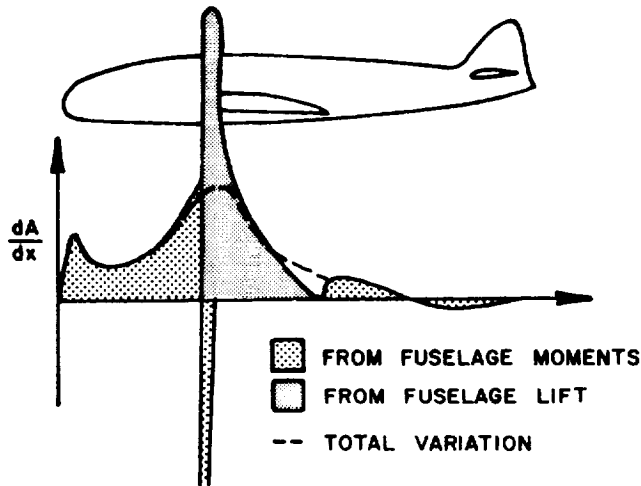
By reason of the disappearance of b_R *the so computed total fuselage lift is zero at both its ends, hence gives the desired free moments additionally supplied by the fuselage. This free moment is for any reference point*

$$\frac{M_R}{q} = \frac{\pi}{2}\int_0^\ell \frac{d}{dx}\left(\beta\, b_R^2\right)x\,dx$$

and, after partial integration:

$$\frac{M_R}{q} = -\frac{\pi}{2}\int_0^\ell \beta\, b_R^2\,dx$$

For surfaces of revolution on which the flow is not disturbed by the presence of the wing, we get, because β = *constant*

$$\frac{M_R}{q\beta} = -\frac{\pi}{2}\int_0^\ell D^2\,dx = -2\,vol.$$

42

or the same result as Munk's for the free moments of airship hulls. It then might be advisable to apply a correction factor to these free fuselage moments on Munk's pattern, containing the effect of the finite fuselage length, except for the difficulty of not quite knowing what slenderness ratio to apply. The reduction relative to the theoretical value is primarily due to the fact that the flow at the fuselage ends still varies somewhat from the assumed two-dimensional pattern; and while the rear end contributes almost nothing to the free fuselage moment, the portions of the fuselage directly before the wing, which certainly are not encompassed by this reduction through the effect of the finite length, contribute very large amounts. Hence the actual value for the correction factor is likely to be far closer to 1 than Munk's quantity $(K_2 - K_1)$.

The presence of the wing is allowed for by relating β *to the wing circulation. The change of the moment with the angle of attack is:*

$$\frac{1}{q}\frac{dM_R}{d\alpha} = -\frac{\pi}{2}\int_0^{\ell} b_R^2 \frac{d\beta}{d\alpha}\, dx \ .$$

The change of the yawed flow with the angle of attack $d\beta/d\alpha$ *is expressed as follows: The flow in the region of the wing is practically parallel to the wing chord; hence* $d\beta/d\alpha = 0$. *Behind the wing the downwash reduces the yawed flow; at the fuselage stern in the vicinity of stabilizer and elevator there is obtained:*

$$\frac{d\beta}{d\alpha} = 1 - \frac{d\alpha_W}{d\alpha}\ .$$

(This is depicted graphically in the accompanying sketch.)
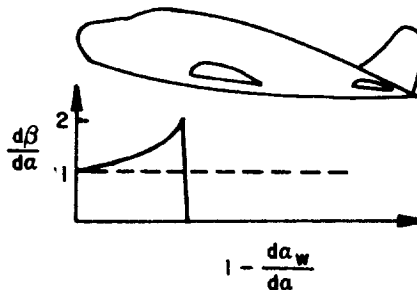
It is sufficiently exact, when assuming that $d\beta/d\alpha$ *rises linearly from the wing trailing edge to this value. Before the wing* $d\beta/d\alpha$ *is always greater than 1 because of the prevalent upwash.* (This variation is shown in the graph below.)

$$\frac{\overline{d\beta}}{d\alpha} = \frac{1}{x} \int_0^x \frac{d\beta}{d\alpha} dx$$

$$\frac{d\beta}{d\alpha}$$

$$\frac{x}{t}$$
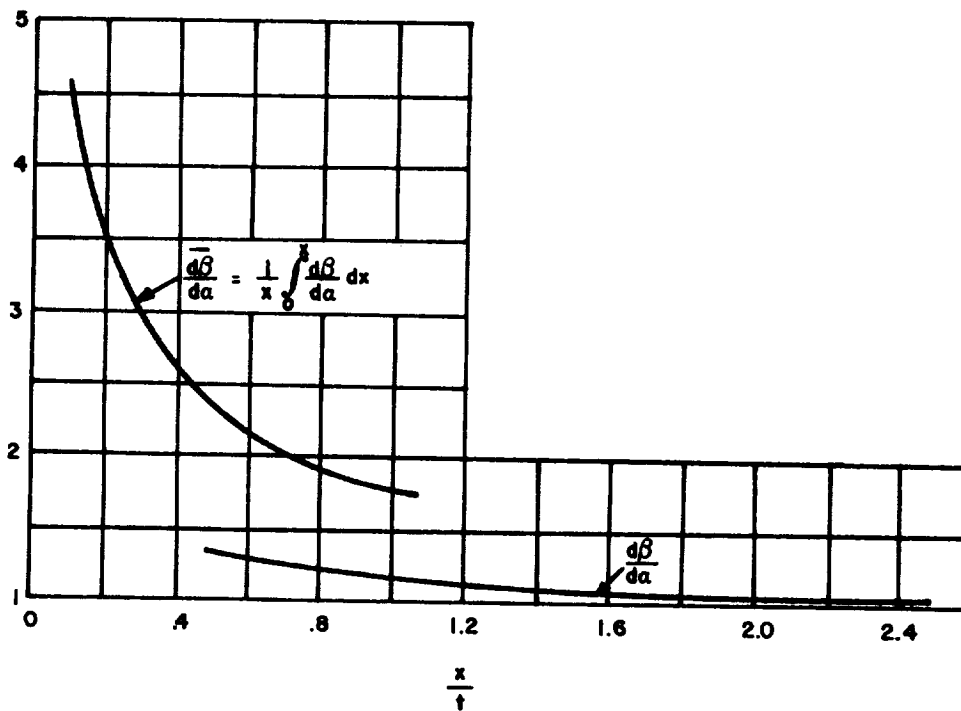
For the determination of the fuselage effect on the lift distribution of the wing the flow transverse to the fuselage was assumed to be two-dimensional; then all the mathematical difficulties which the fuselage of itself would entail, can be removed by a conformal transformation of the fuselage cross section to a vertical slit. Then the calculation of the lift distribution for a wing-fuselage combination reduces to that of an equivalent wing, wherein the fuselage effect is represented by a change in chord distribution and also, to some extent, in the angle-of-attack distribution. Then the conventional methods of computing the lift distribution of a wing are fully applicable. Multhopp's transform forms the basis of the method discussed in detail in the next chapter for extending two-dimensional airfoil characteristics to three-dimensional wings. For that reason it will not be discussed further here.

As mentioned earlier, modern treatments of the fuselage contribution to airplane lift, drag, and moments usually represent the flow displacement caused by the presence of the fuselage through source distributions either on the fuselage surface (Ref. 23) or along the fuselage axis (Ref. 48). The three-dimensional source distribution (Ref. 23) can of course be extended to include the wing, etc. A means to account for lift, such as

44

a distribution of horseshoe vortices, must then be added and, finally, a consistent boundary layer computation must be included in order to complete the calculation. This is, all told, a very complex procedure. Some details of these treatments are given in the next chapter.

It should be noted that the treatment of the fuselage as a body surrounded by a boundary layer is tenable only so long as the angle of attack does not get very large. Reference 68 points out that when a thin cone exceeds an angle of attack of 8° or so vortices begin to be shed from the edges in the streamwise direction resulting in a larger-than-expected normal force coefficient.

Cebeci, Mosinskis, and Smith (Ref. 109) obtained some interesting results from their efforts to predict the drag coefficients of axisymmetric bodies. Their procedure took the following course:

1. Calculate the inviscid pressure distribution on the body using a distribution of sources technique (Ref. 23).

2. Consider the aft end of the body to occur at that longitudinal station where the pressure coefficient first returns to zero. Note that the inviscid source distribution method will always yield a stagnation point at the aft end. The point at which $C_p = 0$ is therefore located somewhat upstream. For the fineness ratio 4-10 bodies which they studied, this point occurs at roughly 90% of the length. Bodies with blunt trailing edges such as ellipsoids and airship hulls were found experimentally to have separated boundary layers aft of the 85%-90% point so that the assumption of $C_p = 0$ in this region is fairly reasonable. Some boundary layer separation also occurs on bodies which taper slowly to a point because the boundary layer cannot withstand a pressure rise to a stagnation value. Again, the assumption of separation at about the 90% or 95% point is probably reasonable although the ratio of boundary layer thickness to body radius at separation (used as indicated below) is different than for bodies with blunter trailing edges. Data presented in the paper indicates a pressure coefficient of about +0.1 in the separated flow region.

3. Calculate the boundary layer displacement thickness and momentum thickness at the point taken to be the trailing edge. When the body radius is expected to be large compared with the displacement thickness, a two-dimensional value for the momentum thickness is found to give better results.

4. Then use Granville's formula (Ref. 111)

$$C_D = \frac{4 r_o \theta_{2-D}}{R_o^2} \left(\frac{u_e}{u_\infty} \right)^{[7(\frac{\delta^*}{\theta} + 2) + 3]/8}$$

45

to determine the total drag of the body. In this formula r_o is the radius of the body at the tail "end", R_o is the maximum radius, u_e is the flow velocity outside the boundary layer at the tail end of the body, and u_∞ is the free stream velocity. If the "end" is chosen to be that point at which $C_p = 0$, then $u_e/u_\infty = 1.0$.

For a fineness ratio 4.0 body (similar to an ellipsoid but having the tail end modified to come to a point) the method gave excellent agreement with experimental results up to a Reynolds Number of about 6 million. Above this value, the prediction was about 20% low. The skin friction drag constituted about 90% of the total drag in this case. For the fineness ratio 7.0 body (same contour) the agreement with experiment was excellent for all Reynolds Numbers and the skin friction consititutes virtually all the drag.

The airship hull (fineness ratio = 4.2) predictions agreed very well with experimental results for all Reynolds numbers. Skin friction was again about 90% of the total drag and flow separation occurred at about 90% of the length.

Cebeci, Mosinskis, and Smith comment that their studies showed a great sensitivity to the effective end location on the body to be used in the drag formulas and to the inviscid pressure distribution used in the drag calculations. They note also that the two-dimensional analog of Granville's formula really applies only to the case where the wake is symmetrical with respect to the body chord. This condition is not satisfied for $\alpha > 6°$. Thus, a similar situation can be expected to prevail for axisymmetric bodies.

It would seem reasonable to expect that one might someday develop a version of Granville's formula for asymmetric bodies more representative of aircraft fuselages. To do this, however, it will be necessary to solve the three-dimensional boundary layer equations for the body in question and to develop a proper averaging method for final average values of θ_{2-D}, r_o, R_o, u_e/u_∞, and δ^*/θ. Such bodies also require more careful scrutiny for the presence of and locations of flow separations. In a recent paper Cebeci, Mosinskis, and Smith (Ref. 109) address this problem for <u>axisymmetric</u> bodies. Using a finite difference method of numerically solving the axisymmetric boundary layer equations which included tranverse curvature effects, they were able to predict the location of separation with less than 1% error. Their two-dimensional calculations also gave excellent results for separation on airfoils at high angles of attack. It should, perhaps, be pointed out at this point that finite difference techniques for solving the boundary layer equations take on the order of twenty times the computer time that are required for the less accurate momentum integral technique described in some detail in the next chapter.

A very recent analytical effort to describe the drag-producing separated flow behind bodies of revolution is described by Marshall and Deffenbaugh (Ref. 117). They first treat three-dimensional steady separation as equivalent to two-dimensional unsteady flow. This assumption, a heuristic one, is suggested primarily by experimental data. They then describe the two-dimensional unsteady wake by a distribution of inviscid point vortices superimposed on the un-

46

separated potential flow solution, suitably modified by diffusive effects. Their
argument for following this tack is the way the wake is observed to develop with
time. The object of this approach is to avoid solving the complete Navier-Stokes
equations for the separated flow field by a laborious and lengthy finite-dif-
ference technique. Instead, one pieces together by computer an ensemble of
relatively well-known solutions for sub-flow fields which together form the whole.
This is essentially the approach followed by the present work although it dif-
fers in details. Marshall and Deffenbaugh include a computer program listing
and user instructions in their report. They also compared results obtained by
their method with experimental data. For a prolate spheroid C_n agreed with
experiment for $\alpha \le 10°$. C_m agreed well for $\alpha \le 20°$. No drag computations were
presented, however.

INTERFERENCE EFFECTS

The flow at the junction of a wing and a fuselage is not that which
one would obtain by combining the flows about isolated wings and fuselages.
The flow about the wing modifies or influences the fuselage flow and vice
versa. Hence the name interference effects. Küchemann and Weber (Ref. 64)
provide two relatively simple means of determining the effect of the junction
on the inviscid pressure distribution: (a) Ring vortices are placed on the
surface of the body and their strengths are determined from the condition
that their induced radial velocities are proportional to the slope of the
wing-body junction in the streamwise direction; (b) A fictitious body,
obtained by subtracting the wing thickness inside the body from the given
body thickness, is considered and is replaced by a source distribution
along the body centerline. Applied with care to situations which the models
represent reasonably well, the methods give results for the interference
velocities which agree well with experiment.

With the advent of the computer, computation of the three-dimensional
potential field can be carried out on a fairly routine basis. Loeve
(Ref. 50) describes a computer program for the calculation of subsonic flow
about wing-body combinations. A three-dimensional distribution of sources
on the surface of the wing and the body is used to represent the disturbance
to the free stream caused by the presence of the non-lifting wing-body
combination. The source strengths are so adjusted that the flow is always
everywhere parallel to the surface of the wing-body. To treat the effects
of lift, a system of horseshoe vortices is placed on the camber line of
the wing.

The superposition of the flow due to sources located on the surface
of a wing-body combination is also discussed by Hess and Faulkner (Ref. 46)
who give some examples obtained through the aid of a computer program.

Success at treating the wing-body combination would naturally lead one
to attempt a method for the determination of the inviscid pressure distribution
on complete aircraft. The problem becomes tractable by considering the
aircraft surface to consist of a finite number of panels (hence the name
sometimes applied: finite element technique). As reported by Carmichel
(Ref. 48), the computer program being used at the NASA/AMES Center represents

(a) body thickness by line sources,

(b) body lift by line doublets,

(c) wing thickness by constant source panels,

(d) wing lift by constant pressure panels, and

(e) wing-body inteference by constant pressure panels.

These finite element methods can be considered a "brute force" approach, in that if one is willing to expend the computer time, the accuracy of the computation can generally be improved by using more and smaller elements. Ultimately, limitations in machine accuracy and in the numerical methods used provide a bound for the accuracy which can be obtained.

A recent correlation of wing-body lift interference effects in provided by Reference 67. Five effects are noted: (1) body upwash on the local angle of the wing; (2) local body flow parameters such as dynamic pressure on the wing characteristics; (3) lift carry-over from the wing to the body; (4) wing upwash on the body ahead of the wing; (5) wing lifting vortices on the body behind the wing. The correlation provided in the paper suggests that one may find the lift on a wing-body combination as follows: (1) Find the lift-curve slope of the wing from

$$C_{L_\alpha} = \frac{2\pi AR}{2 + \sqrt{4 + AR^2\beta^2(1 + \tan^2 \Delta/\beta^2)}}$$

where $\beta^2 = 1 - M^2$ and Δ is the sweep angle of the maximum thickness point on the wing. Aspect ratio here is based on exposed wing area, *i.e.* that part away from the fuselage. C_L is then based on the same area. (2) Multiply C_{L_α} thus found by an interference factor, F, given graphically in the paper. The present authors have determined that for M < 0.8 this curve can be fit by the equation

$$F = 1 + 2\left(\frac{d}{b}\right) + 24\left(\frac{d}{b}\right)^4$$

where d = body diameter and b = wing span. Working through the numbers shows that most of the effect is lift carry over from the wing. Most of the remainder can be accounted for with Multhopp's transform. (See next section.)

The fact that viscous effects have been excluded from all of these computations tends to compromise their accuracy or applicability somewhat. To use them successfully, one must first be sure that no flow separations are present such as at wing-body junctions.

In addition to the effect on lift, wing-body interference also complicate the estimation of drag since the boundary layer flow around wing-body junctions, for example, is subject to an external pressure distribution which is the resultant of all the effects listed above. It is not surprising then

48

that interference drag at subsonic speeds is usually evaluated experimentally or from correlations of data on similar configurations. A "brute force" means of accounting for viscosity in evaluating interference effects, however, would seem to follow from the methods used to calculate the viscous pressure distributions on airfoils (Ref. 31).

UNIFIED ANALYTICAL TREATMENTS OF WING-BODY CHARACTERISTICS

By 1962 (Ref. 83) the potential flow about a three-dimensional, non-lifting body had been determined quite successfully through the expedient of representing the surface by an ensemble of connecting plane quadrilaterals, placing on each a source of undetermined strength, and then finding the source strengths by requiring that the total flow produced by the interaction of all the sources and the free stream be parallel to a point on each of the quadrilaterals. A rather large number of "panels" was found to be necessary to obtain results which agreed well with closed form analytical solutions for simple bodies. In addition one had to keep the areas of all panels nearly the same. Despite the very significant amount of computer time required for even the simplest of such computations, the prospect of being able to extend the scheme to include wings, other protuberances and/or lift and thereby treat entire practical configurations at one time was unusually attractive to industry researchers. As a result there are now a number of numerical methods and associated computer programs for treating all or part of the wing-body problem. Among these one may cite References 84, 85, 86, 87, 88, and 89. The differences among the various methods cited lie principally in the numerical procedures employed. Indeed, since these bear so heavily on the economic practicality of the methods, it is not surprising to find a paper (Ref. 93) dealing entirely with this facet of the prediction procedure. Finally, the review article by Widnall (Ref. 90) is quite helpful in distinguishing among the approaches to these and other aspects of the problem used by 19 other authors.

The use of such calculation procedures to examine other aerodynamic characteristics (stability derivatives for example) of complete configurations is discussed in References 91 and 92. Here also the procedures are useful for determining only those contributions to the parameter values which do not depend significantly upon viscous effects. The ability to integrate the determination of these effects into the procedures in a rational yet computationally-manageable way seems to be a skill yet to be learned.

A THEORY FOR THE PREDICTION OF LIFT, DRAG, AND PITCHING

MOMENT OF LIGHT AIRCRAFT WINGS

INTRODUCTION

One of the fundamental tasks in aircraft design is the estimation of the forces which the air will exert on the vehicle during flight so that the structure may be made appropriately strong, the wings may be made sufficiently large to carry the desired load, and the engine may have sufficient power to propel the vehicle at the desired speed. Because no two masses may occupy the same space at the same time, the airplane must move air out of its way temporarily as it flies. To move air one must exert a force on it. Even the mere "sliding" of the air over the skin of the aircraft requires that one exert a force to overcome the friction.

To represent these phenomena quantitatively we consider a fictitious cube through which the air is moving. The cube is fixed to the airplane. Now in general, the velocity of the air moving across one face of the cube does not have to be the same as that moving across the opposite face. Thus we say that in any one of the three principal directions there is a net flux of mass across the cube boundaries given by

$$\rho \frac{\partial u_i}{\partial x_i}$$

where ρ represents the density of the fluid; u, the velocity; x, the length of the cube; and i indicates that it applies to any one of the three principal directions.

We assert that for the purpose of this analysis the volume occupied by a given mass of fluid always stays the same. As a result, if more net mass flows into the cube in one direction more must flow out in another direction. A mathematical statement of this concept is

$$\rho \frac{\partial u}{\partial x} + \rho \frac{\partial v}{\partial y} + \rho \frac{\partial w}{\partial z} = 0 \quad . \tag{1}$$

Newton's Second Law of Motion states in effect that the force required to change the direction of a mass, *i.e.* move it out of the way, is equal to the product of its mass and its acceleration. This force can be assumed to have three components, *i.e.* one along each principal axis, so that we should write three mathematical statements or equations to describe the complete picture:

$$F_1 = ma_1$$

$$F_2 = ma_2 \tag{2}$$

$$F_3 = ma_3$$

Now, acceleration is a time rate of change of velocity. But if the velocity crossing one face of the cube is different from that crossing an opposite face _at the same time_ there has been a change in velocity with distance. The product of this change with distance and the velocity at any point in the cube is also an acceleration. For acceleration along x, one may write

$$a_1 = \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \quad . \qquad (3)$$

For the time being we will ignore the force applied to the airplane (and therefore also to the air, according to Newton's Third Law) by the air sliding over the skin. We note that air is caused to move by a pressure difference. In fact, the greater the pressure difference _per unit distance_ the greater will be the force which causes the air to move. Thus, according to Equations (2) and (3) we write

$$-\frac{\partial P}{\partial x} = \rho \left[\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right]$$

$$-\frac{\partial P}{\partial y} = \rho \left[\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial y} + u \frac{\partial v}{\partial x} + w \frac{\partial v}{\partial z} \right] \qquad (4)$$

$$-\frac{\partial P}{\partial z} = \rho \left[\frac{\partial w}{\partial t} + w \frac{\partial w}{\partial z} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} \right] \quad .$$

At this point we will assume that the aircraft velocity is steady—unchanging in time—and that the speed of the air flowing over the airplane depends only upon _its position with respect to the aircraft_. This assumption permits us to ignore terms of the type $\partial u/\partial t$ and to write Equation (4) as follows

$$-\frac{\partial P}{\partial x} = \rho \left[u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right]$$

$$-\frac{\partial P}{\partial y} = \rho \left[v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + u \frac{\partial v}{\partial x} \right] \qquad (5)$$

$$-\frac{\partial P}{\partial z} = \rho \left[w \frac{\partial w}{\partial z} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} \right] .$$

We will need these equations later to relate the pressure forces on wings and fuselages to the velocity of the air moving around them.

Let us now examine the imaginary cube through which we assume the air to be flowing. Consider one face as shown in the sketch below:
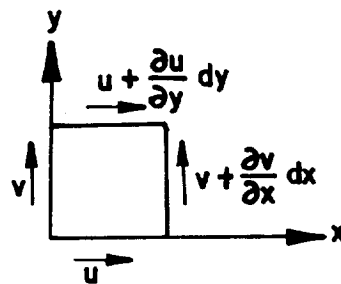
By assuming that the air slides over the surface of the airplane without applying any force to it we have in effect assumed that the air has no way of transmitting shearing forces to itself or to solid bodies with which it comes in contact. The measure of this shearing action is the viscosity of the fluid. We have assumed therefore that air is <u>inviscid</u>. Since we have no way of causing one fluid cube to shear against another it is reasonable to conclude that there is no way a fluid cube can be caused to rotate. From the sketch we see that a mathematical statement of this conclusion is

$$u \, dx + \left(v + \frac{\partial v}{\partial x} \, dx\right) dy - \left(u + \frac{\partial u}{\partial y} \, dy\right) dx - v \, dy = 0 \tag{6}$$

Simplifying the equation, one has

$$\frac{\partial v}{\partial x} \, dxdy - \frac{\partial u}{\partial y} \, dxdy = 0$$

or

$$\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \quad . \tag{7}$$

Similar expressions can be obtained for the other five faces of the cube.

We move now to consider the consequences of the condition represented by Equation (7). A general differential $d\phi$ can be written

$$d\phi = u \, dx + v \, dy + w \, dz \quad . \tag{8}$$

Since

$$d\phi = \frac{\partial \phi}{\partial x} \, dx + \frac{\partial \phi}{\partial y} \, dy + \frac{\partial \phi}{\partial z} \, dz \quad , \tag{9}$$

$$u = \frac{\partial \phi}{\partial x} \, , \quad v = \frac{\partial \phi}{\partial y} \, , \quad w = \frac{\partial \phi}{\partial z} \quad . \tag{10}$$

But

$$\frac{\partial^2 \phi}{\partial x \partial y} = \frac{\partial^2 \phi}{\partial y \partial x} \quad .$$

Therefore

$$\frac{\partial u}{\partial y} = \frac{\partial v}{\partial x} \, , \quad \frac{\partial v}{\partial w} = \frac{\partial w}{\partial y} \, , \quad \frac{\partial w}{\partial x} = \frac{\partial u}{\partial z} \quad . \tag{11}$$

Consequently, requiring that the fluid be irrotational means that $d\phi$ is an exact differential. The integral of an exact differential is independent of the path of integration and depends only on the value of the function at the two end points. We say then that for an irrotational fluid a function ϕ, called a velocity potential, exists such that its partial derivatives represent the components of the fluid velocity. See Equation (10).

Using this result in Equation (1) one may write

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \tag{12}$$

52

This partial differential equation is called the Laplace Equation and is one of the most studied in the mathematical literature. Note that there is only one dependent variable. Thus if one can find a function ϕ which satisfies the equation and the boundary conditions he should be able to find the velocity around a body represented by the boundary conditions. Note that since the Laplace Equation is linear, a sum of solutions is also a solution. We shall make use of this fact to represent complex bodies as a sum of simple solutions.

For the calculation of the lift and drag of aircraft wings we will restrict our attention for the time being to flow in a plane aligned with the airstream. Therefore we take $w = 0$ and ignore derivatives with respect to z. We postulate that a function

$$\phi = \frac{\Gamma}{2\pi} \tan^{-1} \frac{y}{x} \quad , \tag{13}$$

where Γ is a constant, is a solution of the equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad . \tag{14}$$

Now

$$\frac{\partial \phi}{\partial x} = - \frac{\Gamma}{2\pi} \frac{y}{x^2 + y^2} \tag{15}$$

$$\frac{\partial \phi}{\partial y} = \frac{\Gamma}{2\pi} \frac{x}{x^2 + y^2} \quad , $$

hence

$$\frac{\partial^2 \phi}{\partial x^2} = - \frac{\Gamma}{2\pi} \frac{y(-2x)}{(x^2 + y^2)^2} \tag{16}$$

$$\frac{\partial^2 \phi}{\partial y^2} = \frac{\Gamma}{2\pi} \frac{x(-2y)}{(x^2 + y^2)^2} \quad . $$

The sum of the two equations in (16) is seen to equal zero and thus Equation (13) is a solution of Equation (14). The figure below depicts this function.

— Lines of constant velocity

--- Lines of equal value of velocity potential
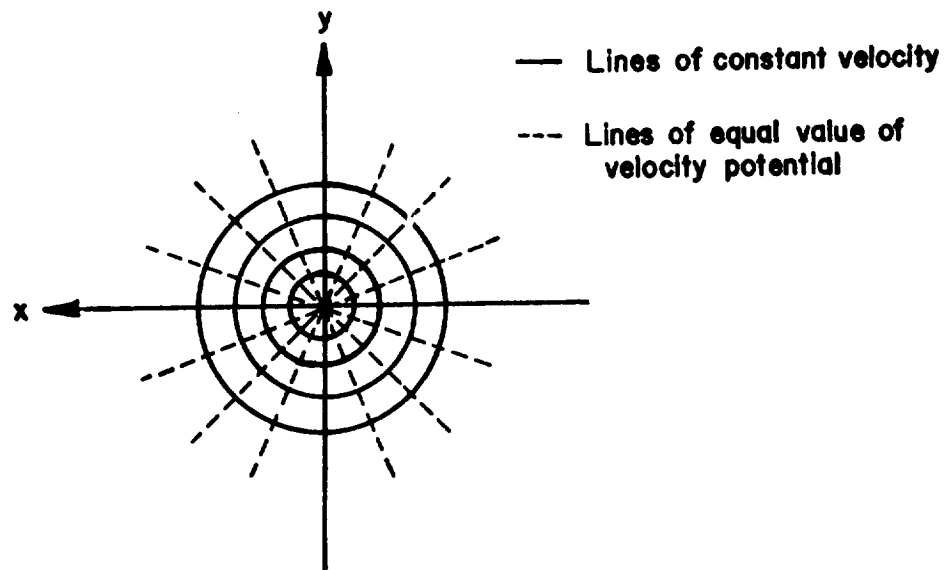
For a given value of Γ the velocity decreases as $1/(x^2 + y^2)^{\frac{1}{2}}$. This type of flow is called a vortex. It is irrotational. The quantity Γ is called the vortex strength.

If the vortex is not located at the origin the expression for the potential becomes

$$\phi = \frac{\Gamma}{2\pi} \tan^{-1} \frac{y - y_0}{x - x_0} \tag{17}$$

where (x_0, y_0) is the location of the center of the vortex. Corresponding expressions for the velocity components are

$$u = - \frac{\Gamma}{2\pi} \frac{y - y_0}{(x - x_0)^2 + (y - y_0)^2}$$

$$v = \frac{\Gamma}{2\pi} \frac{x - x_0}{(x - x_0)^2 + (y - y_0)^2} \quad . \tag{18}$$

Now suppose we place a number of vortices in a plane and ask what is the velocity induced at a point (x, y). We should be able to assume that the vortices may all be treated separately and that their contributions at a point may be summed to find the net velocity. We write therefore

$$\bar{u} = - \frac{1}{2\pi} \sum_{N=1}^{k} \frac{(y - y_{0_N}) \Gamma_N}{(x - x_{0_N})^2 + (y - y_{0_N})^2}$$

$$\bar{v} = \frac{1}{2\pi} \sum_{N=1}^{k} \frac{(x - x_{0_N}) \Gamma_N}{(x - x_{0_N})^2 + (y - y_{0_N})^2} \tag{19}$$

where k is the total number of vortices.

REPRESENTATION OF AN AIRFOIL IN TWO-DIMENSIONAL FLOW

Consider then the possibility of placing the vortices on the perimeter of an airfoil. If we can choose the strength of each vortex properly we should be able to make the net flow velocity along the airfoil surface satisfy, at least approximately, the boundary condition that the flow be parallel to the airfoil surface. To do this it is necessary to consider in addition the contribution of the free stream to the total flow picture. The velocity potential for a uniform stream is readily shown to be

$$\phi = -Vx \quad . \tag{20}$$

Hence, the effect of the free stream is to add a velocity V to $\bar{u}$. The boundary condition may then be stated as

$$\tan^{-1} \left(\frac{\bar{v}}{V + \bar{u}} \right) = \tan^{-1} \left(\frac{dy}{dx} \right)_{wing} - \alpha$$

54

or

$$\frac{\overline{v}}{V + \overline{u}} = \left(\frac{dy}{dx}\right)_{wing} - \tan \alpha \qquad (21)$$

where α is the angle of attack of the wing chord line.

$\left(\dfrac{dy}{dx}\right)_{wing}$ is the slope of the wing surface (measured relative to the axis system) at the point where the boundary condition is to be satisfied. The use of this notation follows from the concept of describing the surface by the equation $y = f(x)$. The local value of the tangent to this curve is of course given by $dy/dx|_x$.

In Equations (19) Γ_N, the vortex strength at each of the k points, is unknown and must be found by applying the boundary condition k times, that is, by writing k equations. Since we can accomodate no more than k values of the surface slope, one usually depicts the airfoil as consisting of straight line segments connecting the vortex centers. For this reason one usually desires k to be large in order to describe thick or cambered airfoils accurately.

The concept for representing airfoils and the significance of the various symbols used in the text is illustrated in the sketches below.

PHYSICAL AIRFOIL SHAPE
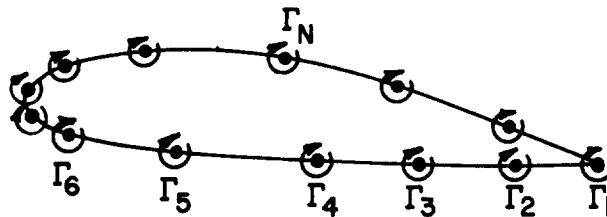
REPRESENTATION OF THE DISTURBANCE CAUSED IN A UNIFORM STREAM BY THE PRESENCE OF AN AIRFOIL THROUGH THE USE OF A DISTRIBUTION OF VORTICES OF DIFFERENT STRENGTHS PHYSICALLY SITUATED AS IF THEY WERE LOCATED ON THE AIRFOIL SURFACE. ARROWS INDICATE DIRECTION OF VORTEX FLOW AND DOTS SHOW LOCATION OF LINE VORTICES EXTENDING TO INFINITY INTO AND OUT OF PAPER.
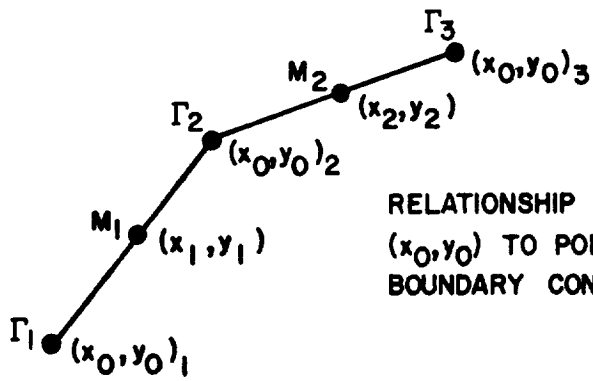
RELATIONSHIP OF VORTEX CENTERS
(x_0, y_0) TO POINTS (M) AT WHICH
BOUNDARY CONDITIONS ARE SATISFIED

VELOCITY COMPONENTS,
u AND v, INDUCED AT
POINT M BY VORTEX
AT ①

slope of surface

$= \left(\dfrac{dy}{dx}\right)_3$

VELOCITY COMPONENTS
INDUCED AT M BY
VORTEX AT ②

56

VELOCITY COMPONENTS
INDUCED AT M BY
VORTEX AT ③

slope of surface

$$= \left(\frac{dy}{dx}\right)_2$$

VELOCITY COMPONENTS
INDUCED AT M BY
VORTEX AT ④

SUM OF INDUCED VELOCITIES
PLUS FREE STREAM
AT M IS PARALLEL TO
SURFACE.

ANGLE OF RESULTANT IS
EQUAL TO ANGLE SURFACE
MAKES WITH STREAM
DIRECTION LESS α.

57

Note that each vortex on the entire airfoil contributes its share to the net velocity which is produced at each M. Note also that we have chosen to bound each line segment by a vortex. This choice facilitates the description of point M in terms of the vortex locations as may be seen in the analysis below. We could have chosen instead to represent the airfoil by a group of line segments, locate a vortex at the 1/4 point, say, and satisfy the boundary condition at the 3/4 point on each line segment. Such a procedure requires that one first find the equation of each line segment and then locate the vortex and the control point (point where boundary condition is satisfied) on it.

Let us call the right-hand side of Equation (21) B_M, the tangent of the required flow angle at point M. M has coordinates (x, y). Let us also call

$$\frac{(y_M - y_{O_N})}{(x_M - x_{O_N})^2 + (y_M - y_{O_N})^2} = a_{MN} \tag{22}$$

$$\frac{(x_M - x_{O_N})}{(x_M - x_{O_N})^2 + (y_M - y_{O_N})^2} = b_{MN} \quad , \tag{23}$$

then

$$
\begin{aligned}
B_1 &= \frac{b_{11}\Gamma_1 + b_{12}\Gamma_2 + \cdots b_{1N}\Gamma_N + \cdots b_{1k}\Gamma_k}{V + a_{11}\Gamma_1 + a_{12}\Gamma_2 + \cdots a_{1N}\Gamma_N + \cdots a_{1k}\Gamma_k} \\
\cdot & \\
\cdot & \\
\cdot & \\
\cdot & \\
B_k &= \frac{b_{k1}\Gamma_1 + b_{k2}\Gamma_2 + \cdots b_{kN}\Gamma_N + \cdots b_{kk}\Gamma_k}{V + a_{k1}\Gamma_1 + a_{k2}\Gamma_2 + \cdots a_{kN}\Gamma_N + \cdots a_{kk}\Gamma_k}
\end{aligned}
\tag{24}
$$

represents the system of equations which must be solved to find all the Γ's.

To evaluate a_{MN} and b_{MN} in Equation (24) we need to locate points M in relation to $(x_O, y_O)_N$. For convenience we will choose M halfway between successive values of $(x_O, y_O)_N$. Now the line which extends from x_{O1}, y_{O1} to x_{O2}, y_{O2} is described by

$$y = y_{O_1} + \left[\frac{y_{O_2} - y_{O_1}}{x_{O_2} - x_{O_1}}\right](x - x_{O_1}) \quad . \tag{25}$$

If we let

$$x = x_{O_1} + \frac{x_{O_2} - x_{O_1}}{2} \quad ,$$

then

$$y = y_{O_1} + \frac{y_{O_2} - y_{O_1}}{2} \quad . \tag{26}$$

By generalizing Equation (26) to

$$x_M = x_{O_N} + \tfrac{1}{2} (x_{O_{N+1}} - x_{O_N})$$

$$y_M = y_{O_N} + \tfrac{1}{2} (y_{O_{N+1}} - y_{O_N}) \quad , \tag{27}$$

we can find the coordinates of all except one of the points where the boundary conditions must be matched.

The reason one boundary value cannot be found by this procedure is that by tracing line segments between successive points we draw what can be termed an open polygon. Such a polygon has one less line segment than points. For example, if one numbers four arbitrary points and draws line segments from pt. one to pt. two, from pt. two to pt. three, and then from pt. three to pt. four, he will have only three line segments connecting the four points. As a result, the maximum value of M is therefore $k - 1$. Note that if $N = k$ there is no $k + 1$ point. We can resolve this difficulty by closing the polygon and giving one point two names, $i.e.$ $N = 1$ and $N = k$. While this step does not provide us with an extra line segment at which to match the boundary condition, it does permit one to invoke the physics of the problem to reduce the number of unknown Γ_N's by one and thereby obtain a determinant system.

At this point the question is sure to arise in the mind of the uninitiated reader, why could not one simply locate one vortex and one control point on each line segment? The system is then completely determinant and there is no need to worry about finding another constraint. Unfortunately, such an approach encounters difficulties satisfying our present view of the physical situation. Consider for example the fact that with such a mathematical model we know the flow to be parallel to the line segments only at the control points. We do not know what the direction or magnitude of the resultant flow is anywhere else. The flow could very well cross the airfoil "surface". In particular we must be concerned about the situation at the trailing edge. Now we call one boundary for a quantity of flow a streamline. Far from the airfoil, or course, the streamlines are parallel. As the flow approaches the airfoil one streamline marks the exact line above which all the flow moves over the upper surface of the airfoil and below which all the flow moves over the lower surface of the airfoil. This line is called the stagnation streamline because at the point where it intersects the airfoil surface the flow velocity is exactly zero or stagnant.

The flow in the immediate neighborhood of this line obviously cannot move into the surface and it splits, half of it moves up and half down along the surface at this point with a net velocity of zero. Now, there must be another point of this type on the lee side of the airfoil where the flow around the airfoil comes together and leaves the surface. Where is this point located? Note the accompanying sketch.
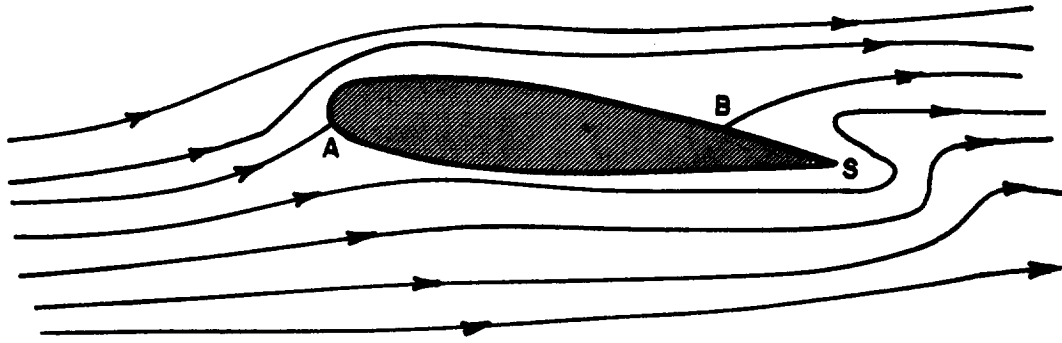
Point A here is the forward stagnation point. It would seem reasonable to
locate B such that the distance the flow would have to travel between A
and B over upper surface would be equal to the distance from A to B using
a lower surface route. For low drag, airfoils usually have trailing edges,
S, with very small radii of curvature. If the stream is to remain attached
to the surface and flow smoothly around this sharp corner, then it must
change direction very rapidly. This means there must then be portions of
this region where the velocity is very high and, according to Equation 31,
where the pressure is very low.

If the fluid experiences no frictional forces, the kinetic energy of
its motion at S is just sufficient to drive it to the stagnation point B.
Viscous forces, which are present in the actual case, retard this motion
so that the fluid does not quite reach B but stops somewhere on the way.
In fact, anytime the fluid does not quite reach B whether because of
viscosity or not a counter flow is set up (see sketch below) as some

fluid seeks to move from the high pressure area at B to the low pressure
area at S. This tends to separate the flow coming around S from the
upper surface. The action of these two flows creates a vortex. This
vortex, which is formed along the whole trailing edge of the airfoil,
is unstable and separates from the trailing edge. It is carried along
by the general motion.

According to Helmholtz's theorem, vortex cores are closed figures.
Thus there must be a net circulatory flow around the entire airfoil in
the clockwise direction which is attached (at infinity) to the vortex
shed off the trailing edge (see following sketch). Incidently, this
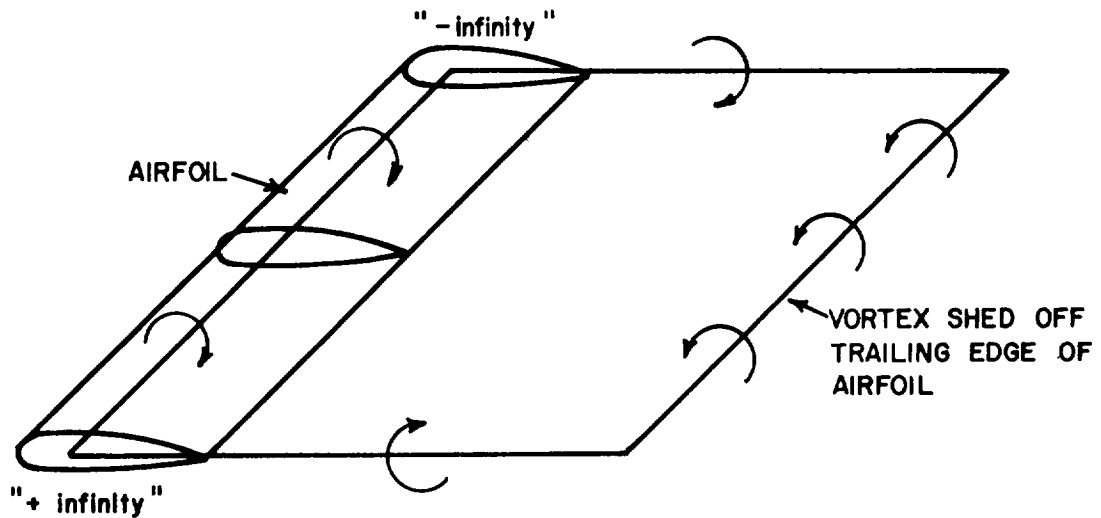
flow pattern can be seen in photographs taken of an airfoil just as it is
set into motion. The flow is made visible by inserting the airfoil verti-
cally into stagnant water on which finely ground aluminum powder is
floating. Such a vortex is shed everytime there is a change in lift.

Simultaneously with the vortex shedding process and the creation of
a net clockwise circulation around the airfoil itself, the stagnation point
B is displaced until it resides approximately at S. The fluid then no
longer moves around the trailing edge but flows off tangentially with
equal velocity at both sides. The assumption that the flow will leave the
trailing edge smoothly was put forward independently by Kutta and Joukowski.
It is the salient point in the theory of lift because it determines the
magnitude of the circulation. By means of this hypothesis the whole problem
of lift becomes purely mathematical: one has only to determine the amount
of circulation so that the velocity of the flow leaving the upper surface
at the trailing edge is equal to that of the flow leaving the lower
surface. The rule stated in this way applies to wings with zero vertex
angle. If tangents to the upper and lower surfaces form a finite angle,
the trailing edge is a stagnation point. Most airfoils with which we
will be concerned have finite trailing edge angles; according to the Kutta-
Joukowski hypothesis the flow velocity and hence the circulation at the
trailing edge of these airfoils is zero. If we locate the first of the
vortex filaments describing the airfoil at the trailing edge and require
that the net velocity at this point be zero, we must also choose the
strength of the vortex filament at that point to be zero since the velocity
at the core of a vortex is infinite. At first thought one would simply
take $\Gamma_1 = \Gamma_k = 0$ but this would mean that there is one more equation in
(24) than is necessary, something we know is not true. To skirt this
mathematical problem we take

$$\Gamma_1 = -\Gamma_k \ .$$

<div align="right">(28)</div>

Obviously, if we have two vortices located at the same point with equal and opposite strengths the sum of those strengths (which is what we would see physically) is zero. Through this device we satisfy the requirement that a stagnation point will always exist at point 1. Simply saying that $\Gamma_1 = \Gamma_k$ does not insure that Γ_1 will come out to be zero even if point 1 is at the trailing edge. Finally, with these constraints Equations (24) now read

$$B_1 = \frac{1}{V}\left[(b_{11} - B_1 A_{11})\Gamma_1 + (b_{12} - B_1 A_{12})\Gamma_2 + \cdots (b_{1N} - A_{1N}B_1)\Gamma_N + \right.$$

$$\left. \cdots (B_1 a_{1k} - b_{1k})\Gamma_1 \right] \tag{29}$$

$$B_{k-1} = \frac{1}{V}\left[(b_{(k-1)1} - B_{k-1}a_{(k-1)1})\Gamma_1 + (b_{(k-1)2} - B_{k-1}a_{(k-1)2})\Gamma_2 + \right.$$

$$\left. \cdots (b_{(k-1)N} - B_{k-1}a_{(k-1)N})\Gamma_N + \cdots (B_{k-1}a_{(k-1)k} - b_{(k-1)k})\Gamma_1 \right]$$

Equations (29) contain $k - 1$ distinct values of Γ_N and $k - 1$ values of B_M so that the system is solvable for all $k - 1$ values of Γ_N. Admittedly, when k is a large number this is a task which is practical only when one is able to use a digital computer with sufficient storage (memory) to carry out the solution process. Generally, the process should be one able to accommodate a system of 65 or more simultaneous algebraic equations. The most favored method for solution of such large systems is a generalization of Cramer's Rule, familiar to college algebra students as a technique for solving systems of 3 or 4 simultaneous algebraic equations. Fortunately, the nature of the matrix form of this large system is such that the solution is less difficult than one might expect to encounter for systems of this size. Those readers interested in the mathematical details of the numerical methods one might employ for this purpose are referred to texts and papers on matrix algebra and in particular to papers on matrix inversion techniques. Since these techniques are not elementary (they are, fortunately, usually available as standard computer library programs) and since their details are not crucial to an understanding of the physical reasoning used to formulate Equations (29), they will not be discussed here.

A cautionary note regarding Equations (29) should, perhaps, be injected at this point. In their formulation, it was assumed that the coordinates of the vortex centers were measured from a fixed reference. Therefore, when the angle of attack changes, the locations of the vortex centers as well as the coordinates of the center of the connecting line segments change. In other words, a_{MN} and b_{MN} depend for their values upon the airfoil angle of attack. On the other hand, the local airfoil slope, $((dy)/(dx))_{wing}$ can be taken as constant for all α since the term $\tan \alpha$ is substracted from the slope to calculate the boundary condition, Equation (21). An alternate formulation considers the airfoil to be fixed and the flow to rotate as α changes. The left hand side of Equation (21) will then read

$$\frac{\overline{v} + V \sin \alpha}{\overline{u} + V \cos \alpha}$$

while the right hand side will no longer have the term $\tan \alpha$.

62

In addition to the necessity of locating two vortex filaments at the trailing edge as discussed above, there is a second geometrical aspect which should be considered in choosing the locations of the other vortex filaments by which one represents the airfoil. This aspect is the surface curvature. Obviously, the line segments between filaments must be more numerous in regions of high curvature if one is to describe the surface accurately. Thus, for most airfoils the preponderance of points should be located near the leading edge. One can quantify this procedure by specifying that the distance from the line segment to the surface should never exceed a fixed, small percentage of the airfoil chord (distance from leading to trailing edge). The orientation of the local surface to the stream also has an effect on the vortex filament spacing. Surfaces nearly parallel to the free stream can be represented more accurately with a few filaments than surfaces with large inclinations.*

The solution of Equation (29) is k values of Γ_N with $\Gamma_1 = -\Gamma_k$. Now that we have these values we ask how we may use them to find what is really of interest to us: the local pressures on the surface of the airfoil. We begin by considering how one might write a two-dimensional version of Equations (5) in a coordinate system with one axis tangent to stream line and the other normal to it. In such a coordinate system (5) reduces to

$$- \frac{\partial P}{\partial s} = \rho \tilde{u} \frac{\partial \tilde{u}}{\partial s} \tag{30}$$

where $\tilde{u}$ is the total fluid velocity and s is the distance along the streamline. Equation (30) is easily integrated to yield

$$P_s = P + \tfrac{1}{2} \rho \tilde{u}^2 . \tag{31}$$

P_s is the stagnation pressure along a streamline, that is, the pressure which would exist if the fluid were brought to rest. It is usually assumed that the disturbances produced by the airfoil decay to zero at great distances upstream and downstream of the airfoil, *i.e.* at these stations. If the flow is uniform then the stagnation pressure must be the same along all streamlines. Since there is no mechanism in an irrotational, incompressible, inviscid flow by which the stagnation pressure can change, P_s is then the same on all streamlines throughout the flow field. It is easily evaluated since P_∞, ρ, and V are known in a given flow.

---

* One might postulate that by representing both the boundary on which parallel flow is to be required and the vortex strength explicitly by continuous functions of the surface coordinates with continuous derivatives everywhere, the intervals over which these functions are fitted can be significantly larger, for equal accuracy, than the straight line segments (with their delta-function-like circulation strengths) used in the present approach. Larger segments, of course, means fewer (although perhaps more complex) equations to solve. Hess (Refs. 81, 82) studied this question at some length. He begins with the implicit assumption that one would always wish to represent the distribution of vorticity over the airfoil surface as a continuous distribution. The strength of the vorticity

over any segment of the airfoil surface is then an integral of the vortex strength per unit length over that segment. Since this integral equation usually cannot be solved in general, one usually seeks to find approximate means of solution. Hess states that "the most straightforward means of evaluating the integral is by means of a quadrature formula that replaces the integral by a weighted sum of values of the integrand evaluated at certain points. That is, the effect of a continuous singularity distribution is approximated by a sum of concentrated point singularities...This is not a satisfactory procedure. The basic difficulty is that the velocity approaches infinity more rapidly in the neighborhood of a point singularity than it does near a finite-strength singularity distribution on an arc of a curve. Thus the spacing of the quadrature points must be small compared to all physical dimensions of the boundary. This is not practical for airfoils which are often quite thin." Near the trailing edge, particularly, the problem is quite difficult. "Adjacent to the corner the ratio of the normal distance between two points to the spacing between them is approximately the sine of the trailing edge angle, no matter how many points are used."
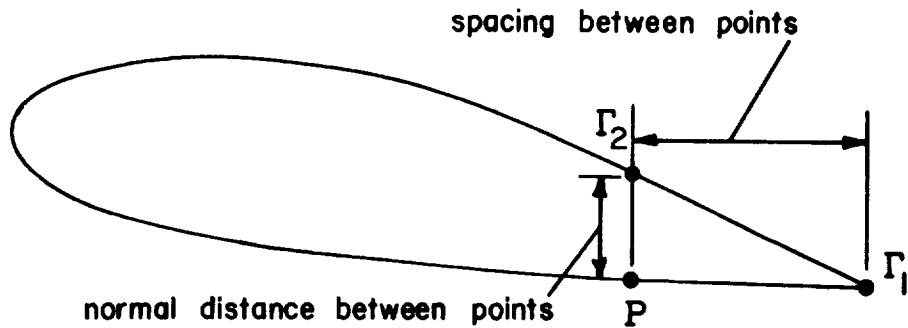
As a result, the velocity at P on the lower surface of the airfoil in the sketch above is dominated by the contribution from Γ_2, whereas in the integral representation the contribution from vorticity at some distance from Γ_2 will still be significant.

To illustrate these concepts consider the following example. Take two flat plates which join at an angle of 3°.
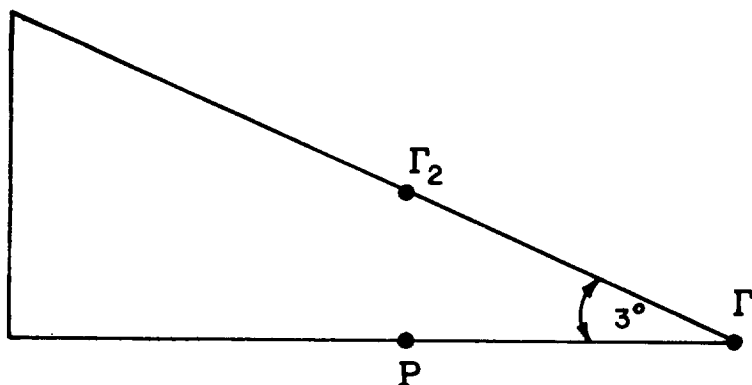
Place vortices Γ_1 and Γ_2 10 units apart as shown and assume $\Gamma_1 = \Gamma_2$. Now Γ_2 induces a velocity which is parallel to the lower plate at P which has a magnitude equal to $(\Gamma/2\pi)(1.91)$ while Γ_1 induces a velocity at P which is normal to the plate and has a magnitude equal to $(\Gamma/2\pi)(0.1)$.

The integral representation for constant vortex strength per unit length is given by

$$V \approx \frac{1}{2\pi} \int_0^{10} \frac{\Gamma}{10} \left[\ln \sqrt{1.002742\ x^2 - .05483\ x + .2742} \right] dx$$

$$\approx \frac{\Gamma}{2\pi} \left(1.37 \right) \ .$$

Note that in this example the magnitude of the induced velocity at P is, as stated by Hess, less for a continuous distribution of vorticity than for a distribution of concentrated point singularities. It would appear, therefore, that the use of continuous distributions of vorticity would lead to less "wavy" flows in the neighborhood of airfoil surfaces and is therefore desirable.

Hess systematically investigated the effectiveness of higher order approximations of the integral equation, including the use of curved surface elements and parabolically-varying vorticity. He found that the approach using flat elements with constant singularity is mathematically consistent as is the next higher-order approach with parabolic elements and linearly-varying singularity strength. The popular approach based on flat elements with linearly-varying singularity strength he shows to be mathematically inconsistent. Hess concludes that "(1) the higher order solutions give very little increase in accuracy for the important case of exterior flow about a convex body; (2) for bodies with substantial concave regions and for interior flows in ducts, the use of parabolic elements and linear varying singularity can give a dramatic increase in accuracy; and (3) the use of still higher order solutions leads to a rather small additional gain in accuracy."

The results which Hess obtained for the surface velocity on a semi-infinite body whose forward portion is a semi-circle concave to the flow has a particular bearing on the question of whether the use of higher order solutions can lead to shorter computational times for equal accuracy. Using 36 curved elements with linearly-varying source strengths he was able to recover the values given by the analytical solution. Even with five times that number of straight elements with constant source strengths—and, more importantly, one hundred times the computing time—he was not able to recover the correct values. The results were qualitatively correct but were quantitatively in error, particularly on the centerline.

Hess also studied the use of the various solutions on a highly cambered Kármán-Trefftz airfoil for which an analytical solution is known. Since the lower surface of this airfoil is concave, use of the higher order method was found to be necessary in order to obtain good agreement with the analytical result.

The implications of Hess's work, so far as the prediction of airfoil characteristics is concerned, are (1) for conventional airfoils little reduction in computational time—for the same accuracy—can be achieved by using fewer curved elements and linearly-varying singularity strengths, (2) for unconventional airfoils—airfoils with concave areas on their lower surfaces—use of the higher order solution will yield substantial improvements in accuracy for a given computational time.

Now the velocity along a streamline is given by

$$\sqrt{(V + \overline{u})^2 + \overline{v}^2} = \widetilde{u} \tag{32}$$

so that the fluid pressure which acts normal to the airfoil surface at point (x, y) is simply

$$P = P_\infty + \tfrac{1}{2}\rho V^2 - \tfrac{1}{2}[(\overline{u} + V)^2 + \overline{v}^2]$$

or

$$P = P_\infty - \tfrac{1}{2}\rho[\overline{u}^2 + \overline{v}^2 + 2\overline{u}V] \quad . \tag{33}$$

Note that we have used all of the Γ_N found from a solution of (29) to determine $\overline{u}$ and $\overline{v}$ at a set of points (x, y) according to Equations (10). It may be well to note here that requiring the flow generated by the system of vortices to be parallel to the surface at the mid-segment points does not guarentee that it is also parallel to the surface everywhere. Thus it is prudent to select x,y for the pressure computation to be those same mid-segment points. Pressures at points inbetween are probably best found from a third order polynomial fit to four successive calculated values.

The net lift on the airfoil is obtained by integrating the components of the pressure which are normal to the free stream direction over the entire surface of the airfoil. Mathematically, this process can be expressed by

$$L = \cos\alpha \oint P(x) \cos\left[\tan^{-1}\left(\frac{dy}{dx}\Big|_x\right)\right] dx - \sin\alpha \oint P(x) \sin\left[\tan^{-1}\left(\frac{dy}{dx}\Big|_x\right)\right] dx \quad . \tag{34}$$

The lift coefficient is then $L/\tfrac{1}{2}\rho V^2 c$.

To perhaps aid the reader in grasping some of the foregoing concepts, we will digress momentarily from the main thrust of our argument, retrace some of our steps and approach the calculation of airfoil lift from a slightly different direction. We will employ for this purpose the analysis given by von Kármán and Burgers in Volume II of the six volume set, Aerodynamic Theory, edited by W. F. Durand, (Julius Springer, 1935). Equations (19) when expressed in polar coordinates become

$$v_r = \sum_{N=1}^{k} \frac{\Gamma_N}{2\pi} \frac{r_{O_N}\sin(\theta - \theta_{O_N})}{r^2 + r_{O_N}^2 - 2r\,r_{O_N}\cos(\theta - \theta_{O_N})}$$

$$v_\theta = -\sum_{N=1}^{k} \frac{\Gamma_N}{2\pi} \frac{r - r_{O_N}\cos(\theta - \theta_{O_N})}{r^2 + r_{O_N}^2 - 2r\,r_{O_N}\cos(\theta - \theta_{O_N})} \quad .$$

In these expressions it is assumed that the positive sign of the circulation corresponds to a clockwise rotation of the fluid and that the nose of the airfoil points to the left into the oncoming stream. v_θ is positive in the counterclockwise direction. We now expand the velocities in a series containing decreasing powers of r. These series are convergent so long

as r is greater than any of the values $r_1, r_2, \ldots, r_n$, that is, so long as the point where we wish to compute the velocity components is further from the origin than any point on the airfoil surface. Under these conditions, the first term in the denominator of each series above is dominant and one may write

$$v_r = \frac{1}{r^2} \sum_{N=1}^{k} \frac{\Gamma_N}{2\pi} r_{0N} \sin(\theta - \theta_{0N}) + \text{higher terms}$$

$$v_\theta = -\frac{1}{r} \sum_{N=1}^{k} \frac{\Gamma_N}{2\pi} - \frac{1}{r^2} \sum_{N=1}^{k} \frac{\Gamma_N}{2\pi} r_{0N} \cos(\theta - \theta_{0N}) + \text{higher terms}.$$

We now assume a uniform and parallel fluid motion with velocity component V_x and V_y to be superimposed on the flow produced by the vortex filaments. We apply to the fluid within a circle of radius $r = K$ the theorem that the difference between the fluid momentum entering the circle and that leaving it is equal to the resultant of the forces acting on the fluid. The forces involved are (a) the pressure distributed along the circle and (b) the forces acting between the vortices and the fluid. The theorem may therefore be stated thus:

Resultant of forces = Resultant of pressures − Change in momentum

or

$$F_x = -\oint P \cos\theta \, ds - \rho \oint W_n W_x \, ds$$

$$F_y = -\oint P \sin\theta \, ds - \rho \oint W_n W_y \, ds$$

where F_x, F_y are the resultant forces in the x and y directions, θ locates the element ds and

$$W_n = V_x \cos\theta + V_y \sin\theta + v_r$$

$$W_x = V_x + v_r \cos\theta - v_\theta \sin\theta$$

$$W_y = V_y + v_r \sin\theta + v_\theta \cos\theta.$$

Then

$$F_x = -K \int_0^{2\pi} P \cos\theta \, d\theta - \rho K \int_0^{2\pi} (V_x \cos\theta + V_y \sin\theta + v_r)$$

$$\cdot (V_x + v_r \cos\theta + v_\theta \sin\theta) d\theta$$

$$F_y = -K \int_0^{2\pi} P \sin\theta \, d\theta - \rho K \int_0^{2\pi} (V_x \cos\theta + V_y \sin\theta + v_r)$$

$$\cdot (V_y + v_r \sin\theta + v_\theta \cos\theta) d\theta$$

$$P = P_\infty + \tfrac{1}{2}\rho(V_x^2 + V_y^2) - \tfrac{1}{2}\rho(V_x + v_r \cos\theta - v_\theta \sin\theta)^2$$

$$- \tfrac{1}{2}\rho(V_y + v_r \sin\theta + v_\theta \cos\theta)^2$$

$$= P_\infty - \tfrac{1}{2}\rho(v_r^2 + v_\theta^2 + 2v_r[V_x \cos\theta + V_y \sin\theta]$$

$$+ 2v_\theta[V_y \cos\theta - V_x \sin\theta])$$

If K is large, then $v_r \to 0$ and $v \to -\dfrac{1}{K} \displaystyle\sum_{N=1}^{k} \dfrac{\Gamma_N}{2\pi}$.

Further, the integral from 0 to 2π of $\sin\theta$, $\cos\theta$, and $\sin\theta\cos\theta$ are all zero so that in computing F_x and F_y one need do only

$$F_x = K \int_0^{2\pi} \frac{\rho}{2} 2\, v_\theta V_y \cos^2\theta\, d\theta + K\rho \int_0^{2\pi} v_\theta V_y \sin^2\theta\, d\theta$$

$$F_y = -K \int_0^{2\pi} \frac{\rho}{2} 2\, v_\theta V_x \sin^2\theta\, d\theta - \rho K \int_0^{2\pi} v_\theta V_x \cos^2\theta\, d\theta$$

which become

$$F_x = K\rho\, 2\pi\, v_\theta V_y$$

$$F_y = -K\rho\, 2\pi\, v_\theta V_x$$

Then with $v_\theta = -\dfrac{1}{K} \displaystyle\sum_{N=1}^{k} \dfrac{\Gamma_N}{2\pi}$ we have

$$F_x = -\rho V_y \sum_{N=1}^{k} \Gamma_N$$

$$F_y = \rho V_x \sum_{N=1}^{k} \Gamma_N$$

In the case where the fluid motion is parallel to the x-axis we see that $V_x = V$ and $V_y = 0$ so that the resultant force or lift,

$\rho V \displaystyle\sum_{N=1}^{k} \Gamma_N$, is normal to the direction of the stream. Since Joukowski

found that the lift of an airfoil is $\rho V\Gamma$, it is evident that the net circulation about the airfoil is the sum of the individual vortex filament strengths.

A simplified form of Equation (34) can also be used to show an interesting relationship between the vortex filament strengths and the surface velocities. Since

$$P(x) = P_\infty - \frac{\rho}{2} [\overline{u^2} + \overline{v^2} + 2\overline{uv}]$$

we could take for $\alpha = 0$ and an airfoil surface made up of straight line segments

$$L = \int P(x) \left(\frac{\Delta x}{\Delta s} \right)_x dx \;,$$

where $\left(\dfrac{\Delta x}{\Delta s} \right)_x$ represents the cosine of the angle formed by a particular

line segment and the chord line. If we are willing to say that this cosine is always near unity then we may ignore the v-component in the velocity and also $\overline{u}^2$ in comparison with $2\overline{u}V$. With these approximations we have for the lift

$$L = \oint \left[P_\infty - \frac{\rho}{2} (2\overline{u}V) \right]_x (\Delta x)_x \, dx \;.$$

Since we are discussing a surface made up of line segments $(\Delta x)_x$ and we assume that the pressure is constant over each line segment, the integral is readily approximated by a series:

$$L = \sum_x \rho V \overline{u}_x (\Delta x)_{x_{upper}} + \sum_x \rho V \overline{u}_x (\Delta x)_{x_{lower}} \;.$$

Note that $\oint P_\infty \, dx = 0$. From a comparison between this expression for the

lift and that in terms of the vortex filament strengths where

$$L = \rho V \sum_N \Gamma_N$$

we see that

$$\overline{u}_x = \frac{\Gamma_x}{(\Delta x)_x} \;.$$

In other words, the velocity induced along a segment of the airfoil surface is equal to the vorticity per unit length existing over that segment.

Some comments concerning the results of the procedure leading to (34) are appropriate at this point. First, it must be recalled that the procedure considers air to be an inviscid, incompressible fluid. Thus, there is no dissipative mechanism available to produce a drag. Hence, the integral of the streamwise components of the pressure force on the airfoil must, if the calculation has been carried out accurately and correctly, be zero, or in mathematical terms,

$$\sin \alpha \oint P(x) \cos \left[\tan^{-1} \left(\frac{dy}{dx}\Big|_x \right) \right] dx$$

$$+ \cos \alpha \oint P(x) \sin \left[\tan^{-1} \left(\frac{dy}{dx}\Big|_x \right) \right] dx \leqq 0.00005 \;. \tag{35}$$

Carrying out the procedure indicated by Equation (35) is an excellent way to check the accuracy and validity of the method used to obtain a numerical result for Equation (34).

A second anomaly resulting from the use of an inviscid theory is that one cannot predict the approach of the phenomenon pilots call "stall." Physically, stall is characterized by a loss of lift and a sharp increase

70

In drag as the wing's angle of incidence to the stream is increased. Thus an airfoil exhibits a maximum lift coefficient just prior to stall. It usually occurs at angles of incidence to the airstream (α) in the neighborhood of $16°$ to $20°$. The inviscid theory, on the other hand, will predict maximum lift at $\alpha \to 45°$. Despite this deficiency, the inviscid theory will generally give quite satisfactory predictions of lift for $\alpha < 6°$ or $8°$.

TREATMENT OF VISCOUS EFFECTS

It should be evident from the previous comments that to be able to make a realistic estimation of aircraft performance one must find some means to consider in the calculations the property by which air is able to resist the motion of aircraft and propellors: viscosity. It has been known for a long time that inclusion of this property in the equations describing fluid motion makes them (a) non-linear, (b) higher order, (c) consider energy transport, and, consequently, a pressure-density-temperature relation—factors which make them virtually insolvable in general. In one of the greatest contributions to the analytical description of physical reality, Ludwig Prandtl argued in 1904 that for most practical applications one could consider the effects of viscosity to be confined to a thin layer of the fluid immediately adjacent to the airfoil surface which he called the boundary layer. Prandtl argued that the remainder of the flow field can be treated quite adequately by retaining the fiction that the fluid is inviscid. If we assume that we have an acceptable method to calculate the lift on an airfoil-like body we must ask ourselves how can we include this viscous boundary layer in the treatment and what are its effects.

The concept of viscosity means that there is a transport or communication of the momentum of the fluid in one layer to the fluid in the adjacent layer. Whereas in inviscid theory we assumed that the fluid layer immediately adjacent to the surface of a flat plate has the same velocity as that far from the surface, we recognize that the stationary character of the surface must be known to the fluid immediately adjacent to it. This fluid cannot be moving very rapidly with respect to the surface because the molecules lose a significant portion of their tangential momentum in striking the stationary surface. This change in momentum appears as a frictional force on the surface. The layers outside the one closest to the surface, however, continually feed in additional momentum so that the net result is the development of a gradation in fluid velocity from the surface to the edge of the boundary layer. At the surface the velocity relative to the surface is zero. At the outer edge of the boundary layer the velocity is equal to that in the inviscid flow. See the following sketch.

Shown on the sketch is a graph of the variation of fluid velocity with height above a solid surface. u_e is the inviscid free stream velocity. A similar graph can be constructed at each streamwise location along the surface. Generally, the height over which the fluid velocity moves from zero to the inviscid free stream value, the distance labeled δ on the graph, increases as one goes downstream.
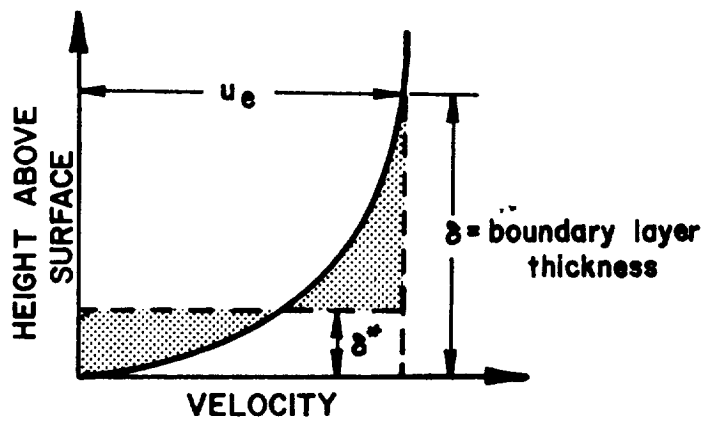
Looking at the sketch one notes that the two cross-hatched areas are approximately equal. This suggests that one could say, for purposes of modeling, that all of the fluid mass in the boundary layer is really concentrated in a region of uniform velocity u_e extending from δ^* outward to δ and that the region from the surface to δ^* could be considered part of the body since in this model there is no fluid in it. It would seem then that a way of accounting for some of the effects of viscosity is to determine δ^*, add this value to the airfoil ordinates, and recompute the lift values by inviscid theory based on this modified shape. As we shall see, this is an iterative process since the value of δ^* depends upon the value of the surface pressures.

Because of the existence of the boundary layer and its accompanying viscous dissipation, the flow in the immediate area of the trailing edge has a lower stagnation pressure than that at the leading edge. Since the pressure just aft of the airfoil must be about the same throughout a plane normal to the stream, there will be a region formed by the confluence of the two surface boundary layers where the fluid velocity will be less than free stream. This is called a wake, that is, a region where the fluid is relatively static with respect to the airfoil. Wakes tend ultimately to diffuse and disappear downstream. A wake exists whether or not the boundary layer(s) separate from the airfoil. The wake is of course much thicker if there is separation.

Another way of looking at the effect of a wake is to note that in moving over an obstruction the fluid velocity increases in proportion to the vertical displacement of the obstruction at the particular streamwise location. As the velocity goes up, the pressure must come down, according to Equation (31). Then, having passed the peak of the obstruction, the fluid begins to decelerate and the pressure begins to rise. Now, the effect of the wake coming off the trailing edge of the airfoil is to prevent the inviscid flow from returning all the way to its orginal free stream value. See the sketch below. Consequently, the pressure at the beginning of this wake region, point "a" on the sketch, is substantially lower than stagnation pressure. Since the streamlines are essentially straight downstream of point "a" it means the flow is more or less uniform and the pressure over

72
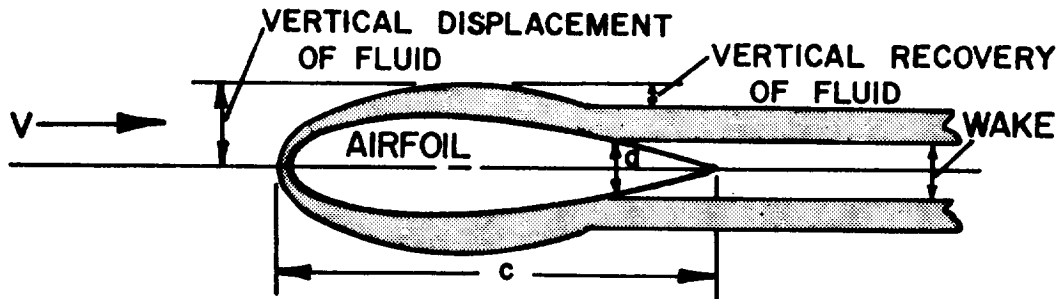
the entire airfoil aft of point "a" is approximately constant at the low value that exists at point "a". An integral of the surface pressures in the streamwise direction is now not zero. There is a finite drag, called in this case the form drag since it is dependent for its value on the form or shape of the body.

To accomodate this situation within the bounds of our inviscid theory we can proceed as follows: take as our body shape the airfoil plus δ^* up to x = c. At x = c, assume the body continues to extend downstream by an amount equal to

$$\left(\delta^* \middle/ \frac{dy}{dx} \right)_{x=c} ,$$

the projections from the two surfaces coming together in a point at this location. By this device we artificially create the sharp trailing edge we must have to satisfy the condition $\Gamma_1 = -\Gamma_k$. Γ_1 and Γ_k of course are then placed at this ficticious trailing edge point. Next we compute the pressures as a function of x (to x = c only) in the conventional manner and then employ (34) to obtain the lift and (35) to obtain the form drag. In essence then, we apply the pressures computed with the perturbed shape (geometric plus displacement thickness, δ^*) to the actual physical airfoil shape at the same chordwise station. There is of course some error involved in this procedure because the shape of the pseudo-body aft of x = c may cause the inviscid flow to decelerate somewhat more quickly than is actually the case and some of this effect will be apparent in the pressures calculated for points just upstream of the corners. In other words, it will tend to make the computed drag somewhat lower than it actually is. This effect is not serious as long as the wake is small compared with the airfoil thickness. Unfortunately, the model is not readily amenable to more sophisticated treatments of the wake effect* and the problem of accuracy is probably best

---

* Other schemes for treating the trailing edge condition which come to mind include (1) replacing the Kutta condition by the requirement that the vorticity shed into the wake from upper and lower surfaces be the same (2) extending the airfoil as a thin sheet along the wake centerline with the local curvature determined from considerations of the velocity distribution in the

73

wake according to viscous theory and the fact that the change in the fluid momentum in the y-direction in the upwash and downwash (aft of the airfoil) fields must equal the airfoil lift. One may also consider the superposition of a vortex distribution related in some predetermined way to that found above which will make the integral of the pressures over the pseudo-airfoil in the drag direction equal to zero.

Callaghan and Beatty (Ref. 69) in their treatment represent the displacement thickness with a source near the trailing edge. The pseudo-body then never closes and the source strength must be chosen to yield the proper wake thickness.

The very interesting approach used by Bhateley and McWhirter (Ref. 80) to treat this problem is in some respects quite similar. They do not employ the Kutta condition and, in addition, locate a source of undetermined strength within the airfoil. They must therefore supply two additional boundary conditions to obtain a solvable system. These are obtained by specifying two pseudo airfoil surface points just behind the trailing edge on both the upper and lower surfaces. The condition of continued tangential flow to the last surface element is satisfied at these pseudo boundary points. This type of analysis permits them to treat with good accuracy airfoils with slightly blunted trailing edges. These configurations are currently of increasing popularity because they can yield higher lift coefficients for the same angle of attack than the same airfoils with sharp trailing edges.

Bhateley and McWhirter further apply this concept to airfoils with partially separated boundary layers. Thus they are able to predict the variations in C_ℓ, C_d, and C_m with α up to α_{STALL} quite accurately. In their method the conditions of tangential flow are satisfied only on that part of the body having attached flow. If the boundary layer calculations indicate that the lower surface flow will separate, this fact is ignored and the displacement thickness is computed in the usual fashion. The two additional corner points are generated: one very close to the separation point on the upper surface and the other at the trailing edge on the lower surface of the pseudo body. The condition of continued tangency of the lower surface flow at the additional boundary point is satisfied. In specifying the additional boundary point aft of the separation point on the upper surface the user must select, based on experience, other analysis, etc., the direction of the flow leaving the upper surface. No pressures are calculated in the separated flow region. The pressure distribution downstream of the separation point is assumed constant and equal to that value of the pressure obtained by linear interpolation of the last two boundary point pressures prior to the separation point.

74

In this treatment, the separation point on the upper surface is considered to be the upper trailing edge point of a blunt trailing edge airfoil. It is easily seen, then, that the trailing edge thickness (and the source strength responsible for it) of the pseudo airfoil grows rapidly as α_{STALL} is approached. The pseudo airfoil begins to take on the appearance of a blunt body. Blunt bodies, of course, are known to have relatively high drag and relatively low lift so that it is easily seen how this approach can be used to account for the change in airfoil behavior from a low drag, high lift, relatively wake-free body at moderate angles of attack to a high drag, low lift, large wake body at high angles of attack.

The success of such a technique is, of course, highly dependent upon the accuracy with which one can predict the location of the boundary layer separation point on the airfoil. For this purpose Bhateley and McWhirter use a finite difference method in place of the momentum integral method discussed here. Studies conducted by colleagues of the present authors indicate that the momentum integral technique predicts increasingly more rearward separation locations (compared with predictions of the finite difference technique) as the angle of attack increases. Thus the lift predictions will be too large and the drag predictions too small at higher angles of attack compared with those obtained using a finite difference approach. On the other hand, the finite difference technique was found to require 20 times the computing time needed by the momentum integral technique.

The considerable success enjoyed by Bhateley and McWhirter in predicting the pressure distributions on airfoils at high angles of attack however seems to be more a function of the boundary layer routine they use than because they use an embedded source and two off-body tangency conditions. One can obtain similar results, for example, by replacing the Kutta condition by a requirement that the pressures at the upper surface and lower surface separation points and all points inbetween over the aft portion of the airfoil be the same, provided one does not then wish to construct a new pseudo body and compute from this a new potential solution. The source and two off-body tangency conditions are needed to determine the shape of the pseudo body aft of the separation point and thus to determine the potential flow about the pseudo body.

For airfoils with sharp trailing edges (and no boundary layer) Bhateley and McWhirter chose the Kutta condition in one of two forms: the flow 10^{-5} chord lengths behind the trailing edge is constrained to move in a direction which is an average of the airfoil surface slopes at the trailing edge or the net vorticity at the trailing edge is required to be zero.

handled for the present in a semi-empirical fashion, with the correction expected to be a function of angle of attack.

There is in addition to the form drag what is termed skin friction or shear drag, drag that is due directly to the sliding of the air over the airfoil surface. For well-streamlined shapes this accounts for perhaps 80% of the total drag. Reverting to our concept of momentum transport across layers of fluid as the drag mechanism, it is reasonable to postulate that the skin friction drag is proportional to the change in fluid velocity with distance from the surface. This may be expressed mathematically as

$$D_f = \mu A \left.\frac{du}{dy}\right|_{y=0} \tag{36}$$

where A is the surface area and μ is the coefficient of viscosity, a property of the particular fluid which usually varies with temperature. A very accurate relation for air is that due to Sutherland:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}}\right)^{3/2} \frac{T_{ref} + 198}{T + 198} \ . \tag{37}$$

The values for temperature, T, should be given in °R. For $T_{ref} = 519°R$, $\mu_{ref} = 373 \times 10^{-9}$ slugs/ft-secs.

---

* CONTINUED

The potential solution employed by Bhateley and McWhirter represents the airfoil surface by straight line segments and the vorticity distribution by linear variations between values at the corner points. This seems a bit strange in view of Hess's comment in Reference 82.

The troublesome trailing edge condition has also been the subject of several recent, extended, theoretical investigations. Spence (Ref. 70) presents a very graphic explanation of the problems and argues that because of the presence of a viscous wake the circulation around the airfoil should be multiplied by the factor

$$1 - \left(\frac{C_D}{4\pi}\right)\left(\ell n \frac{4}{C_D}\right) \ .$$

For a typical two-dimensional value of $C_D \approx 0.01$ this factor is about .995. Riley and Stewartson (Ref. 71) in examining the flow in the neighborhood of airfoil trailing edges conclude that if the angle of attack is small and the trailing edge angle is less than $1/Re^{\frac{1}{4}}$ then the flow will be maintained, without separation, up to the trailing edge. This is obviously a necessary condition for treating the effects of the viscous wake in a more general way since it establishes the point at which the flow leaves the surface although not necessarily the angle at which it leaves or its radius of curvature.

76

It is apparent from an examination of (36) that the problem in evaluating D_f (and δ^* for that matter) comes in finding

$$\left.\frac{du}{dy}\right|_{y=0}$$

as a function of position on the airfoil surface. One would like to integrate the local value of D_f over the total wetted surface to find the total skin friction drag. Several things complicate the problem:

1. The thickness of the boundary layer can be shown to depend upon the local Reynolds number, $\rho V_x/\mu$.

2. The character of the boundary layer—whether it flows in well-defined layers or lamina, or whether it flows in a more disorderly or turbulent fashion which one can only represent rather approximately—is also dependent upon the local Reynolds number <u>and</u> the condition of the surface. Further, the boundary layer may change from laminar to turbulent during the course of its travel over the airfoil.

3. The thickness of either a laminar or turbulent boundary layer depends upon the nature of the pressures outside the boundary layer. If the pressures are such that the flow tends to accelerate (high pressure upstream and lower pressure downstream) then the boundary layer grows very slowly. If the reverse is true (decelerating flow) the boundary layer grows rapidly. It may .even separate from the surface entirely.

The analysis of two-dimensional boundary layer flows to find $\delta^*(x)$ and $C_f(x)$ where $C_f = D_f/\frac{1}{2}\rho V^2$(unit area) is considerably simplified if we are willing to assume (1) the flow is incompressible, (2) there is no heat transfer from the surface to the flow, (3) the boundary layer is laminar, and (4) the pressure across the boundary layer is constant. With these assumptions the mass conservation Equation (1) has the same form as in the inviscid analysis. The x-momentum Equation (5) has a viscous stress term

$$\mu \frac{\partial^2 u}{\partial y^2}$$

added to it while the y-momentum equation reduces to $\partial P/\partial y = 0$. Thus the equations for this analysis are written

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{38}$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \frac{\partial^2 u}{\partial y^2} \quad .$$

Since $\partial P/\partial y = 0$, we can use the relationship between pressure and velocity in the free stream just outside the boundary layer to express

$$-\frac{1}{\rho} \frac{\partial P}{\partial x} \text{ as } u_e \frac{\partial u_e}{\partial x} \quad .$$

See for example Equation (30). With this, (38) becomes

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

<div align="right">(39)</div>

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = u_e \frac{\partial u_e}{\partial x} + v \frac{\partial^2 u}{\partial y^2} \ .$$

Now if we multiply the first equation of (39) by $(u_e - u)$ and subtract from this the second equation we obtain

$$\frac{\partial}{\partial x}\left[u(u_e - u)\right] + \frac{\partial}{\partial y}\left[v(u_e - u)\right] + \left(u_e - u\right)\frac{\partial u_e}{\partial x} + v \frac{\partial u_e}{\partial y} = -v \frac{\partial^2 u}{\partial y^2} \ . \quad (40)$$

The term $\partial u_e/\partial y = 0$ because u_e does not change in the y-direction in the boundary layer. If we now define

$$\delta^* = \int_0^\delta \left(1 - \frac{u}{u_e}\right) dy \ ,$$

<div align="right">(41)</div>

$$\theta = \int_0^\delta \frac{u}{u_e}\left(1 - \frac{u}{u_e}\right) dy$$

and rewrite (40) as

$$\frac{\partial}{\partial x}\left[u_e^2 \left(\frac{u}{u_e}\left[1 - \frac{u}{u_e}\right]\right)\right] + \frac{\partial}{\partial y}\left[v\,(u_e - u)\right] + u_e\left(1 - \frac{u}{u_e}\right)\frac{\partial u_e}{\partial x} = -v \frac{\partial^2 u}{\partial y^2}$$

or

$$\frac{\partial}{\partial x}\left[u_e^2 \frac{\partial \theta}{\partial y}\right] + \frac{\partial}{\partial y}\left[v\,(u_e - u)\right] + u_e \frac{\partial \delta^*}{\partial y}\frac{\partial u_e}{\partial x} = -v \frac{\partial^2 u}{\partial y^2} \ , \quad (42)$$

we can integrate with respect to y to obtain

$$\frac{\partial}{\partial x}\left[u_e^2\, \theta\right] + \int_0^\delta \left\{\frac{\partial}{\partial y}\left[v\,(u_e - u)\right]\right\} dy + u_e\, \delta^* \frac{\partial u_e}{\partial x} = v \left.\frac{\partial u}{\partial y}\right|_{y=0}. \quad (43)$$

Note that

$$\int_0^\delta v \frac{\partial^2 u}{\partial y^2} dy = v \left[\left.\cancel{\frac{\partial u}{\partial y}}^{\,0}\right|_{y=\delta} - \left.\frac{\partial u}{\partial y}\right|_{y=0}\right]$$

and by the first equation of (39)

78

$$\int_0^\delta \left\{ \frac{\partial}{\partial y} \left[v \left(u_e - u \right) \right] \right\} dy = \int_0^\delta \left\{ - \left(u_e - u \right) \frac{\partial u}{\partial x} + v \frac{\partial u_e}{\partial y} - v \frac{\partial u}{\partial y} \right\} dy$$

$$= \int_0^\delta \left\{ - \left(u_e - u \right) \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \int_0^\delta \frac{\partial u}{\partial x} \, dy \right\} dy$$

$$= \int_0^\delta \left\{ - \left(u_e - u \right) \frac{\partial u}{\partial x} \right\} dy + \int_0^\delta \left(u_e - u \right) \frac{\partial u}{\partial x} \, dy = 0.$$

Thus (43) is

$$\frac{d\theta}{dx} + \frac{2\theta + \delta^*}{u_e} \frac{du_e}{dx} = \frac{\nu}{u_e^2} \frac{\partial u}{\partial y}\bigg|_{y=0} . \tag{44}$$

We should note that (44) is only an approximate description of boundary layer flow because the process of integration used to obtain (43) in effect smooths out local departures from mean values. Further, although we intuitively understand what we mean by the outer edge of the boundary layer, the fact is we have never defined this value of y precisely. The approximate method, however, gives results which are within a few percent of those obtained with exact solutions in the few cases where the latter are known. For this reason and the ease with which it is applied to airfoil-like bodies, we shall find this approximate technique extremely useful. Empirical corrections to improve the agreement with experiment can be applied at the conclusion of the calculation if desired.

To complete the solution of (44), that is to find $\delta^*(x)$ and $\mu \frac{\partial u}{\partial y}\big|_{y=0}$ (x) which we need to calculate the form drag and the skin friction drag respectively, we need an expression for u(y). We shall assume, following Pohlhausen, that we can represent this function by the polynomial

$$\frac{u}{u_e} = A\eta + B\eta^2 + C\eta^3 + D\eta^4 \tag{45}$$

where $\eta = y/\delta$. We choose this polynomial because it is the lowest order polynomial which can represent the essential character of what we know of boundary layer flow and is the highest order polynomial for which we can easily evaluate the constants. The constants are found from the boundary conditions:

$$\text{at} \quad y = 0, \quad u = 0; \quad \frac{dp}{dx} = -\rho u_e \frac{du_e}{dx}$$

$$y = \delta, \quad u = u_e; \quad \frac{\partial u}{\partial y} = 0; \quad \frac{\partial^2 u}{\partial y^2} = 0 . \tag{46}$$

The result is that

$$A = 2 + \frac{1}{6} \frac{\delta^2}{\nu} \frac{du_e}{dx} ; \qquad B = -\frac{1}{2} \frac{\delta^2}{\nu} \frac{du_e}{dx}$$

$$C = -2 + \frac{1}{2} \frac{\delta^2}{\nu} \frac{du_e}{dx} ; \qquad D = 1 - \frac{1}{6} \frac{\delta^2}{\nu} \frac{du_e}{dx} . \tag{47}$$

79

Substitution of (47) into (45) yields

$$\frac{u}{u_e} = 2\eta - 2\eta^3 + \eta^4 + \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{dx}\left(\eta - 3\eta^2 + 3\eta^3 - \eta^4\right)$$

or

$$\frac{u}{u_e} = \left[2 + \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]\eta - \frac{1}{2}\frac{\delta^2}{\nu}\frac{du_e}{dx}\eta^2 - \left[2 - \frac{1}{2}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]\eta^3 + \left[1 - \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]\eta^4.$$

$$(48)$$

du_e/dx of course is found from the potential, or inviscid, solution. Equation (48) may be substituted into (41) to obtain

$$\delta^* = \delta\left[\frac{3}{10} - \frac{1}{120}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]$$

$$(49)$$

$$\theta = \delta\left[\frac{37}{315} - \frac{1}{945}\frac{\delta^2}{\nu}\frac{du_e}{dx} - \frac{1}{9072}\left(\frac{\delta^2}{\nu}\frac{du_e}{dx}\right)^2\right].$$

The local shearing stress on the surface, $\mu\left.\frac{\partial u}{\partial y}\right|_{y=0}$, is simply

$$\frac{\mu u_e}{\delta}\left[2 + \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right].$$

$$(50)$$

To complete the numerical evaluation of δ^* and the shearing stress, $\tau_{y=0}$, we substitute (49) and (50) into (44) to obtain

$$\frac{37}{315}\frac{d\delta}{dx} - \frac{3\delta^2}{945\,\nu}\frac{du_e}{dx}\frac{d\delta}{dx} - \frac{5\delta^4}{9072\,\nu^2}\left(\frac{du_e}{dx}\right)^2\frac{d\delta}{dx} - \frac{1}{945}\frac{\delta^2}{\nu}\frac{d^2u_e}{dx^2} - \frac{2\delta^4}{9072\,\nu^2}\frac{du_e}{dx}\frac{d^2u_e}{dx^2}$$

$$+ \frac{2\delta}{u_e}\left[\frac{37}{315} - \frac{1}{945}\frac{\delta^2}{\nu}\frac{du_e}{dx} - \frac{1}{9072}\left(\frac{\delta^2}{\nu}\frac{du_e}{dx}\right)^2\right]\frac{du_e}{dx} + \frac{\delta}{u_e}\left[\frac{3}{10} - \frac{1}{120}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]\frac{du_e}{dx}$$

$$= \frac{\mu}{\rho u_e\delta}\left[2 + \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]$$

or

$$\left[\frac{37}{315} - \frac{3\delta^2}{945\,\nu}\frac{du_e}{dx} - \frac{5\delta^4}{9072\,\nu^2}\left(\frac{du_e}{dx}\right)^2\right]\frac{d\delta}{dx} - \frac{1}{945}\frac{\delta^2}{\nu}\frac{d^2u_e}{dx^2} - \frac{2\delta^4}{9072\,\nu^2}\frac{du_e}{dx}\frac{d^2u_e}{dx^2}$$

$$+ \frac{2\delta}{u_e}\left[\frac{37}{315} - \frac{1}{945}\frac{\delta^2}{\nu}\frac{du_e}{dx} - \frac{1}{9072}\left(\frac{\delta^2}{\nu}\frac{du_e}{dx}\right)^2\right]\frac{du_e}{dx} + \frac{\delta}{u_e}\left[\frac{3}{10} - \frac{1}{120}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right]\frac{du_e}{dx}$$

$$= \frac{\mu}{\rho u_e\delta}\left[2 + \frac{1}{6}\frac{\delta^2}{\nu}\frac{du_e}{dx}\right].$$

$$(51)$$

Equation (51) is a highly non-linear first-order ordinary differential equation
with δ the dependent variable and x the independent variable. Note that u_e,
du_e/dx, d^2u_e/dx^2 must all be supplied as data from the external, or potential,
solution and the value of each of these quantities depends upon the particular
value of x. The equation may be solved by predicting $d\delta/dx$, solving the
resultant polynomial for δ and then checking to be sure that the predicted
value of $d\delta/dx$ is sufficiently close to that found. If not, the process is
repeated until sufficient accuracy is achieved. In the interests of speed
and accuracy the solution is best obtained on a digital computer. The solution
will be in the form of δ for each value of x along the chord for each surface.
By the use of the first equation of (49) and by (50) one can find then $\delta^*(x)$
and $\tau_{y=0}(x)$. Schlichting (Ref. 65) gives the more common evaluation of (51).

As a check on the solution technique one can look at (51) as applied to
a flat plate at zero angle of attack. Under these circumstances (51) reduces
to

$$\frac{37}{315} \frac{d\delta}{dx} = \frac{2\mu}{\rho u_e \delta} .$$

(52)

This has the solution

$$\frac{\delta^2}{2} = \frac{630}{37} \frac{x\mu}{\rho u_e}$$

or

$$\delta = \frac{5.84 \; x}{\sqrt{\frac{\rho u_e x}{\mu}}} = \frac{5.84 \; x}{\sqrt{R_{e_x}}} .$$

(53)

This overestimates the value of δ by about 1.8% compared with more exact
solutions. δ^* is therefore also high by about 1.2%.

The effect of du_e/dx is to reduce δ for a given x when du_e/dx is
positive and increase it when du_e/dx is negative. Both types of behavior
will be present on airfoils. From the leading edge to the crest on the upper
surface of the airfoil du_e/dx will be positive. From the crest aft, du_e/dx
will be negative. The boundary layer will therefore be very thin up to the
crest and will begin to grow rapidly downstream of that point.

If the du_e/dx is sufficiently negative and persists for a sufficient
extent of x it will cause the boundary layer to leave the surface. This
condition is manifest to the pilot as a stall or loss of lift accompanied by
a sharp increase in drag. Usually, the flight Reynolds number, airfoil
geometry, and surface condition are such that the boundary layer becomes
turbulent before separation occurs so that we will postpone our discussion
of how to calculate the aerodynamic characteristics near maximum lift.

Assuming we now have the value of $\delta^*(x)$ as calculated above, how do we
use it to correct the surface pressure values obtained from the inviscid
computation? By adding $\delta^*_{upper \; surface}$ to the actual upper surface
ordinates and $\delta^*_{lower \; surface}$ to the actual lower surface ordinates we
obtain a new fatter airfoil. We submit these new ordinates to the inviscid
computation procedure and obtain a new pressure distribution. If du_e/dx

is substantially different from what we used to compute $\delta^*(x)$ we must use the new values of du_e/dx in a predictor-corrector scheme to compute a new $\delta^*(x)$. The process must be repeated until the values of du_e/dx obtained from the inviscid computation agree with what we used in computing $\delta^*(x)$. Generally, as speed and aircraft size increase, fewer iterations will be required to achieve satisfactory agreement because the boundary layer will be proportionately thinner.

As we have noted there are a number of things which can cause a laminar boundary layer to become turbulent: existence of a large Reynolds number, surface roughness, and to some degree the sign on du_e/dx. Because wings on light aircraft will usually experience a turbulent boundary layer on at least some portion of their surfaces it is necessary to examine how the procedure to calculate δ^* and $\tau_{y=0}$ are altered for this conditon and how one determines when to change from laminar to a turbulent calculation. It should be recognized at the outset that turbulent motion is a very complex phenomenon, never successfully treated in a completely analytical fashion. It is necessary therefore to employ rather crude analytical models or semi-empirical correlations in order to retain the usual equations of motion (38) as the describing equations of the fluid behavior. Following this approach we observe that Equation (44) may also be used to represent turbulent boundary layer flow provided we use a suitable relation for skin friction in place of

$$\mu \left.\frac{\partial u}{\partial y}\right|_{y=0}$$

and a consistent expression for $u(y)$. One empirical formula for $\tau_{y=0}$ which finds considerable use is that due to Ludwieg and Tillmann (as quoted in Ref. 65):

$$\tau_{y=0} = \frac{\rho u_e^2 \left(0.123 \times 10^{-0.678 \frac{\delta^*}{\theta}}\right)}{\left(\frac{u_e \theta}{\nu}\right)^{0.268}} \quad . \tag{54}$$

The relation for the velocity profile commonly used in related studies is

$$\frac{u}{u_e} = \left(\frac{y}{\delta}\right)^{\frac{1}{n}} \quad , \tag{55}$$

where n is between 4 and 6, but is usually taken to be the latter value.

The use of (54) in (44) still leaves the equation non-integrable because an explicit relation between δ^* and θ has not been given. This could be developed from the definitions of θ and δ^*, (41) and (55), or the problem skirted by proceeding as follows: Multiply the second equation of (39) by u and then integrate with respect to y. We obtain what might be termed an energy integral equation

$$\frac{1}{u_e^3} \frac{d}{dx}\left[u_e^3 \delta^{**}\right] = \int_0^\delta \frac{\tau_{y=0}}{\rho u_e^2} \frac{\partial}{\partial y}\left(\frac{u}{u_e}\right) dy \quad , \tag{56}$$

82

analogous to the momentum integral Equation (44). The name of course refers to the fact that momentum times velocity has the units of energy. Note that

$$\delta^{**} = \int_0^\delta \frac{u}{u_e}\left(i - \frac{u^2}{u_e^2}\right) dy \quad .$$

(57)

It has been found experimentally that there exists a unique relationship between δ^*/θ and δ^{**}/θ which can be expressed as

$$\frac{\delta^{**}}{\theta} = \frac{1.269\frac{\delta^*}{\theta}}{\frac{\delta^*}{\theta} - 0.379} \quad .$$

(58)

Experiments have also led to the conclusion that

$$\int_0^\delta \frac{\tau_{y=0}}{\rho u_e^2}\frac{\partial}{\partial y}\left(\frac{u}{u_e}\right) dy = \frac{0.56 \times 10^{-2}}{\left(\frac{u_e\theta}{\nu}\right)^{1/6}} \quad .$$

(59)

Substitution of these experimental results, (58) and (59), into (56) yields

$$\frac{1}{u_e^3}\frac{d}{dx}\left[\frac{1.269\, u_e^3\frac{\delta^*}{\theta}\theta}{\frac{\delta^*}{\theta} - 0.379}\right] = \frac{0.56 \times 10^{-2}}{\left(\frac{u_e\theta}{\nu}\right)^{1/6}}$$

(60)

while (54) substituted into (44) gives

$$\frac{d\theta}{dx} + \frac{2\theta + \delta^*}{u_e}\frac{du_e}{dx} = \frac{0.123 \times 10^{-0.678\frac{\delta^*}{\theta}}}{\left(\frac{u_e\theta}{\nu}\right)^{0.268}} \quad .$$

(61)

Simultaneous solutions of (60) and (61) will yield $\delta^*(x)$ and $\theta(x)$. When these results are substituted into (54) one has $\tau_{y=0}(x)$. The skin friction drag is then computed by integrating $\tau_{y=0}(x)$ over both surfaces:

$$D_f = \left[\int_0^c \tau_{y=0}(x)\, dx\right]_{\text{upper surface}} + \left[\int_0^c \tau_{y=0}(x)\, dx\right]_{\text{lower surface}} \quad .$$

(62)

The dimensions of D_f are force per unit span. As noted previously, at low angles of attack, the drag as computed by (62) should be about 4 times that found by (35). As α increases, the form drag tends to predominate. The total drag of course is the sum of the drags calculated from (35) and (62).

The correct expression for $\tau_{y=0}$ in (62) depends, as has been indicated, on whether the boundary layer is laminar or turbulent or some combination thereof. Generally one would expect that the boundary layer is laminar over the forward

portion of the wing and then changes—goes through what is called transition—to a turbulent boundary layer. Thus, Equation (50) would give the correct expression for $\tau_{y=0}$ upstream of the transition point and (54) gives the correct expression for $\tau_{y=0}$ downstream of this point. Since the boundary layer already has a finite thickness at the transition point, one chooses as a starting point for the turbulent calculation that point which will give the same δ at the transition point as the laminar solution beginning at the leading edge. The laminar values of $\tau_{y=0}$ and δ^* are used up to transition and the turbulent values downstream. They are approximately the same at transition.

The beginning of transition has been found to occur at a Reynolds number between 3×10^5 and 4×10^6. This is a very substantial range. As a point of reference consider that the Reynolds number per foot of chord for a airplane flying at 200 ft/sec at sea level is 1.275×10^6. Transition could, according to this criterion, occur anywhere from 4" from the nose to 3 ft from the nose. Since the chord for most light aircraft is at least 4 ft, the boundary layer on the aft portion of the wing will always be turbulent. Whether the transition begins 4" from the nose or 3 ft from the nose depends upon such things as surface roughness, free stream turbulence, and du_e/dx. The latter influence, however, is the only one which can be determined *a priori*, that is, before the wing is built and flown under particular conditions. It is, therefore, the only one we will attempt to evaluate.

A laminar boundary layer is said to be unstable—that is, it tends to become turbulent—when a velocity disturbance in this boundary layer can grow. Tollmein was able to show that a necessary and sufficient condition for neutral stability of disturbances in laminar boundary layers is the existence of a point of inflection in the boundary layer's velocity profile, $u(y)$.

Using a sixth order polynomial to represent the velocity profile, Schlichting and Ulrich were able to plot a relationship between the value of $u_e\delta^*/\nu$ for which an inflection point exists and $\delta^2 du_e/\nu dx$. With this plot one can take the values of δ, δ^*, u_e, du_e/dx, and ν and determine whether or not the boundary layer is unstable. The precise distance between the onset of transition and the point of neutral stability as determined above depends upon the rate of amplification of disturbances in the boundary layer and consequently upon du_e/dx in that region. It has been found that plotting experimental data on a graph where

$$\left(\frac{u_e\theta}{\nu}\right)_{transition} - \left(\frac{u_e\theta}{\nu}\right)_{neutral \ stability}$$

is the ordinate and

$$\frac{\int_{x_{n.s.}}^{x_{tr}} \frac{\theta^2}{\nu} \frac{du_e}{dx} dx}{x_{tr} - x_{n.s.}}$$

84

is the abscissa leads to excellent correlation of the data. For a laminar boundary layer we know θ, u_e, du_e/dx, and $(\theta^2/\nu)(du_e/dx)$ as functions of x. We also know $(u_e\theta/\nu)_{n.s.}$ because we know the relationship between θ and δ^*. Marching downstream of the neutral stability point we can easily find a $u_e\theta/\nu$ and an $x_{tr} - x_{n.s.}$ for each point. The other data then permits us to locate a point on the graph for each value of x. The first point which falls on or above the data correlation is taken as the x-location for which transition has taken place.

In addition to the change in δ^* and $\tau_{y=0}$ as one goes from laminar to turbulent boundary layer another type of behavior associated with boundary layer flows which significantly affects the lift, drag and moment characteristics of the airfoil in boundary layer separation is usually identified by the disappearance of the local skin friction, *i.e.* when

$$\frac{du}{dy}\bigg|_{y=0} = 0 \ .$$

The geometry of very thin airfoils is such that regions of laminar separation or separation followed by reattachment confined to the front half of the airfoil are possible at moderate angles of attack. However, light aircraft operating at moderate Mach numbers can be expected to employ airfoils of 12% or greater thickness for which this type of phenomenon is not to be expected. The separation characteristic of thick airfoils is a turbulent separation from the region of the trailing edge. Thus, if one terminates the calculation of δ^* and $\tau_{y=0}$ at that angle of attack for which $\tau_{y=0} = 0$ over a significant portion of the airfoil, he is reasonably assured of having closely approached $C_{L_{max}}$. This is about the most one could expect of the procedure outlined above.

Perhaps it would now be appropriate to review briefly and comment upon the procedure for estimating the aerodynamic characteristics of airfoils as developed up to this point.

1. We first locate the vortex filaments, taking care that we space them sufficiently close together to represent the surface accurately. This is particularly important in regions of high curvature and/or regions where the surface slope is significantly different from the free stream flow direction.

2. Equations (29) are then solved for a particular angle of attack to obtain the values of all Γ_N. These values are then substituted into (19) to obtain the net induced velocities and the surface pressures computed from (33).

3. The lift and drag are evaluated by (34) and (35). The drag at this point should be zero.

4. $u_e(x)$ is evaluated from (32) for both surfaces. Data smoothing procedures are employed to insure that the results represent physical reality as closely as possible with as few inflection points as possible. du_e/dx and d^2u_e/dx^2 are then calculated numerically.

5. $\delta(x)$ for both surfaces is then found from a solution to (57). $\delta^*(x)$ is obtained from (49) and $\tau_{y=0}$ from (50).

6. The location of transition is identified and the turbulent boundary layer computation begun using (60) and (61) and then (54). At the conclusion of this process one has complete values of $\delta^*(x)$ and $\tau_{y=0}(x)$ for both surfaces.

7. $\delta^*(x)$ is then added to the physical ordinates of the airfoil along with artificial trailing edge found by extending the chord

$$\delta^*(c)\bigg/\frac{dy}{dx}\bigg|_{x=c} .$$

8. The previous seven steps are then repeated for the new ordinates.

9. The new value of du_e/dx is compared with the value obtained from step 4. If they differ by more than a few percent, we modify our estimates of $\delta^*(x)$ according whether du_e/dx seems to favor larger or smaller values of δ^*.

10. The process is repeated until the final value of du_e/dx would give the same $\delta^*(x)$ as used in the inviscid calculation. Obviously, care must be taken to insure that adequate precision is maintained during such an extensive series of computations, else the resulting numbers are meaningless.

11. The procedure is valid for a Mach number of zero. Since most light aircraft operate at Mach numbers not far above zero, the simple Prandtl-Ghauert correction to the pressures for Mach number effects is usually quite adequate:

$$P_x = P_\infty + \frac{(P_x - P_\infty)_{M=0}}{\sqrt{1 - M_\infty^2}} . \tag{63}$$

12. The corrected pressure is then applied to the physical boundaries of the airfoil to obtain the lift and form drag according to (34) and (35). The skin friction drag is computed with (62) using the most updated value of $\tau_{y=0}(x)$.

13. The pitching moment about the leading edge can be computed by integrating the product of the pressure forces and the distance from the nose

$$M = \oint P(x) \ x \ \cos\left[\tan^{-1}\frac{dy}{dx}\bigg|_x\right]dx + \oint P(x) \ y \ \sin\left[\tan^{-1}\frac{dy}{dx}\bigg|_x\right]dx. \tag{64}$$

There will also be a small contribution to the moment from the skin friction forces but these are (a) generally in the chordwise direction and (b) the moment generated by the friction on the upper surface opposes that generated by the friction on the lower surface. As a result the net contribution is generally small enough to neglect.

EXTENSION TO THREE DIMENSIONS: THE FINITE WING

The theoretical procedure described above provides a means for estimating the aerodynamic characteristics of a section of an infinite wing. Real wings, of course, have to have ends and frequently, for reasons that will become apparent as the discussion proceeds, are tapered, twisted, and change airfoil sections with changes in span locations.

Some means must therefore be found to account for such influences in determining the aerodynamic characteristics of complete wings. The principal considerations in wing design are (1) low drag for good performance, (2) light weight with adequate strength for good payload capacity, and (3) fairly simple structure with few changes in shape for low cost in manufacturing. We shall be concerned here with means for calculating the lift, drag and moments of wings which we will assume satisfy the latter two criteria. We will also limit our consideration to unswept wings of moderate-to-high span-to-chord (aspect) ratio. Further, in order to keep the computation to a reasonable length we would like the technique we employ to use as much of the previous result as possible. In order to do so we ask the question, what are the effects of ends, twist, and taper? If an untapered wing has the same airfoil over its entire span then it would seem reasonable to conclude that, at least near the center, twisting the wing has the effect of changing the local angle of attack. It will be recalled that the two-dimensional calculation is carried out for specific angle of attack. Generally, both lift and drag increase with increasing angle of attack. Twist will therefore change the local lift and drag values.

The existence of a wing tip permits high pressure air from the lower surface of the wing to flow up around the tip to the low pressure regions on the upper surface. The result is that the lift in the tip regions is reduced and a vortex filament is created. This filament begins near the tip and extends downstream in a plane parallel to the fuselage. The fact that in theory vortices must be closed or infinite led to the thought that since a vortex is actually observed extending rearward from the tip regions of lifting wings, some sort of vortex system must also extend from tip to tip. If this is true, perhaps one could actually represent such a wing by a series of "horseshoe"-like vortices which "roll up" in the tip regions to form a single vortex extending to infinity. A series of horseshoe vortices with different strengths would enable one to represent a variety of lift distributions on a wing. If the wing has a moderate-to-high aspect ratio, little or no sweep, and moderate dihedral then it would seem that one could take the various vortex filaments as being co-linear with little error. Such an assumption, the so-called lifting-line theory of Prandtl, obviously leads to fewer mathematical difficulties than having to consider a chord-wise lattice of vortices or a lifting surface. Because most light aircraft wings meet the criteria for applicability of the lifting line theory and because we intend to use the theory only to modify our 2-dimensional results, we expect the procedure to give us accurate results.

It will be observed that a horseshoe system with the trailing vortex's velocity moving inboard over the upper surface and outboard under the lower surface induces an upwash component ahead of the wing and a downwash component behind the wing. This combination of upwash ahead and downwash

behind the wing is, in effect, a change in local angle of attack in so far as the flow facing a section of wing is concerned. The finite span of the wing can therefore be thought of as producing the same effect as does geometric twist. For consistency it would seem to be convenient to attempt to represent the effects of taper also as a change in local angle of attack. If we are successful in this endeavor, we have reduced the entire procedure for calculating the lift, drag, and moment characteristics of complete wings to one of (1) finding the effective local angle of attack by the lifting line theory, (2) determining the two-dimensional or section characteristics corresponding to these effective local values of α by reference to equations or tables computed by the methods discussed above, and then (3) integrating these section characteristics over the entire wing area to average the values.

In current designs, wings are not free entities but are attached to large structures such as fuselages. It is important, therefore, to model the fuselage-wing junction region in such a manner that the effects of its presence can be handled within the framework of a method to calculate finite wing characteristics. We do this by mathematically transforming this region of the fuselage into a part of the wing. Other examples of the use of mathematical transformations to simplify the analysis of complex problems are well known. Problems involving flow through pipes, structural analysis of tubes, etc. become much simpler when transformed from Cartesian or rectangular coordinates to cylindrical coordinates. Differential equations become algebraic when transformed from the time domain to the frequency domain. Complex shapes, such as airfoils, can be transformed into circles about which the flow behavior is well known. The flow at any point about an airfoil can then be found by locating the equivalent point on the circle.

Generally, in selecting a transformation we seek one which either simplifies the mathematical representation of a physical situation or, as in the present case, distorts a complex shape into a simpler shape for which the physical phenomena are well understood and easily analyzed. We then go through an _inverse_ transform to find how the well-understood behavior distorts in going back to the original physical situation. Devising a suitable transform is usually a trial and error process, guided by experience, skill, and to some extent, luck. Certain mathematical requirements must also be met, dependent upon the framework in which the transform is used. The transform used here was devised by H. Multhopp (Ref. 51).

The thought processes followed by Multhopp in devising the transform shown below for elliptical cross-section fuselages probably included these elements: Under the Joukowski transform,

$$\zeta = z + \frac{c^2}{z} \quad ,$$

a circle in the z-plane with its center at the origin transforms to a flat plate along the ξ-axis in the ζ-plane for positive values of c^2. If one takes $c^2 = -1$, then

88

$$\zeta = \xi + i\eta = x + iy - \frac{1}{x + iy}$$

$$= x + iy - \frac{x - iy}{x^2 + y^2}$$

$$= x\left(1 - \frac{1}{x^2 + y^2}\right) + iy\left(1 + \frac{1}{x^2 + y^2}\right)$$

and the circle (of radius 1.0) transforms into a flat plate along the η-axis in the ζ-plane. From this result one concludes that a similar transform, suitably modified to account for elliptical fuselages and variable placement of the wing with respect to the origin, should produce the requisite figure.

Consider also the ellipse shown in the sketch below:
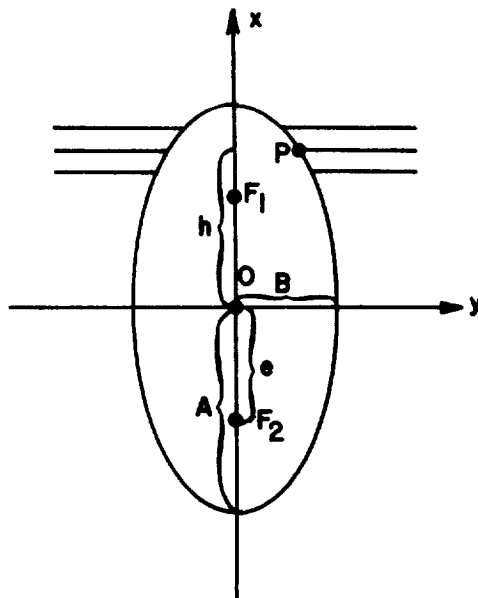
The properties of an ellipse are such that

$$A^2 - B^2 = e^2 \quad,$$

$$\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1 \quad,$$

and

$$\overline{F_1 P} + \overline{F_2 P} = 2A \quad.$$

From the last statement it follows that

$$2A = \sqrt{y^2 + (h - e)^2} + \sqrt{y^2 + (h + e)^2}$$

from which one can obtain

$$A^2 = y^2 + h^2 + e^2$$

or

$$B^2 = y^2 + h^2 \quad .$$

For points <u>outside</u> the ellipse we can generalize this by writing

$$a^2 = y^2 + h^2 + e^2$$

$$b^2 = y^2 + h^2$$

$$e^2 = a^2 - b^2 \quad .$$

It has been a feature of textbooks on hydrodynamics for some years to show that the equation for mapping an ellipse in the ω-plane into a circle of radius $\frac{1}{2}(A + B)$ in the z-plane is

$$z = \tfrac{1}{2} (\omega + \sqrt{\omega^2 - e^2}) \quad .$$

Thus to map the ellipse into a flat plate we write

$$\zeta = z + \frac{\tfrac{1}{4}(A + B)^2}{z} = \tfrac{1}{2} (\omega + \sqrt{\omega^2 - e^2}) + \frac{\tfrac{1}{2}(A + B)^2(\omega - \sqrt{\omega^2 - e^2})}{\omega^2 - \omega^2 + e^2}$$

$$= \frac{(A^2 - B^2)}{2e^2} \left(\omega + \sqrt{\omega^2 - e^2}\right) + \frac{(A + B)^2}{2e^2} \left(\omega - \sqrt{\omega^2 - e^2}\right)$$

$$= \left(\frac{A^2 - B^2}{2e^2} + \frac{A^2 + 2AB + B^2}{2e^2}\right)\omega + \left(\frac{A^2 - B^2}{2e^2} + \frac{A^2 + 2AB + B^2}{2e^2}\right)(\sqrt{\omega^2 - e^2})$$

$$= \left(\frac{2A^2 + 2AB}{2e^2}\right)\omega + \left(\frac{-2AB - B^2}{2e^2}\right)\sqrt{\omega^2 - e^2}$$

$$= \frac{A}{A - B}\, \omega - \frac{B}{A - B}\sqrt{\omega^2 - e^2}$$

which is the form employed by Multhopp.

The figure below shows how the trace of the wing-fuselage combination in the y-z plane transforms from the physical u-plane to the $\bar{u}$-plane according to the relation

$$\bar{u} = \frac{1}{A' - B'} \left[A'u - B'\sqrt{u^2 - A'^2 + B'^2}\right] \quad . \tag{65}$$

u - plane

ū - plane
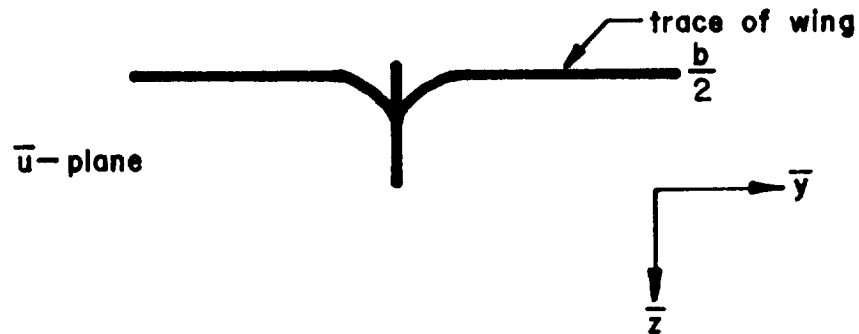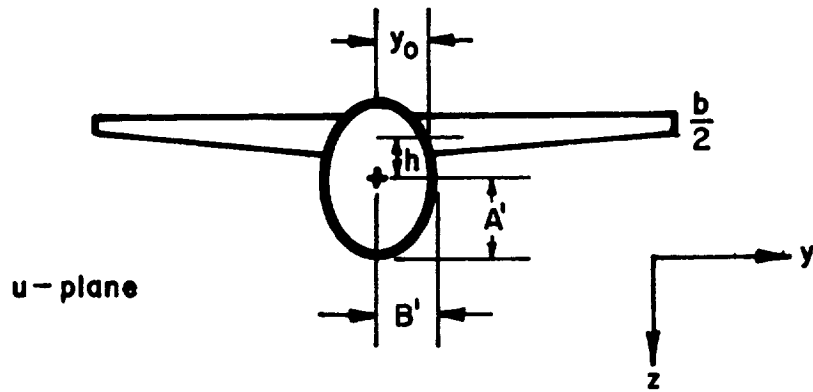
We use this transformation because it changes an elliptical fuselage into a thin vertical line which then provides no resistance to the vertical or induced component of flow over the wing. As we noted above, it is these vertical components which determine the magnitude of the local angle of attack. In essence, then, the transform distributes the flow components due to the fuselage over the span of the distorted wing. A similar transformation $\overline{u} = u + R^2/u$ is used for circular fuselages. More complex shapes can be handled in the same fashion through the use of a suitable transform. By writing u and $\overline{u}$ as complex variables

$$u = z + iy = a \cos \psi + ib \sin \psi$$

$$\overline{u}' = \overline{z} + i\overline{y}$$

(66)

and making the following definitions

$$a = \tfrac{1}{2}\left[\sqrt{y^2 + (h - e')^2} + \sqrt{y^2 + (h + e')^2}\right]$$

$$b = \sqrt{a^2 - e'^2}$$

(67)

$$e = \sqrt{A'^2 - B'^2} = \sqrt{a^2 - b^2}$$

91

then

$$\overline{u} = \frac{1}{A' - B'} \left[A'u - B'\sqrt{u^2 - e'^2} \right]$$

$$= \frac{1}{A' - B'} \left[A'u - B'\sqrt{u^2 - a^2 + b^2} \right] . \tag{68}$$

But

$$u^2 - a^2 + b^2 = a^2 \cos^2 \psi - b^2 \sin^2 \psi + 2abi \sin \psi \cos \psi - a^2 + b^2$$

$$= a^2(\cos^2 \psi - 1) - b^2(\sin^2 \psi - 1) + 2abi \sin \psi \cos \psi$$

$$= a^2 \sin^2 \psi + b^2 \cos^2 \psi + 2abi \sin \psi \cos \psi \tag{69}$$

$$= (b \cos \psi + ia \sin \psi)^2$$

therefore,

$$\overline{u} = \frac{1}{A' - B'} \left[A'u - B'(b \cos \psi + ia \sin \psi) \right]$$

$$= \frac{1}{A' - B'} \left[A'z + iA'y - B'b \cos \psi - iB'a \sin \psi \right] \tag{70}$$

$$= \frac{1}{A' - B'} \left[(A'z - B'b \cos \psi) + i(A'y - B'a \sin \psi) \right] = \overline{z} + i\overline{y} .$$

Comparing real and imaginary parts, one finds that

$$\overline{y} = \frac{1}{A' - B'} \left[A'y - B'a \sin \psi \right] . \tag{71}$$

Since

$$y = b \sin \psi = \sqrt{a^2 - e'^2} \sin \psi$$

then

$$\sin \psi = \frac{y}{\sqrt{a^2 - e'^2}} .$$

One may therefore write for $\overline{y}$

$$\overline{y} = \frac{y}{A' - B'} \left[A' - B' \frac{a}{\sqrt{a^2 - e'^2}} \right] . \tag{72}$$

This relationship determines how points along the span in the physical or u-plane transform into the $\overline{u}$-plane.

92

We seek now to find how the flow in the neighborhood of the fuselage influences the flow direction and magnitude at each station along the wing's span. The real part of $d\bar{u}/du$ in effect distributes the vertical components of the fuselage flow along the wing. If α_B is the fuselage angle of attack in the $\bar{u}$-plane then the induced* upwash along the span is given by

$$\Delta\alpha(y) = \left[R \frac{d\bar{u}}{du} - 1 \right] \alpha_B \quad . \tag{73}$$

Because of the nature of the transformation α_B will have the same value in either plane.

Now
$$\frac{d\bar{u}}{du} = \frac{1}{A' - B'} \left[A' - \frac{B'u}{\sqrt{u^2 - e'^2}} \right]. \tag{74}$$

The real part of $d\bar{u}/du$ is, of course, its vertical component:

$$\frac{d\bar{u}}{du} = \frac{1}{A' - B'} \left[A' - B' \frac{a \cos\psi + ib \sin\psi}{b \cos\psi + ia \sin\psi} \right]$$

$$= \frac{1}{A' - B'} \left[A' - B' \frac{(a \cos\psi + ib \sin\psi)(b \cos\psi - ia \sin\psi)}{(b \cos\psi + ia \sin\psi)(b \cos\psi - ia \sin\psi)} \right]$$

$$\therefore \quad R \frac{d\bar{u}}{du} = \frac{1}{A' - B'} \left[A' - B' \frac{ab(\cos^2\psi + \sin^2\psi)}{b^2 \cos^2\psi + a^2 \sin^2\psi} \right]$$

$$= \frac{1}{A' - B'} \left[A' - B' \frac{ab}{b^2 \left(1 - \frac{y^2}{b^2} + \frac{a^2 y^2}{b^2} \right)} \right] \tag{75}$$

$$= \frac{1}{A' - B'} \left[A' - B' \frac{\frac{a}{\sqrt{a^2 - e'^2}}}{1 + \frac{e'^2 y^2}{(a^2 + e'^2)^2}} \right] \quad .$$

If the wing is very thick at its junction with the fuselage then the actual $\Delta\alpha$ obtained is less than that predicted by (73). It has been suggested that one should therefore reduce (73) by a factor T, taken as constant across the wing span, which is the ratio of the body cross-sectional area above and below the wing to the total frontal area of the body. The area of the

---

* The amount by which the flow angularity exceeds that due to geometric inclination.

elliptical fuselage is of course $\pi A'B'$. The segment of the fuselage which represents a continuation of the wing has an area of approximately $2y_0 t_{root}$. Thus

$$T = 1 - \frac{2 \; y_0 \; t_{root}}{\pi A'B'} \qquad (76)$$

and a more general expression for (73) is

$$\Delta\alpha(y) = T \; \alpha_B \left[\frac{R}{} \frac{d\overline{u}}{du} - 1 \right] \; . \qquad (77)$$

We could also have written (77) as

$$\Delta\alpha(y) = \alpha_B \left[\left(\frac{R}{} \frac{d\overline{u}}{du} \right)_T - 1 \right] \qquad (78)$$

if we had known how to write

$$\left(\frac{R}{} \frac{d\overline{u}}{du} \right)_T$$

explicitly. However, by comparing (77) and (78) we find that as a first approximation

$$\left(\frac{R}{} \frac{d\overline{u}}{du} \right)_T = 1 + T \left[\frac{R}{} \frac{d\overline{u}}{du} - 1 \right] \; . \qquad (79)$$

In addition to the flow angularity induced along the wing by the presence of a fuselage there is also a flow angle (a downwash) induced by the lift associated with a finite wing. This angle in the $\overline{u}$-plane is written $\overline{\alpha}_i(\overline{y})$. We seek now to transform this angle into the u-plane so that we may see more easily its influence on the actual lift and drag of the local airfoil section. We note that the induced angle in the $\overline{u}$-plane multiplied by the real part of the change in $\overline{u}$ for a given change in u is just the induced angle in the u-plane. Thus for thick airfoils

$$\alpha_i(y) = \overline{\alpha}_i(\overline{y}) \left[\frac{R}{} \frac{d\overline{u}}{du} \right]_T \; . \qquad (80)$$

This angle is negative in the usual sense. The geometric angle of attack of the wing can be given in terms of the angle of attack at the root and the twist relative to the root angle as a function of span: $\alpha_e(y) = \alpha_R + \epsilon(y)$. To the geometric angle we must add flow angularities due to body upwash, α_B, and due to wing lift, α_i. The result is that for thick wings the effective section angle of attack in the physical plane is given by

$$\alpha_e(y) = \alpha_R + \epsilon(y) + \left[\alpha_B - \overline{\alpha}_i(\overline{y}) \right] \left[1 + \left(1 - \frac{2 \; y_0 \; t_{root}}{\pi A'B'} \right) \right.$$
$$\left. \cdot \left(\frac{1}{A' - B'} \left\{ A' - B' \frac{\dfrac{a}{\sqrt{a^2 - e'^2}}}{1 + \dfrac{e'^2 \; y^2}{(a^2 - e'^2)^2}} \right\} \right) - 1 \right] \qquad (81)$$

94

with

$$a = \frac{1}{2} \left[\sqrt{y^2 + (h - A'^2 - B'^2)^2} + \sqrt{y^2 + (h + A'^2 - B'^2)^2} \right]$$

and

$$e = \sqrt{A'^2 - e'^2} \quad .$$

Note that with the exception of $\overline{\alpha}_i(\overline{y})$ all the quantities in (81) can be determined from the geometry of the design.

For the evaluation of $\overline{\alpha}_i(\overline{y})$ we employ a variant of the technique used to determine the inviscid velocity distribution about an airfoil section. Consider the sketch below.
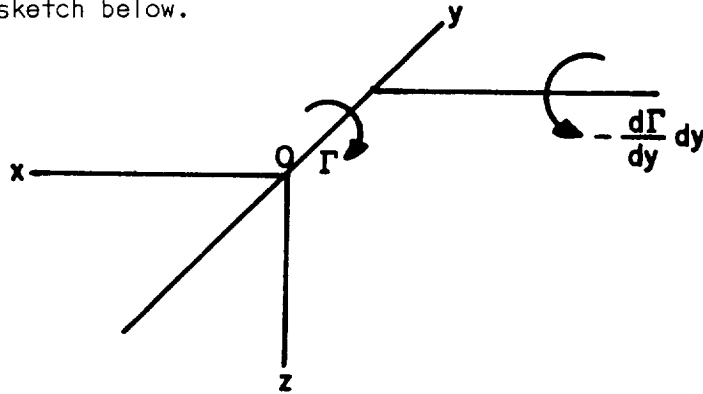
Recall also that we had indicated previously that we would represent a wing by a group of horseshoe-like vortices which physically "roll-up" at the tips to form single trailing vortices. Thus the circulation Γ will vary along the span, being symmetrical about the point 0 and falling to zero at the tips. Between the point y and y + dy on the span the circulation decreases by an amount

$$- \frac{d\Gamma}{dy} dy \quad .$$

Ideally, a trailing vortex of this strength springs from the element of span dy. There is therefore a sheet of trailing vortices extending across the span and the induced velocity normal to the free stream velocity must be obtained as the sum of the effects of all trailing vortices in this sheet.

To determine the form of the expression giving the sum of the effects of all trailing vortices consider first the case of a wing represented by a single horseshoe vortex. We see from Equation (15) that the velocity induced by a vortex at a point depends upon the distance from the point to the filament. In the following sketch the distance from P to the wing filament is PM. Now in Equations (15) the total velocity, $\sqrt{u^2 + v^2}$, is normal to the line, $\sqrt{x^2 + y^2}$, connecting the filament to the point at which one desires to know the velocity. One could therefore write

$$\sqrt{u^2 + v^2} = \frac{\Gamma}{2\pi\sqrt{x^2 + y^2}} \quad . \tag{82}$$

It seems reasonable to conclude that at any point the velocity induced by a semi-infinite vortex filament is half that induced by an infinite vortex. Let us ask then what is the velocity induced at a point by a small segment of a vortex filament. We know that the velocity depends upon the distance from the segment to the point. If we call $x^2 + y^2 = r^2$, then this distance from the vortex segment at A' is $r/\sin \theta_1$ where θ_1 is the angle PA'M in the sketch below.

Substitution of this expression for distance into an expression for the velocity induced at a point, say in the XOZ plane, by a segment of a semi-infinite vortex can be seen to yield

$$dV = \frac{\Gamma \sin \theta}{4\pi r} d\theta \quad . \tag{83}$$

Integration of (83) gives

$$V_1 = \frac{\Gamma}{4\pi} \left[- \cos \theta \, \Big|_{\theta_1}^{\pi/2} \right] = \frac{\Gamma}{4\pi} \cos \theta_1 \quad . \tag{84}$$

Calculation of the contribution to the velocity from the part of the filament beyond O yields

$$V_2 = \frac{\Gamma}{4\pi} \cos \theta_2 \tag{85}$$

so that the total induced velocity is given by

$$V = \frac{\Gamma}{4\pi} (\cos \theta_1 + \cos \theta_2) \quad . \tag{86}$$

In the notation of the sketch, the downwash velocity at point P (normal to PM) may be written
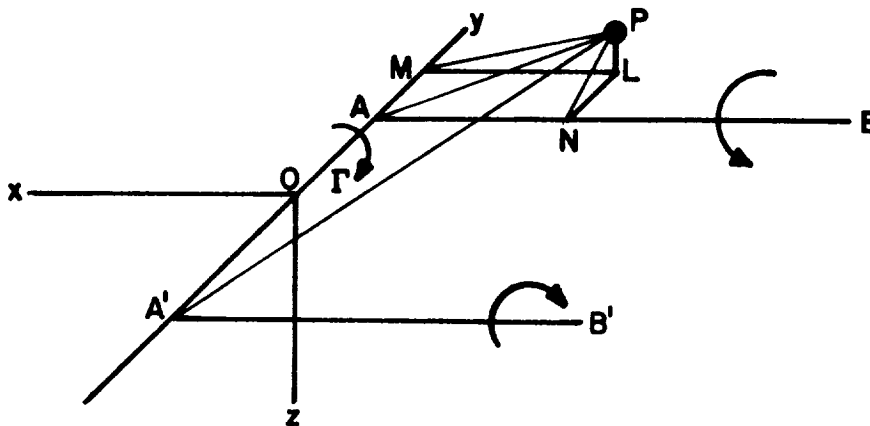
$$V = \frac{\Gamma}{4\pi PM} \left[\cos PA'A + \cos PAA' \right].$$ (87)

If we now assume that point P is located along the span at, say, y_1, then the distance from y_1 to any other point is $y_1 - y$. Further, if Γ is variable along y then

$$\Gamma = \int_{-b/2}^{b/2} \frac{d\Gamma}{dy} \, dy.$$

One may therefore write the induced velocity at some point y_1 along the span as

$$V \, \alpha_i(y_1) = \frac{1}{4\pi} \int_{-b/2}^{b/2} \frac{\frac{d\Gamma}{dy} \, dy}{y_1 - y} \quad .$$

In the $\bar{u}$-plane this is simply

$$\bar{\alpha}_i(\bar{y}_1) = \frac{1}{4\pi V} \int_{-b/2}^{b/2} \frac{\frac{d\Gamma}{dy} \, d\bar{y}}{\bar{y}_1 - \bar{y}} \quad .$$ (88)

If $\bar{\alpha}_i(\bar{y}_1)$ is to be in degrees, we multiply the right hand side of (88) by $180/\pi$.

Unfortunately the circulation Γ about an airfoil is not readily measured nor is it a quantity which is easily thought of in physical terms. More commonly, the characteristic of an airfoil is stated in terms of its lift coefficient, a quantity easily measured and important in aerodynamic and structural design. It can be shown on analytical grounds (Ref. 9) that the lift per unit span of a wing is $\rho V\Gamma$ where ρ is the air density. The lift coefficient is therefore

$$C_\ell = \frac{\rho V\Gamma}{\frac{1}{2}\rho V^2 c} = \frac{2\Gamma}{Vc} \quad ;$$

hence

$$\Gamma = \tfrac{1}{2} C_\ell Vc \quad .$$ (89)

Because both C_ℓ and c can vary as functions of y we put

$$\frac{d\Gamma}{dy} = \frac{V}{2} \frac{d(C_\ell c)}{dy}$$

or in non-dimensional form

$$\frac{d\Gamma}{dy} = \frac{Vb}{2} \frac{d\left(\frac{C_\ell c}{b}\right)}{dy} \qquad (90)$$

with (90), (88) becomes

$$\overline{\alpha}_i(\overline{y}_1) = \frac{180\overline{b}}{8\pi^2} \int_{-\overline{b}/2}^{\overline{b}/2} \frac{\left[\frac{d\left(\frac{C_\ell \overline{c}}{\overline{b}}\right)}{d\overline{y}}\right]}{\overline{y}_1 - \overline{y}} \, d\overline{y} \quad . \qquad (91)$$

where $\overline{\alpha}_i(\overline{y}_1)$, is now in degrees.

We note that the transformation (65) is an analytic function (and therefore conformal). It does not, however, affect geometric quantities in the chord-wise direction so that $\overline{c} = c$. Wing twist is also not affected, nor does the transformation affect quantities such as the local value of the circulation, Γ. In other words, the circulation or lift that exists at y in the u-plane is the same circulation or lift that exists at $\overline{y}$ in the $\overline{u}$-plane:

$$C_\ell(y)c = \overline{C}_\ell(\overline{y})\overline{c} \quad . \qquad (92)$$

The key to the evaluation of Equation (81) is therefore a suitable expression for $C_\ell(y)c$, the spanwise lift distribution. Unfortunately we do not know *a priori* what it is. We do know from the two-dimensional values calculated previously and the problem geometry what lift we would obtain if $\overline{\alpha}_i(\overline{y})$ could be neglected. We have only to find $\alpha_e(y)$ and the corresponding C_ℓ comes from the two-dimensional data. We can assume some modification to the local angle of attack for 3-dimensional effects and see if our calculated value of $\overline{\alpha}_i(\overline{y})$ is equal to the assumed value. If not, we can modify our assumption until it is. What we seem to require then is a systematic procedure for doing this.

Let us represent $C_\ell(y)c/b$ by a series,

$$\frac{C_\ell(y)c}{\overline{b}} = \sum_{n=1}^{\infty} A_n \sin\left[n \cos^{-1}\left(\frac{-2\overline{y}}{\overline{b}}\right)\right] \qquad (93)$$

and call $\overline{\theta} = \cos^{-1}(2\overline{y}/\overline{b})$. Then (91) could be written

$$\overline{\alpha}_i(\overline{y}_1) = \frac{180}{4\pi^2} \int_0^{\pi} \frac{\sum_{n=1}^{\infty} n A_n \cos n\overline{\theta}}{\cos \overline{\theta} - \cos \overline{\theta}_1} \, d\overline{\theta} \quad . \qquad (94)$$

Since it can be shown that

$$\int_0^{\pi} \frac{\cos n\theta}{\cos \theta - \cos \phi} \, d\theta = \pi \frac{\sin n\phi}{\sin \phi} \qquad (95)$$

98

$$\overline{\alpha}_i(\overline{y}_1) = \frac{180}{4\pi} \left[\frac{\displaystyle\sum_{n=1}^{\infty} n\, A_n \sin n\overline{\theta}_1}{\sin \overline{\theta}_1} \right] . \tag{96}$$

To proceed with the evaluation of (96) one returns for the moment to (93). Let us assume that we can adequately represent the lift at each of m evenly spaced points along the span with a finite series of r - 1 terms. m and r we assume are related by $\overline{\theta} = m\pi/r$ and m = 1, 2,...,r - 1. In other words, we divide the range $0 \le \overline{\theta} \le \pi$ into r intervals. Over each interval, the lift is a constant which we can represent by a finite series, the number of terms of which being also the number of points at which we calculate the lift. Thus, to use a large number of terms in the series to represent the lift in each interval, it is necessary to take a large number of intervals. The lift in each interval is written

$$\left(\frac{C_\ell(y)c}{b} \right)_m = \sum_{n=1}^{r-1} A_n \sin n\, \frac{m\pi}{r} \tag{97}$$

for which

$$A_n = \frac{2}{r} \sum_{m=1}^{r-1} \left(\frac{C_\ell c}{b} \right)_m \sin n\, \frac{m\pi}{r} . \tag{98}$$

Substitution of this result into (96) yields

$$\overline{\alpha}_i(\overline{y}_1) = \frac{180}{4\pi \sin \overline{\theta}_1} \left\{ \sum_{n=1}^{\infty} n \left[\frac{2}{r} \sum_{m=1}^{r-1} \left(\frac{C_\ell c}{b} \right)_m \sin n\, \frac{m\pi}{r} \right] \sin n\overline{\theta}_1 \right\} . \tag{99}$$

It will be readily recalled that

$$\sin \alpha \sin \beta = \tfrac{1}{2} [\cos (\alpha-\beta) - \cos (\alpha+\beta)] \quad ;$$

with this identity (99) is

$$\overline{\alpha}_i(\overline{y}_1) = \frac{180}{4\pi r \sin \overline{\theta}_1} \left\{ \sum_{m=1}^{r-1} \left(\frac{C_\ell c}{b} \right)_m \sum_{n=1}^{r-1} n \left[\cos n \left(\overline{\theta}_1 - \frac{m\pi}{r} \right) - \cos n \left(\overline{\theta}_1 + \frac{m\pi}{r} \right) \right] \right\} \tag{100}$$

where we have chosen to terminate the infinite series in (99) at r - 1 terms for computational convenience.

We choose to evaluate the induced angle of attack at the same points along the span at which we are required to find C_ℓ. To accomplish this we put $\overline{\theta}_1 = k\pi/r$. if we define

$$\beta_{mk} = \frac{180}{4\pi r \, \sin \frac{k\pi}{r}} \sum_{n=1}^{r-1} n \left[\cos n \frac{(k-m)\pi}{r} - \cos n \frac{(k+m)\pi}{r} \right] \qquad (101)$$

then

$$\bar{\alpha}_i(\bar{y}_1) = \bar{\alpha}_{ik} = \sum_{m=1}^{r-1} \left(\frac{c_{\ell}c}{b} \right)_m \beta_{mk} \qquad (102)$$

Now if $k = m$

$$\beta_{mk} = \frac{180}{4\pi r \, \sin \frac{k\pi}{r}} \sum_{n=1}^{r-1} n \left[1 - \cos \frac{2kn\pi}{r} \right] \qquad (103)$$

The sum of the series is not easily found but can be shown by evalauting series of varying numbers of terms to be $r^2/2$; hence

$$\beta_{mk} = \frac{180r}{8\pi \, \sin \frac{k\pi}{r}} \qquad (104)$$

for

$$k = m \qquad .$$

If $k + m$ is even (and $k \neq m$) then β_{mk} contains terms such as

$$1 \left[\cos 1 \left(\frac{1-3}{4} \right) \pi - \cos 1 \left(\frac{1+3}{4} \right) \pi \right]$$
$$+ 2 \left[\cos 2 \left(\frac{-2}{4} \right) \pi - \cos 2\pi \right]$$
$$+ 3 \left[\cos 3 \left(\frac{-2}{4} \right) \pi - \cos 3\pi \right] = 0$$

which always sum to zero. In a similar fashion one can show that when $k + m$ is odd

$$\beta_{mk} = \frac{180}{4\pi r \, \sin \frac{k\pi}{r}} \left[\frac{1}{1 - \cos \frac{(k+m)\pi}{r}} - \frac{1}{1 - \cos \frac{(k-m)\pi}{r}} \right] \qquad (105)$$

Note that the value of β_{mk} depends only on the number of spanwise stations used and is independent of wing aspect ratio or taper ratio.

The perceptive reader will note that (102) really does not supply us with more information regarding the spanwise variation of $\bar{\alpha}_i(\bar{y}_1)$ than we had previously. It does, however, provide us with the physical and geometric

100

tools we need to work out a successful iterative procedure for finding the correct value of $C_\ell c/\overline{b}$ as a function of span. That this is true will become apparent as we proceed. First, let us write (81) as

$$\alpha_e(y) = \alpha_k - \frac{R}{} \frac{d\overline{u}}{du} \sum_{m=1}^{r-1} \left(\frac{C_\ell c}{\overline{b}} \right)_m \beta_{mk} \; . \tag{106}$$

As a first approximation we will assume that the variation of lift coefficient with angle of attack, C_{ℓ_α}, at any spanwise location, k, is constant. Then by writing

$$\alpha_e \left(C_{\ell_\alpha} \frac{c}{\overline{b}} \right)_k = \left(\frac{C_\ell c}{\overline{b}} \right)_k + \Delta_k \tag{107}$$

we can define Δ_k as the amount that must be added to the initial estimate for $(C_\ell c/\overline{b})$ in order to obtain a new value which includes in it effects from other portions of the wing. A second iteration is formed as follows:

$$\left\{ \alpha_k - \frac{R}{} \frac{d\overline{u}}{du} \sum_{m=1}^{r-1} \left[\left(\frac{C_\ell c}{\overline{b}} \right) + \Delta' \right]_m \beta_{mk} \right\} \left(C_{\ell_\alpha} \frac{c}{\overline{b}} \right)_k = \left(\frac{C_\ell c}{\overline{b}} \right)_k + \Delta'_k \; . \tag{108}$$

Subtraction of (108) from (107) yields

$$\left(C_{\ell_\alpha} \frac{c}{b} \right)_k \left(\frac{R}{} \frac{d\overline{u}}{du} \right)_k \sum_{m=1}^{r-1} \Delta'_m \beta_{mk} = \Delta_k - \Delta'_k \; . \tag{109}$$

If we define

$$\left(\frac{C_{\ell_\alpha} c}{\overline{b}} \right)_K \left(\frac{R}{} \frac{d\overline{u}}{du} \right)_k \frac{180r}{8\pi \sin \frac{k\pi}{r}} = G_{kk} \; , \tag{110}$$

then Equation (109) could be written as

$$\Delta'_k + G_{kk} \Delta'_k + \left[\sum_{m=1}^{r-1} \Delta'_m \beta_{mk} \right]_{k \neq m} \left(\frac{C_{\ell_\alpha} c}{\overline{b}} \right)_k \left(\frac{R}{} \frac{d\overline{u}}{du} \right)_k = \Delta_k \; . \tag{111}$$

Dividing by G_{kk} yields

$$\left(1 + \frac{1}{G_{kk}} \right) \Delta'_k + \left[\sum_{m=1}^{r-1} \frac{\beta_{mk}}{\beta_{kk}} \Delta'_k \right]_{k \neq m} = \frac{\Delta_k}{G_{kk}} \; . \tag{112}$$

Equation (112) represents r/2 simultaneous equations which may be represented in matrix form as

101

$$\left[G_{ij} \right] \left\{ \Delta_i' \right\} = \frac{1}{G_{kk}} \left\{ \Delta_i \right\} \tag{113}$$

where $\left[G_{ij} \right]$ is a matrix with all the principal diagonal elements equal to $(1 + 1/G_{kk})$ and the other elements are β_{mk}/β_{kk}. The values to be added to one set of approximate values to obtain a better approximation are therefore given by

$$\left\{ \Delta_i' \right\} = \left[G_{ij} \right]^{-1} \frac{1}{G_{kk}} \left\{ \Delta_i \right\} \quad . \tag{114}$$

As a first approximation to the distribution of lift on the wing we use the expression (Ref. 36)

$$\frac{C_{\ell} c}{b} = C_{\ell} \left(\frac{AR}{AR + 1.8} \right) \left(\frac{c}{c_R} \right) \left(\frac{c_R}{b} \right) \left(\frac{b}{b} \right) \left[\frac{1}{2} + (1 + \lambda) \sqrt{1 - \left(\frac{2\overline{y}}{b} \right)^2} \right] \tag{115}$$

which, as can be seen, contains a simple aspect ratio correction and a simple taper ratio correction to a typical elliptical lift distribution. Here, AR is the aspect ratio, (b^2/wing area), c_R is the root chord, and λ is the taper ratio or the ratio of the tip chord to the root chord. The value of C_{ℓ} on the right-hand side of this equation comes from the two-dimensional data corresponding to the local geometric angle of attack.

Now the flow in the tip regions and its effect on the overall wing characteristics is particularly difficult to determine quantitatively. The more inboard sections of a finite wing are influenced by the downwash generated by the horseshoe vortex system and the upwash due to the fuselage so that the primary effect there is a change in effective angle of attack. In the tip region, on the other hand, there is a substantial spanwise flow which detracts from the flow moving chordwise; consequently, the tip region is able to generate less lift than one would normally expect for a given free stream velocity. This of course reduces the total lift of the wing somewhat. To accommodate this loss in lift within the idea of using two-dimensional data at an appropriate angle of attack, we modify α_e as given by (81) to read (Ref. 36)

$$\alpha_e' = \frac{\left[(\alpha_e - \alpha_{Z.L.})(1 - \sqrt{1 + 4/AR^2}) \right]}{\sqrt{1 + \dfrac{4}{AR^2}}} \quad . \tag{116}$$

We then use α_e' to look up the 3-dimensional value of the section lift coefficient from the 2-dimensional data.

If $C_{\ell} c/\overline{b}$ computed in this fashion is not sufficiently close to the initial estimate of $C_{\ell} c/\overline{b}$ than a correction given by (114) is added. The process is repeated until satisfactory agreement is obtained.

102

When a satisfactory lift distribution has finally been obtained, one can employ the same section data to find the profile drag and moment coefficient at each station along the wing. The local induced drag is simply a product of the lift and the induced angle of attack at that point. The overall force and moment coefficients are obtained by integrating the local values over the span. If one uses Simpson's rule for this integration, explicit relations for the lift, drag, and pitching moment can be obtained:

$$C_L = \left(\frac{\overline{b}}{b}\right)^2 \text{ AR} \sum_{m=1}^{r-1} \left(\frac{C_\ell c}{\overline{b}}\right)_m \left[\frac{\pi}{6r}\left(3 - (-1)^m\right) \sin\frac{m\pi}{r}\right] \tag{117}$$

$$C_{D_i} = \frac{\pi}{180}\left(\frac{\overline{b}}{b}\right)^2 \text{ AR} \sum_{m=1}^{r-1} \left[\left(\frac{C_\ell c}{\overline{b}}\right)\alpha_i\right]_m \left[\frac{\pi}{6r}\left(3 - (-1)^m\right) \sin\frac{m\pi}{r}\right] \tag{118}$$

$$C_{D_o} = \left(\frac{\overline{b}}{b}\right)^2 \text{ AR} \sum_{m=1}^{r-1} \left(\frac{C_{d_o} c}{\overline{b}}\right)_m \left[\frac{\pi}{6r}\left(3 - (-1)^m\right) \sin\frac{m\pi}{r}\right] \tag{119}$$

$$C_m = \frac{\text{AR}\,\overline{b}}{b^2 c'} \sum_{m=1}^{r-1} \left(C_m c^2\right) \left[\frac{\pi}{6r}\left(3 - (-1)^m\right) \sin\frac{m\pi}{r}\right] \tag{120}$$

where

$$C_m = C_{m_{c/4}} - \frac{x}{c}\left[C_\ell \cos(\alpha_B - \alpha_i) + C_{d_o} \sin(\alpha_B - \alpha_i)\right]$$

$$- \frac{z}{c}\left[C_\ell \sin(\alpha_B - \alpha_i) - C_{d_o} \sin(\alpha_B - \alpha_i)\right] \tag{121}$$

and c' is the mean aerodynamic chord. Note that this integration is somewhat analogous to the process represented by

$$\frac{1}{bc'} \int_{-b/2}^{b/2} C_\ell c \, dy \ .$$

Since the computation of $(C_\ell c/\overline{b})$ has really been carried out in the $\overline{u}$-plane so far as the wing span is concerned we must multiply our result by $\overline{b}/b$ to transform it to the physical plane. AR is just b/c' and the average (aerodynamic not geometric) value of c over the span is c'. The term in brackets is the multiplier employed by Simpson's Rule. It is seen therefore that C_L is in fact an average over the span in the physical plane.

103

CONCLUDING REMARKS

The totality of procedures described in this section for computing the aerodynamic characteristics of complete wings can be seen to be capable—at least in principle—of a very accurate representation of physical reality. The vortex filament distribution approach to determining the inviscid pressure distribution on airfoil sections leads in the limit of a multitude of filaments and high computational accuracy to very good results. The application of the lifting line concept to the correction of section characteristics for finite span effects also can yield very reliable results, provided the number of spanwise stations at which the characteristics are computed is large, computational accuracy is maintained, and the underlying assumptions of the lifting-line theory are not violated by the planform it is supposed to represent.

The approach used here has two principal advantages, insofar as the computation of light aircraft wing aerodynamic characteristics is concerned, when compared with the finite element or vortex lattice approach which has been discussed prominently in the literature of late:

(1) three-dimensional drag data is obtained,

(2) the computational time for an equally accurate potential solution is far less since, in effect, the spanwise integration of vorticity is replaced by a single integration along the lifting line. This, of course, is made possible by restricting the interest to unswept, moderate-to-high aspect ratio wings in subsonic flow.

The computation of the effects of fluid viscosity is the least rigorous and least accurate of the procedures. So long as the boundary layer is laminar and the wing surface smooth one could expect to obtain very acceptable drag and boundary layer displacement effects by successive refinement of the psuedo-airfoil shape using a more general polynomial for u/u_e if necessary. Laminar boundary layer methods inherently more accurate than the momentum integral technique are also available as computer programs and could be included if desired, at the cost, however, of greatly increased computer time. The local characteristics during boundary layer transition and for turbulent boundary layers are here computed by what may be termed state of the art techniques—not completely rigorous, but about as reliable as any available. The procedure is therefore expected to be less accurate as the need for a precise knowledge of boundary layer behavior increases—at high angles of attack or in the presence of surface irregularities.

Finally, we should mention that to translate these concepts into useful design tools we must also employ sound, efficient computational procedures. Although a treatment of the rationale behind the choice of one numerical method over another for a particular computation is beyond the scope of the present discussion, it must be emphasized that such decisions can be crucial both economically and technically. For this reason it is to be expected that the major improvements which occur in the procedure presented in this work, at least in the immediate future, will likely be in the area of computational effectiveness.

104

A THEORY FOR THE PREDICTION OF LIFT, DRAG, AND PITCHING MOMENT OF LIGHT AIRCRAFT FUSELAGES

INVISCID FLOW OVER FUSELAGES

In the preceding discussion we were concerned with a means of predicting the lift, drag, and moment characterisitcs of wings. An airplane, however, contains other parts which contribute to the overall aerodynamic force and moment experienced by the vehicle. Most notable of these is the fuselage. The fuselage contributes to the overall drag for two reasons: it has a large surface area against which the air can "rub" and in order to provide a finite volume in which to carry a useful payload it must move air "out of the way." It can also contribute some amount to the lift in addition to that provided by the bridging action between the two wings. This effect is usually significant only at high angles of attack. If the fuselage produces contributions to the lift and drag, then it is also likely to affect the pitching moment.

Because the fuselage is usually not a body of revolution but rather a three-dimensional body with a plane of symmetry its proper analytical representation is more difficult than that of an airfoil. In the case of an airfoil we were able to represent the surface by a series of line vortices. Because a vortex must either be closed or extend to infinity, the only vortex we can use to represent three-dimensional bodies is a ring. To be sure the ring can be contorted, but it must close. The geometric problems associated with describing such a ring leads us to ask whether there is another flow function for describing the inviscid flow about fuselages which may be easier to use. The point source is an elementary flow which can be described quite easily mathematically. The potential at the point (x, y, z) due to a source at (ξ, η, ζ) is given by

$$\phi = \frac{-q}{4\pi} \frac{1}{[(x - \xi)^2 + (y - \eta)^2 + (z - \zeta)^2]^{1/2}} \qquad (122)$$

where q is called the source strength. The components of the velocity associated with a source flow are then

$$u = \frac{q}{4\pi} \frac{(x - \xi)}{[(x - \xi)^2 + (y - \eta)^2 + (z - \zeta)^2]^{3/2}} \qquad (123)$$

$$v = \frac{q}{4\pi} \frac{(y - \eta)}{[(x - \xi)^2 + (y - \eta)^2 + (z - \zeta)^2]^{3/2}} \qquad (124)$$

$$w = \frac{q}{4\pi} \frac{(z - \zeta)}{[(x - \xi)^2 + (y - \eta)^2 + (z - \zeta)^2]^{3/2}} . \qquad (125)$$

105

That this flow is irrotational and satisfies the Laplace Equation is easily verified.

Physically, the three-dimensional source is a point from which fluid emanates in straight lines in all directions. The quantity of fluid emanating from the point depends upon the value of q. By placing a source in a uniform stream one in effect displaces the streamlines in the uniform stream because of local additions to the mass flow. By distributing sources in space and using a different strength for each source it is possible to cause the streamline pattern of the flow to simulate the shape of the windward side of complex bodies quite satisfactorily. To represent a closed body it is necessary to withdraw the mass added by the sources so that the streamlines can return to their original positions downstream of the body. We do this by means of a distribution of sources with negative values of the source strength, or sinks. Hence, the net source strength when integrated over a closed body is zero.

A source flow is also a flow which inherently cannot create a net circulation about a body. Consequently, it cannot be used to represent a situation where lift is present. Such situations can be simulated by a combination of sources, horseshoe vortices, and a uniform stream. Even though a distribution of sources in a uniform stream by itself cannot represent a lifting body, it is suitable for determining the inviscid pressure distribution on a non-lifting body. Then by adding a boundary layer one can determine both the form and friction drag of such bodies.

The technique by which such a calculation is carried out is very similar to that which we employed to determine the aerodynamic characteristics of airfoils. We begin by considering the body surface to be made up of a number of connected plane quadrilaterals. As the number of quadrilaterals approaches infinity, the polygon becomes the body identically. On each plane quadrilateral we place a source of undetermined strength. We require that the flow induced on any one quadrilateral through the interaction of all the sources with the uniform stream be parallel to the quadrilateral surface or that the flow velocity normal to the quadrilateral be zero. This gives us sufficient boundary conditions to write a determinant system of equations, one for each quadrilateral, which we can then solve to find the individual source strengths.

The total velocity induced at a point on the body surface by a series of sources distributed over the body can, following (19), be written

$$\bar{u} = \frac{1}{4\pi} \sum_{N=1}^{k} \frac{q_N(x - \xi_N)}{[(x - \xi_N)^2 + (y - \eta_N)^2 + (z - \zeta_N)^2]^{3/2}}$$

$$\bar{v} = \frac{1}{4\pi} \sum_{N=1}^{k} \frac{q_N(y - \eta_N)}{[(x - \xi_N)^2 + (y - \eta_N)^2 + (z - \zeta_N)^2]^{3/2}} \tag{126}$$

$$\bar{w} = \frac{1}{4\pi} \sum_{N=1}^{k} \frac{q_N(z - \zeta_N)}{[(x - \xi_N)^2 + (y - \eta_N)^2 + (z - \zeta_N)^2]^{3/2}} .$$

To complete the formulation of the equations, we must include the effects of the free stream velocity at (x, y, z) and define the normal to the surface at (x, y, z) so as to be able to state the requirement that the net flow be parallel to the surface. The free stream velocity just adds a term V_∞ to the u-component.

The surface in which the point (x, y, z) lies can, in general, be defined by the equation $a'x + b'y + c'z + d' = 0$. Each quadrilateral composing the fuselage will have a different range of values for x, y, and z as well as for a', b', c', and d'. The values of the coefficients a', b', c', and d' for a particular quadrilateral are found by solving for them from the system of equations

$$a'x_1 + b'y_1 + c'z_1 + d' = 0$$

$$a'x_2 + b'y_2 + c'z_2 + d' = 0$$

$$a'x_3 + b'y_3 + c'z_3 + d' = 0 \qquad\qquad (127)$$

$$a'x_4 + b'y_4 + c'z_4 + d' = 0$$

where (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3) and (x_4, y_4, z_4) are four points all of which lie in the same plane. One can use for these points the four corners of the quadrilateral. One can also rewrite the equation for a plane surface in the form

$$\frac{a'}{\sqrt{a'^2+b'^2+c'^2}}x + \frac{b'}{\sqrt{a'^2+b'^2+c'^2}}y + \frac{c'}{\sqrt{a'^2+b'^2+c'^2}}z + \frac{d'}{\sqrt{a'^2+b'^2+c'^2}} = 0 .$$

Then the quantities

$$a = \frac{a}{\sqrt{a'^2 + b'^2 + c'^2}} , \quad b = \frac{b'}{\sqrt{a'^2 + b'^2 + c'^2}} , \quad c = \frac{c'}{\sqrt{a'^2 + b'^2 + c'^2}}$$

are what are called the direction cosines of the normal to the plane $ax + by + cz + d = 0$. The reason for this may be seen from the sketch below. If we draw a line from point P normal to the plane in which it lies then $a = \cos \gamma_1$, $b = \cos \gamma_2$, and $c = \cos \gamma_3$. a', b', and c' are called direction numbers. The plane through point $P(x_p, y_p, z_p)$ whose normal has direction numbers a', b', and c' is the graph of the equation $a'(x - x_p) + b'(y - y_p) + c'(z - z_p) = 0$. From this discussion we see that direction numbers and direction cosines are proportional.
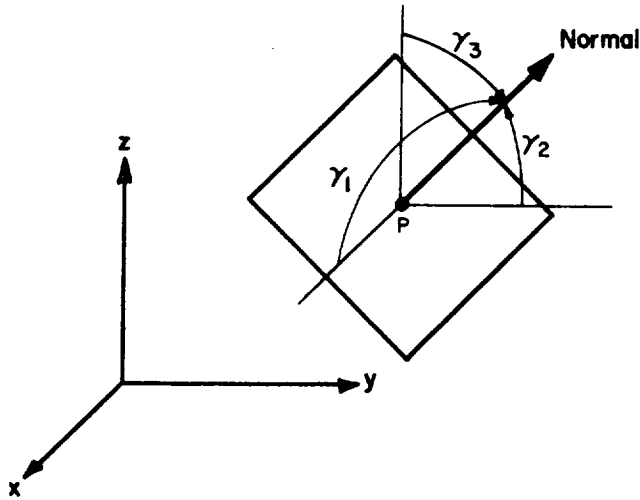
The requirements that there be no flow normal to the surface is satisfied when

$$(\bar{u} + V_\infty)a + \bar{v}b + \bar{w}c = 0 \qquad (128)$$

at one point on each quadrilateral. That this is true is easily seen if one writes the velocity as a vector, $(\bar{u} + V_\infty)\vec{i} + \bar{v}\vec{j} + \bar{w}\vec{k} = \vec{V}$, and the normal as a vector, $a\vec{i} + b\vec{j} + c\vec{k} = \vec{N}$. The scalar or dot product of two vectors is the product of their magnitudes times the cosine of the angle between them. Thus if $\vec{V}\cdot\vec{N} = 0$ the two vectors are at right angles to one another. But $\vec{V}\cdot\vec{N} = (\bar{u} + V_\infty)a + \bar{v}b + \bar{w}c$. When this is zero the flow is parallel to the surface.

It will be recalled that in the two-dimensional case of the airfoil one could as easily chose the condition that the flow must be parallel to the surface as the condition that the flow velocity normal to that surface must be zero. They are equivalent statements and offer similar mathematical problems.* In the three-dimensional case, however, one does not know *a priori* the direction of the flow parallel to the surface. For this reason, a single, fixed-value boundary condition becomes difficult to specify. Requiring that the velocity normal to the surface be zero on the other hand, removes this difficulty.

---

* It has been pointed out by Chen (Ref. 27) that specifying that the tangent of the flow angle be the same as the local angle of the surface makes the solution less sensitive to errors in the coordinates of the surface than requiring that the flow velocity normal to the surface be zero.

While perhaps not evident in the discussion above there is a problem in relating points on the body surface to the coefficients of (127). Simply stated it is that the four points on the body surface closest to x, y, z do not necessarily lie in the same plane. In fact, it would be most remarkable if they did. Further, if the four corner points on any quadrilateral are "adjusted" so that they are in fact coplanar then the edges of that quadrilateral will not necessarily be colinear with the edges of adjacent quadrilaterals. The effect then is to represent the body by an ensemble of planar "scales," similar to those of a fish, rather than by something that resembles a wire gridwork. In the limit, however, as the number of quadrilaterals becomes very large, the two models become the same thing.

How then to resolve the dilema? Forming triangular panels from three adjacent surface points insures that the three points are coplanar but, as one may readily observe, it also doubles the number of panels one needs to cover a given area. Since the computational time required to solve for the source strength on each panel varies approximately as the number of panels squared, one does not take such a step lightly. On the other hand if one is willing to "adjust" the corner points of a quadrilateral to make them coplanar, how should he proceed to insure that the calculated normal is nearly the same as the true normal to the body at x, y, z. One way which has been suggested (Ref. 23) approaches the problem by writing, first of all, equations for the two diagonals of the quadrilateral. This is straightforward since the two opposite corner points of a quadrilateral define a straight line uniquely. Now, if one is willing to say that these two lines are essentially coplanar, then the crossproduct of their vector representations is the normal to the quadrilateral. For example, the equation describing a line through points $P_1(x_1, y_1, z_1)$ and $P_2(x_2, y_2, z_2)$

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1} \tag{129}$$

while the equations describing a line through points $P_3(x_3, y_3, z_3)$ and $P_4(x_4, y_4, z_4)$ are

$$\frac{x - x_3}{x_4 - x_3} = \frac{y - y_3}{y_4 - y_3} = \frac{z - z_3}{z_4 - z_3} \quad . \tag{130}$$

The various points and lines are depicted on the sketch below.

109
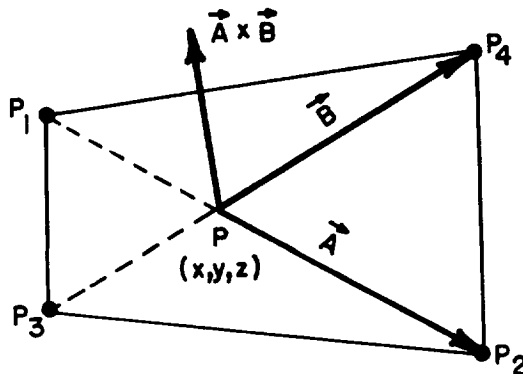
Now, the point at which the two lines have the same values of x and y (that is, the point at which they cross one another) is found by solving the two equations simultaneously to obtain

$$y = \frac{(x_1 - x_3) - y_1(x_2 - x_1)/(y_2 - y_1) + y_3(x_4 - x_3)/(y_4 - y_3)}{\dfrac{x_4 - x_3}{y_4 - y_3} - \dfrac{x_2 - x_1}{y_2 - y_1}}$$

$$x = x_1 - y_1 \left(\frac{x_2 - x_1}{y_2 - y_1} \right)$$

$$+ \left(\frac{x_2 - x_1}{y_2 - y_1} \right) \left[\frac{(x_1 - x_3) - y_1(x_2 - x_1)/(y_2 - y_1) + y_3(x_4 - x_3)/(y_4 - y_3)}{\dfrac{x_4 - x_3}{y_4 - y_3} - \dfrac{x_2 - x_1}{y_2 - y_1}} \right] .$$

(131)

z is then found from

$$z = (y - y_3) \left(\frac{z_4 - z_3}{y_4 - y_3} \right)$$

$$\text{or} \qquad z = (y - y_1) \left(\frac{z_2 - z_1}{y_2 - y_1} \right) .$$

(132)

A vector beginning at $P(x,y,z)$ and terminating at $P_2(x_2, y_2, z_2)$ is represented by

$$\vec{A} = (x_2 - x)\vec{i} + (y_2 - y)\vec{j} + (z_2 - z)\vec{k}$$

(133)

while that beginning at $P(x,y,z)$ and terminating at $P_4(x_4, y_4, z_4)$ is represented by

$$\vec{B} = (x_4 - x)\vec{i} + (y_4 - y)\vec{j} + (z_4 - z)\vec{k} .$$

(134)

The vector which is normal to the plane defined by $\vec{A}$ and $\vec{B}$ is then

$$\vec{A} \times \vec{B} = (x_2 - x)(y_4 - y)\vec{k} - (x_2 - x)(z_4 - z)\vec{j}$$

$$+ (y_2 - y)(z_4 - z)\vec{i} - (y_2 - y)(x_4 - x)\vec{k}$$

$$+ (z_2 - z)(x_4 - x)\vec{j} - (z_2 - z)(y_4 - y)\vec{i}$$

(135)

$$\vec{A} \times \vec{B} = [(y_2 - y)(z_4 - z) - (z_2 - z)(y_4 - y)]\vec{i}$$

$$+ [(z_2 - z)(x_4 - x) - (x_2 - x)(z_4 - z)]\vec{j}$$

$$+ [(x_2 - x)(y_4 - y) - (y_2 - y)(x_4 - x)]\vec{k}$$

110

The condition that the flow be parallel to the surface is then satisfied when

$$\vec{V} \cdot (\vec{A} \times \vec{B}) = 0$$

or $(\bar{u} + V_\infty)\left[(y_2 - y)(z_4 - z) - (z_2 - z)(y_4 - y)\right] + \bar{v}\left[(z_2 - z)(x_4 - x)\right.$

$$\left. - (x_2 - x)(z_4 - z)\right] + \bar{w}\left[(x_2 - x)(y_4 - y) - (y_2 - y)(x_4 - x)\right] = 0.$$

(136)

For simplicity we will define the terms in brackets as $\bar{a}$, $\bar{b}$, and $\bar{c}$ respectively. Note that one must exercise care in choosing the labels for the points so that the normal $(\vec{A} \times \vec{B})$ is always outward.

We have now found a means to represent the boundary condition of no flow across the surface of the body in terms of the coordinates of the corner points of the quadrilaterals making up the body.
The system of equations one must then solve to find the individual source strengths is

$$\left(V_\infty + \frac{1}{4\pi} \sum_{N=1}^{k} \frac{q_N(x_1 - \xi_N)}{[(x_1 - \xi_N)^2 + (y_1 - \eta_N)^2 + (z_1 - \zeta_N)^2]^{3/2}}\right) \bar{a}_1$$

$$+ \frac{\bar{b}_1}{4\pi} \sum_{N=1}^{k} \frac{q_N(y_1 - \eta_N)}{[(x_1 - \xi_N)^2 + (y_1 - \eta_N)^2 + (z_1 - \zeta_N)^2]^{3/2}}$$

$$+ \frac{\bar{c}_1}{4\pi} \sum_{N=1}^{k} \frac{q_N(z_1 - \zeta_N)}{[(x_1 - \xi_N)^2 + (y_1 - \eta_N)^2 + (z_1 - \zeta_N)^2]^{3/2}} = 0$$

$$\vdots$$

(137)

$$\left(V_\infty + \frac{1}{4\pi} \sum_{N=1}^{k} \frac{q_N(x_k - \xi_N)}{[(x_k - \xi_N)^2 + (y_k - \eta_N)^2 + (z_k - \zeta_N)^2]^{3/2}}\right) \bar{a}_k$$

$$+ \frac{\bar{b}_k}{4\pi} \sum_{N=1}^{k} \frac{q_N(y_k - \eta_N)}{[(x_k - \xi_N)^2 + (y_k - \eta_N)^2 + (z_k - \zeta_N)^2]^{3/2}}$$

$$+ \frac{\bar{c}_k}{4\pi} \sum_{N=1}^{k} \frac{q_N(z_k - \zeta_N)}{[(x_k - \xi_N)^2 + (y_k - \eta_N)^2 + (z_k - \zeta_N)^2]^{3/2}} = 0 \quad .$$

111

In these equations k = the number of quadrilaterals by which the surface is represented. x_k, y_k, z_k is the point on each quadrilateral at which the boundary condition is satisfied and ξ_k, η_k, ζ_k is the point on each quadrilateral at which the source is located. Not making these points coincident will usually lead to less waviness in the surface pressures and velocities. If we choose to separate the source point from the point at which the boundary condition is satisfied, *i.e.*, the point at which the normal is computed, then we need to write an equation for a <u>plane</u> quadrilateral having its normal identical with that given by (135) and its corner points located at the same x and y locations but with slightly different z values. We need to do this in order to locate the source approximately on the surface in a known relationship to the corner points and to P(x,y,z). From Equation (127) we see that since $\bar{a}$, $\bar{b}$, and $\bar{c}$ are now the direction numbers of the normal to the plane, picking one value of z, say z_1 as lying on the physical surface permits one to solve for the other value of z and for d. If we choose to locate the source point at 1/3 the quadrilateral chord then the x-coordinate of the source point is given by

$$\tfrac{1}{2}\left[x_1 + \frac{x_4 - x_1}{3} + x_3 + \frac{x_2 - x_3}{3} \right] = \xi_k \,.$$ (138)

The y-coordinate is given by

$$\tfrac{1}{2}\left[y_3 + \frac{y_2 - y_3}{3} + y_1 + \frac{y_4 - y_1}{3} \right] = \eta_k \,.$$ (139)

The z-coordinate is then determined from the equation of that particular quadrilateral. If one wishes to move the point at which the boundary condition is satisfied to some other, fixed point along the quadrilateral chord, he could follow a similar procedure.

The source strengths, q_N, are then found by solving the rather large system of Equations (137) simultaneously. Given these values one can then find local velocity components from (126). The local pressure on the fuselage is then given by

$$P = P_\infty + \tfrac{1}{2}\rho\left[V_\infty^2 - (V_\infty + \bar{u})^2 - \bar{v}^2 - \bar{w}^2 \right] \,.$$ (140)

Obviously, to obtain a reasonably accurate picture of the surface pressures, the number of quadrilaterals with which the surface is represented must be very large, particularly if the fuselage shape deviates significantly from that of a streamlined body. As was the case in the analysis of two-dimensional bodies (airfoils), conventional practice usually represents the potential by a constant source strength over the entire surface of a particular panel rather than by a point source as is done here. Use of the continuous distribution, it will be recalled, usually results in less waviness in the computed surface streamline. This streamline of course is required by our formulation to be parallel with the physical surface only at one point on each panel.

Because of the very large number of panels usually required to achieve reasonably accurate descriptions of the surface pressures on fuselage-like bodies, the opportunity to save substantial amounts of computer time or storage through the use of sophisticated numerical or mathematical procedures exists. Every program for performing such computations that the present authors have examined in detail (three at this writing) either carries out the calculation in pieces, takes advantage of the special character of the coefficient matrix in performing the inversion, recognizes that panels far away from the one on which the boundary condition is satisfied contribute little to the flow and therefore can be represented more approximately with no loss in accuracy, or uses line sources to represent all or part of the fuselage. Line sources, of course, can be used to greatly simplify the formulation of the problem, but unfortunately they can describe only slender fuselages adequately. A detailed discussion of the advantages and disadvantages of the various computation techniques is beyond the scope of the present work. This activity is mentioned merely to indicate to the reader that reducing the cost and complexity of the computations has already received considerable intelligent attention and is likely to be the subject of further intensive study in the near future.

By treating the fuselage as an isolated body, we have assumed implicitly that the fuselage is flying at such an inclination to the stream that it generates no lift other than by providing a bridge between the two halves of the wing. We shall also assume that the effect of the entire fuselage on the wing lift is adequately treated by the method of the previous section. Further, we shall assume that for purposes of finding the fuselage drag we can treat the fuselage as a free body, adding the wing-body interference effects later as an empirical or semi-empirical correction. These assumptions leave us with the necessity of determining only the boundary layer displacement and skin friction effects. In subsequent sections we shall examine some of the rationale for these assumptions and determine in a general way the conditions for which they are reasonable.

VISCOUS FLOW OVER FUSELAGES

The problem of determining the characteristics of a boundary layer flowing over a general, three-dimensional body is one of great fundamental interest. Unfortunately, the only known solutions are for axisymmetric bodies or other special cases. It is known, however, that if the body cross-sectional area and volume do not change rapidly in the streamwise direction and if there is no significant pressure gradient in the cross flow direction, then the boundary layer behaves in much the same way as it would on a two-dimensional body subject to the same history of pressure gradients in the streamwise direction. If we are willing to accept the restrictions and errors inherent in assuming that the three-dimensional boundary layer can be treated through such a concept, then we may use the techniques of the previous section to determine δ^* and $\tau_{y=0}$ along the

body streamlines[†]. One can obtain a reasonable estimate of the circumferential variation of δ^* and $\tau_{y=0}$ at any axial station by calculating δ^* and $\tau_{y=0}$ along a series of these lines approximately equally spaced in the circumferential direction.

A new pseudo-surface can then be constructed by smoothing these displacements from the original surface and a new inviscid pressure distribution can then be calculated. This process for determining the fuselage drag follows essentially the same path as that employed earlier for determining airfoil drag except that the pressures and viscous shear must be integrated over appropriate <u>areas</u> rather than line segments. We will examine the process in somewhat more <u>detail</u> a little later.

One may argue with considerable justification that if the use of a two-dimensional-boundary-layer-along-streamlines procedure gives results which compare well with experimental values for a group of fuselages, then it should be regarded as satisfactory for engineering purposes on this basis alone. However, since it is not exceedingly difficult to determine—to a fair degree of approximation—the conditions under which the additional terms in the three-dimensional boundary layer equations are small compared with the two-dimensional-along-streamlines terms, we shall now make this determination in order to identify those cases where the results obtained by the procedure may be suspect.

We begin by choosing to represent the fuselage locally by a section of a prolate spheroid with its major axis aligned with the local streamline and its center always in the x-z plane. The figures below depict this concept.

---

† Formally, two adjacent streamlines form the boundaries of the flow of a given quantity of fluid. The position of these lines on a body can be determined from the magnitude and direction of the flow over the surface.

114

Prolate spheroid obtained by
fitting points A, B, C, D

FUSELAGE

Streamline 1

Streamline 2

Streamline 3

Panel over which surface of spheroid is approximately the
same as surface of fuselage

At some other point on the fuselage the fitted spheroid will
have different major and minor axes and different center
location. Inclination of major axis is controlled by direction
of streamline crossing panel centroid.

NOTE: Major axis of ellipsoid is forced to be parallel to Streamline 2

VIEW SHOWING HOW A SECTION OF A PROLATE SPHEROID
WITH CENTER IN THE REFERENCE X-Z PLANE MAY BE USED
TO REPRESENT THE SURFACE OF AN AIRCRAFT FUSELAGE LOCALLY

Traces of streamlines over body in x-z plane

Trace of prolate spheroid fitting A,B,C,D in x-z plane

Z

Y

X

ORIGINAL XYZ REFERENCE
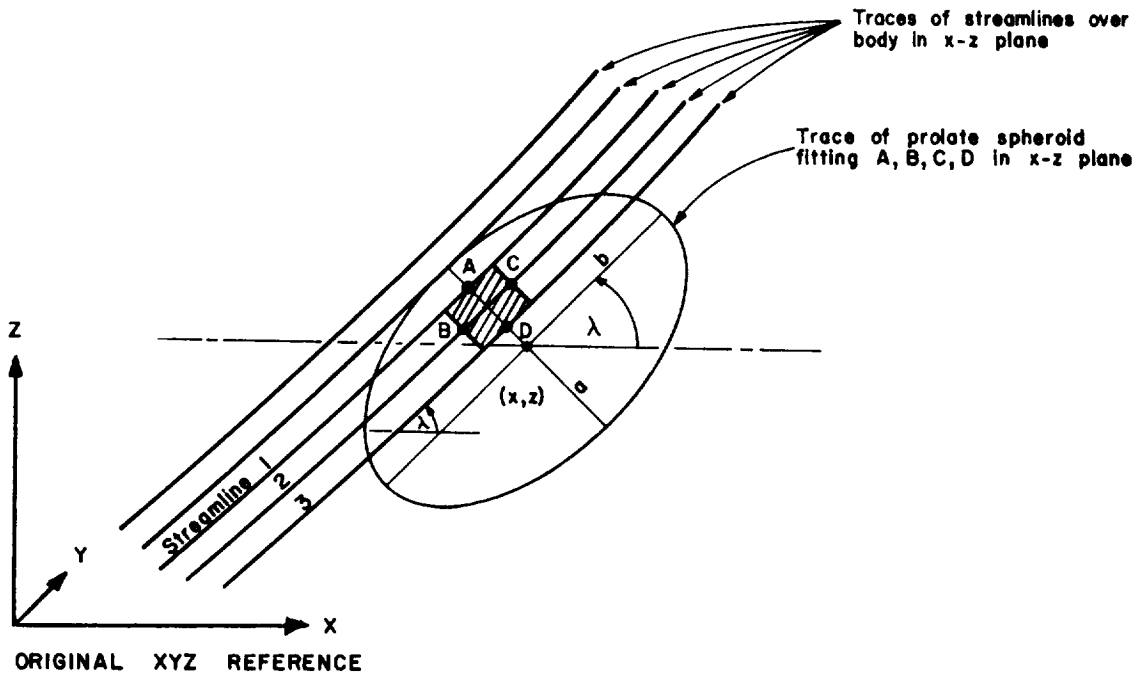
LOCATION OF SPHEROID WITH
RESPECT TO REFERENCE AXES
AND DEFINITION OF SYMBOLS

The solutions of equations (126) give us the magnitude and direction of the flow velocity at one point on each panel. If we observe a group of panels, as in the sketch below,

116
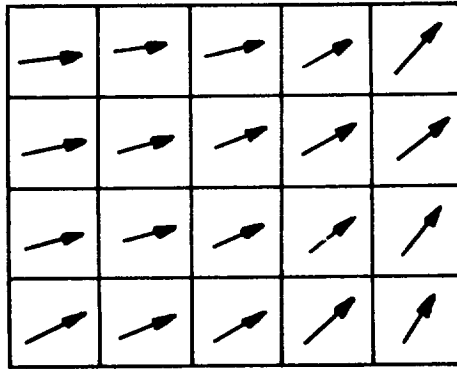
and place, at the point on each where the boundary condition is satisfied, a vector whose direction is the flow direction at that point and whose length is proportional to the flow magnitude, then we may employ the method of isoclines (Ref. 107, page 97) to sketch the streamlines. The streamlines, after all, represent the trajectories of given masses of fluid as they travel over the surface of the body. For computational purposes, however, it is preferable to describe these streamlines in a more analytical fashion. Consider the section of the surface shown in the sketch below:
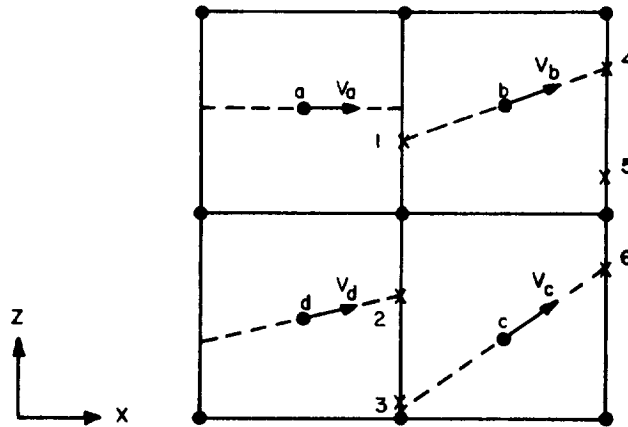
From the magnitudes of the surface velocity components we know the direction of the flow at a, b, c, and d. Let us assume that this direction is constant across the panel (as shown by the dashed line). The quantity of flow that moves between b and c depends upon the distance between b and c and the average velocity between b and c. Since the quantity of fluid between two streamlines is always constant, the appearance of converging streamlines means that the flow velocity is increasing as the flow moves from left to right. If we assume that the fluid velocity at a given x-station between two streamlines can be considered to be constant, with a value between that along line 1-4 and that along line 3-6 at the same x-station, then the position of the streamline continuing to the right from point 2 will be related to points 4 and 6 as $\left(\dfrac{\text{line } 2\text{-}3}{\text{line } 1\text{-}3}\right) = \left(\dfrac{\text{line } 5\text{-}6}{\text{line } 4\text{-}6}\right)$. This insures that the quantity of fluid between 2 and 3 is the same as between 5 and 6. Further the magnitude of the velocity at 5 is that at 2 times $\left(\dfrac{\text{line } 1\text{-}3}{\text{line } 4\text{-}6}\right)$. Since we know the coordinates of the corner points on each planar panel as well as those of points a, b, c, and d in terms of the original reference axes, we may describe the line segment from 2-5 in this reference system without undue difficulty. We can do the same for other similar line segments and so relate the streamlines over the body to one another in an analytical fashion. The equation for any particular streamline will change of course as it crosses a panel boundary.

We recognize that the assumptions we made with regard to the velocity between streamlines and the straightness of streamlines across panels are true only in the limit of vanishing panel size. However, since we will generally employ more than 400 panels to represent a fuselage such assumptions should not introduce significant error in practice.

Having identified the location of the streamlines in this manner we proceed to orient the prolate spheriod by which we represent a section of the surface of the fuselage.

$$\text{Let } \lambda = \tan^{-1}\left[\frac{z_{n+1} - z_{n-1}}{x_{n+1} - x_{n-1}}\right] \tag{141}$$

be the inclination of the streamline and z_{n+1}, x_{n+1} are the z and x coordinates of a point on the streamline just immediately downstream of the point of interest. z_{n-1} and x_{n-1} are then the coordinates of the point immediately upstream of the point of interest. λ defined in this fashion also represents the angle which the major axis of the spheroid makes with the original x-axis, as noted in one of the foregoing sketches.

This method of determining how the major axis should be oriented in order that it lie parallel to a streamline becomes indeterminant when the streamline is near the top or bottom of the fuselage. Indeterminacy occurs because

changes in flow direction are no longer confined principally to the x-z reference plane. For example, crossflow over the top of the fuselage may be evident only as a point in the x-z plane. The condition can be rectified by rotating ones viewpoint so that the changes in flow direction occur almost entirely in a plane normal to the direction of view. However, to illustrate the concept of fitting an arbitrary fuselage-like body locally by a prolate spheroid, we need work only with streamlines lying in a plane parallel to the x-z reference plane.

The equation for a prolate spheroid at the origin reads

$$\left(\frac{x}{b}\right)^2 + \left(\frac{y}{a}\right)^2 + \left(\frac{z}{a}\right)^2 = 1 \tag{142}$$

If, however, the origin is translated and the body rotated, both events taking place only in the x-z plane, the equation becomes

$$Ax^2 + By^2 + Cz^2 + Dxz + Ex + Fz + G = 0 \tag{143}$$

where

$$C = (a^2 \sin^2 \lambda + b^2 \cos^2 \lambda)$$

$$B = b^2$$

$$A = (a^2 \cos^2 \lambda + b^2 \sin^2 \lambda)$$

$$D = 2(b^2 - a^2)\sin \lambda \cos \lambda \tag{144}$$

$$F = 2x_1(b^2 - a^2)\sin \lambda \cos \lambda - 2z_1(b^2 \cos^2 \lambda + a^2 \sin^2 \lambda)$$

$$E = 2z_1(b^2 - a^2)\sin \lambda \cos \lambda - 2x_1(b^2 \sin^2 \lambda + a^2 \cos^2 \lambda)$$

$$G = 2x_1 z_1(b^2 - a^2)\sin \lambda \cos \lambda + (b^2 x_1^2 + a^2 z_1^2)\sin^2 \lambda$$

$$+ (a^2 x_1^2 + b^2 z_1^2)\cos^2 \lambda - a^2 b^2$$

Since we have chosen to specify λ according to (141) and since the body cross-section in the original y-z plane is a circle, the number of unknown constant values is really only four: a, b, x_1, and z_1. We therefore rearrange the equation as follows:

$$a^2(h_1 + h_2 \, x_1 + h_3 \, z_1 + h_4 \, x_1 \, z_1 + h_5 \, x_1^2 + h_6 \, z_1^2)$$

$$+ \, b^2(h_7 + h_8 \, x_1 + h_9 \, z_1 + h_{10} \, x_1 \, z_1 + h_{11} \, x_1^2 + h_{12} \, z_1^2) = a^2 b^2 \qquad (145)$$

In this form

$$h_1 = z^2 \sin^2 \lambda + x^2 \cos^2 \lambda - 2xz \sin \lambda \cos \lambda$$

$$h_2 = - \, 2x \cos^2 \lambda + 2z \sin \lambda \cos \lambda$$

$$h_3 = + \, 2x \sin \lambda \cos \lambda - 2z \sin^2 \lambda$$

$$h_4 = - \, 2\sin \lambda \cos \lambda$$

$$h_5 = \cos^2 \lambda$$

$$h_6 = \sin^2 \lambda \qquad\qquad (146)$$

$$h_7 = z^2 \cos^2 \lambda + y^2 + x^2 \sin^2 \lambda + 2xz \sin \lambda \cos \lambda$$

$$h_8 = - \, 2x \sin^2 \lambda - 2z \cos \lambda \sin \lambda$$

$$h_9 = - \, 2x \sin \lambda \cos \lambda - 2z \cos^2 \lambda$$

$$h_{10} = 2\sin \lambda \cos \lambda$$

$$h_{11} = \sin^2 \lambda$$

$$h_{12} = \cos^2 \lambda$$

This equation we then fit to the fuselage surface at four points (A, B, C, D in sketch) to evaluate a, b, x_1, and z_1. We do this by choosing two points on the streamline of interest, one on either side of the point of interest, and one point on each of the adjacent streamlines as close to the point of interest as possible. We have then a system of four non-linear algebraic equations which we must solve to find a, b, x_1, and z_1. The equations are non-linear because they contain products of the unknowns, e.g., x_1^2, $x_1 z_1$.

One method of solution for such a system of four equations is an adaptation of Newton's method for finding the roots of a single equation. In Newton's method one makes an initial estimate for the value of a root. If the estimate is reasonably close, the procedure yields a second approximation which is closer to the correct value. The procedure is repeated until the desired degree of accuracy is achieved. The generalization to a system of

120

four equations is outlined in Ref. 100, page 105. It is of course a numerical procedure of some length and is therefore best done on a computer.

Having found a and b for the particular prolate spheroid which represents the surface at the point of interest we ask how we might describe in orthogonal coordinates the length of a line along the surface of the spheroid. We wish to do this because the expression for the line element provides us with a means of characterizing the space (prolate spheroidal in this case). We may then use the boundary layer equations written in general curvilinear coordinates with the local values of the space inserted to evaluate the effects of body curvature on the values one might obtain for τ_w and δ^* using a momentum integral formulation.

Consider the figure below:

$$x = b \sin \theta$$
$$y = a \cos \theta$$
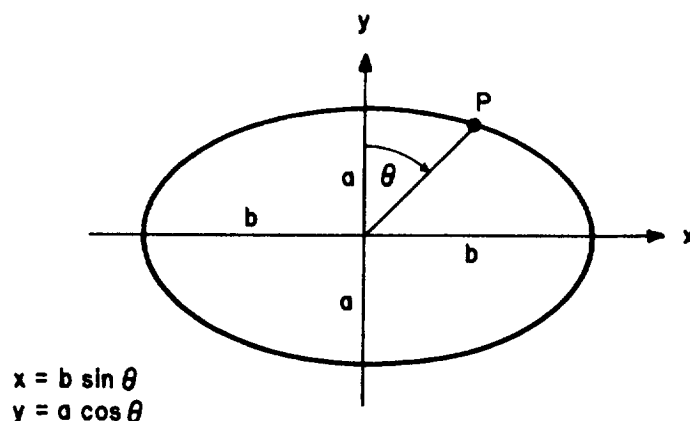
We recognize that the distance along the perimeter is given by

$$ds_1 = \sqrt{a^2 \cos^2 \theta + b^2 \sin^2 \theta}\; d\theta \, , \qquad\qquad (147)$$

while if we rotate the figure about its x-axis the circumferential displacement of P is given by

$$ds_2 = a \cos \theta \, d\phi \; . \tag{148}$$

Thus, we can describe the distance traveled along the surface by the change in the values of two coordinates, θ and ϕ:

$$ds^2 = \{a^2 \cos^2 \theta + b^2 \sin^2 \theta\} \, d\theta^2 + a^2 \cos^2 \theta \, d\phi^2 \tag{149}$$

By specifying λ in the manner which we did we have virtually required that the major axis of the ellipsoid be parallel to the projection of the streamline in the x-z reference plane. In addition, by evaluating a and b in the manner described we have almost assured that the streamline is a section of the curve

$$\frac{\bar{x}^2}{b^2 \left[1 - \frac{\bar{z}^2}{a^2}\right]} + \frac{\bar{y}^2}{a^2 \left[1 - \frac{\bar{z}^2}{a^2}\right]} = 1 \; , \tag{150}$$

where $\bar{z}$ may be treated as a constant. In other words, the streamline of interest is still an ellipse but may not be in the meridional plane. It will, however, lie parallel to this plane. The element of distance traveled along the surface in the streamwise direction now becomes

$$d\bar{s}_1 = \sqrt{a^2 \left(1 - \frac{\bar{z}^2}{a^2}\right) \cos^2 \bar{\theta} + b^2 \left(1 - \frac{\bar{z}^2}{a^2}\right) \sin^2 \bar{\theta}} \; d\bar{\theta} \tag{151}$$

while the distance in the crossflow direction is given by

$$d\bar{s}_2 = \sqrt{\bar{z}^2 + a^2 \left(1 - \frac{\bar{z}^2}{a^2}\right) \cos^2 \bar{\theta}} \; d\bar{\phi} \tag{152}$$

$\bar{\theta}$ is a displacement about the z-axis in the plane z = constant while $\bar{\phi}$ is a displacement about the x-axis in the plane x = constant. In terms of these

coordinates, the element of distance along the surface is given by

$$d\bar{s}^2 = \left[a^2 \left(1 - \frac{\bar{z}^2}{a^2} \right) \cos^2 \bar{\theta} + b^2 \left(1 - \frac{\bar{z}^2}{a^2} \right) \sin^2 \bar{\theta} \right] d\bar{\theta}^2$$

$$+ \left[\bar{z}^2 + a^2 \left(1 - \frac{\bar{z}^2}{a^2} \right) \cos^2 \bar{\theta} \right] d\bar{\phi}^2 \tag{153}$$

or

$$d\bar{s}^2 = g_1^2 \, d\bar{\theta}^2 + g_2^2 \, d\bar{\phi}^2 \ . \tag{154}$$

For purposes of practical calculation we must still evaluate $\bar{z}$, $\bar{\theta}$, and $\bar{\phi}$, in terms of the original body coordinates. The transformation equations for these quantities we can show to be

$$\bar{z} = (z - z_1)\cos \lambda + (x - x_1)\sin \lambda \tag{155}$$

$$\bar{\theta} = \tan^{-1} \left(\frac{(x - x_1)\cos \lambda - (z - z_1)\sin \lambda}{y} \right) \tag{156}$$

$$\bar{\phi} = \tan^{-1} \left(\frac{\bar{z}}{y} \right) \tag{157}$$

We now have the tools we need to describe streamwise and crossflow coordinates locally on the body in terms of the coordinates of points on the body surface as well as to write the boundary layer equations in this curvilinear coordinate system. We will not go through the derivation of the boundary layer equations in a general curvilinear coordinate system; rather we will employ those given for this case by Cebeci et al. (Ref. 99), rewritten in the present notation:

$$\frac{1}{g_1} \frac{\partial u_{\bar{\theta}}}{\partial \bar{\theta}} + \frac{\partial u_n}{\partial n} + \frac{1}{g_2} \overset{\textcircled{1}}{\frac{\partial u_{\bar{\phi}}}{\partial \bar{\phi}}} + \frac{u_{\bar{\theta}}}{g_1 g_2} \overset{\textcircled{2}}{\frac{\partial g_2}{\partial \bar{\theta}}} = 0$$

$$\frac{u_{\bar{\theta}}}{g_1} \frac{\partial u_{\bar{\theta}}}{\partial \bar{\theta}} + u_n \frac{\partial u_{\bar{\theta}}}{\partial n} + \frac{u_{\bar{\phi}}}{g_2} \overset{\textcircled{3}}{\frac{\partial u_{\bar{\theta}}}{\partial \bar{\phi}}} - \frac{u_{\bar{\phi}}^2}{g_1 g_2} \overset{\textcircled{4}}{\frac{\partial g_2}{\partial \bar{\theta}}} = - \frac{1}{\rho g_1} \frac{\partial P}{\partial \bar{\theta}} + \nu \frac{\partial^2 u_{\bar{\theta}}}{\partial n^2}$$

$$\frac{u_\theta}{g_1}\frac{\partial u_{\bar\phi}}{\partial \bar\theta} + u_n \frac{\partial u_{\bar\phi}}{\partial n} + \frac{u_{\bar\phi}}{g_2}\frac{\partial u_{\bar\phi}}{\partial \bar\phi} + \frac{u_{\bar\theta}u_{\bar\phi}}{g_1 g_2}\frac{\partial g_2}{\partial \bar\theta} = -\frac{1}{\rho g_2}\frac{\partial P}{\partial \bar\phi} + \nu \frac{\partial^2 u_{\bar\phi}}{\partial n^2} \qquad (158)$$

The velocities and directions are defined in the sketch below:

Point of interest on body surface
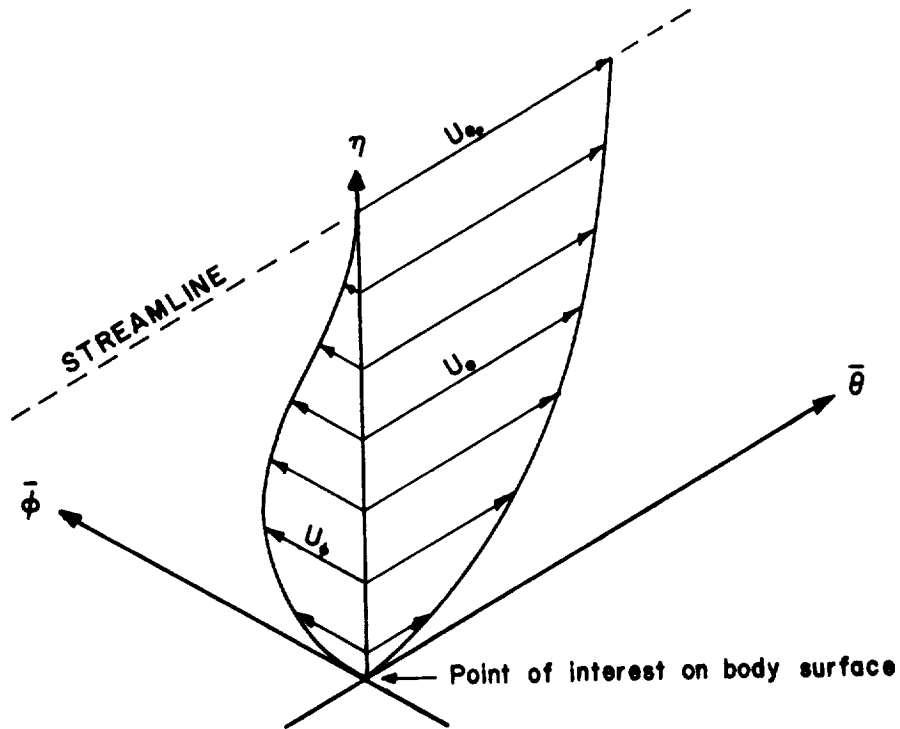
In these equations

$$\frac{\partial g_2}{\partial \bar\theta} = -\frac{\frac{1}{2}a^2\left(1-\frac{\bar z^2}{a^2}\right)\sin^2\bar\theta}{\sqrt{\bar z^2 + a^2\left(1-\frac{\bar z^2}{a^2}\right)\cos^2\bar\theta}} \cdot \qquad (159)$$

Despite the apparent generality of this form of the boundary layer equations we are still forced to assume when using them that the inviscid streamlines are displaced only in the n-direction by the presence of the boundary layer. We must do this in order that the metrics (g_1, g_2) we compute at a point on the surface apply vertically through the boundary layer.

The differences between the two-dimensional boundary layer equations developed earlier and the present three-dimensional treatment lie in the four numbered terms and the third equation. We note that if the circumferential variation in pressures at a given x-station along the body is small compared with the change in pressures in the streamwise direction, then there is no significant force available to produce a crossflow. Since this term, $\frac{\partial P}{\partial \bar{\phi}} \frac{1}{\rho g_2}$, must, because of the basis on which the equation was written, be at least as large in magnitude as any term in the equation, we are justified in ignoring the third equation whenever $\frac{\partial P}{\partial \bar{\phi}} \ll \frac{\partial P}{\partial \bar{\theta}}$.

We now need to find the conditions under which the other four terms may be ignored. Unfortunately we cannot demonstrate these easily in a rigorous fashion. But we will observe that if $\partial P/\partial \bar{\phi}$ is small then it is not likely that $u_{\bar{\phi}}$ will be large; in fact, it will probably be no more than about 1/10 as large as $u_{\bar{\theta}}$. In that event, term number (4) is probably small compared with terms on the right hand side of the equation. Again, if $\partial P/\partial \bar{\phi}$ is small we would not expect $u_{\bar{\theta}}$ to change substantially from one $\bar{\phi}$ value to another (for the same $\bar{\theta}$). This, combined with the small value of $u_{\bar{\phi}}$, means that term (3) probably is also quite small. By the same argument we can ignore term (1). Term (2), however, is not small in regions of large changes in body curvature, i.e., near the nose or the tail of the spheroid. $(\bar{\theta} \rightarrow \pm 90°)$. It is this term which accounts for the spreading of the flow as it moves straight downstream from the nose over the increasing surface area of the body.

It would appear, then, that as long as one observed the criteria

$$1. \quad \frac{\partial P}{\partial \bar{\phi}} \ll \frac{\partial P}{\partial \bar{\theta}}$$

and

$$2. \quad \bar{\theta} < \left| \frac{0.9\pi}{2} \right| \tag{160}$$

the use of a two-dimensional boundary layer calculation along streamlines would be a reasonable way of determining δ^* and τ_w.

By choosing a sufficient number of streamlines we can insure that at least one streamline will cross each panel somewhere near its center. The solution of the boundary layer equation along that streamline will then give us τ_w and δ^*

125

near the centroid of each panel. Because of the very great amount of work involved in successively modifying the body shape to account for displacement thickness effects, we will for the present, adopt the policy of using the first calculation of τ_w as the final value.

We could modify the body shape to account for displacement effects by adding

$$\delta^* = \frac{\bar{a}}{\sqrt{\bar{a}^2 + \bar{b}^2 + \bar{c}^2}} \qquad (161)$$

to the x-dimension of all point on the panel

$$\delta^* = \frac{\bar{b}}{\sqrt{\bar{a}^2 + \bar{b}^2 + \bar{c}^2}} \qquad (162)$$

to the y-dimension and

$$\delta^* = \frac{\bar{c}}{\sqrt{\bar{a}^2 + \bar{b}^2 + \bar{c}^2}} \qquad (163)$$

to the z-dimension, $\bar{a}$, $\bar{b}$, and $\bar{c}$ are given by equation (136).

However, the displacement thickness over the forward part of the body is usually small and the use made of this information on the rear part of the body in our drag calculations, as outlined below, is rather approximate. In these circumstances recomputation of the entire body contour does not appear to be warranted.

It must be constantly borne in mind that the approach used here for determining the pressures and velocities over the body surface is an inviscid one. As such, it always places a stagnation point at the downstream end of a closed body. As a result, the computation always leads to a prediction of rear stagnation pressures at the aft end of the body in what is physically a wake region. See the sketch below. The pressures in this wake are generally less than atmospheric immediately aft of the body and rise to the free stream value some distance downstream. The relatively low pressure exerted on that portion of the body's geometry opposite the high pressure stagnation region near the nose is the principal source of the force acting in the direction of the flow which we call form drag or pressure drag. Bodies producing extensive regions of flow separation will therefore have large pressure drags. It is apparent, therefore, that we must represent the wake effects with reasonable accuracy if we are to make meaningful drag computations.

126

Flow aft of a bluff body

WAKE $P \approx P_\infty$

Streamlined Body

WAKE

stagnation point

stagnation point

BODY IN INVISCID FLOW
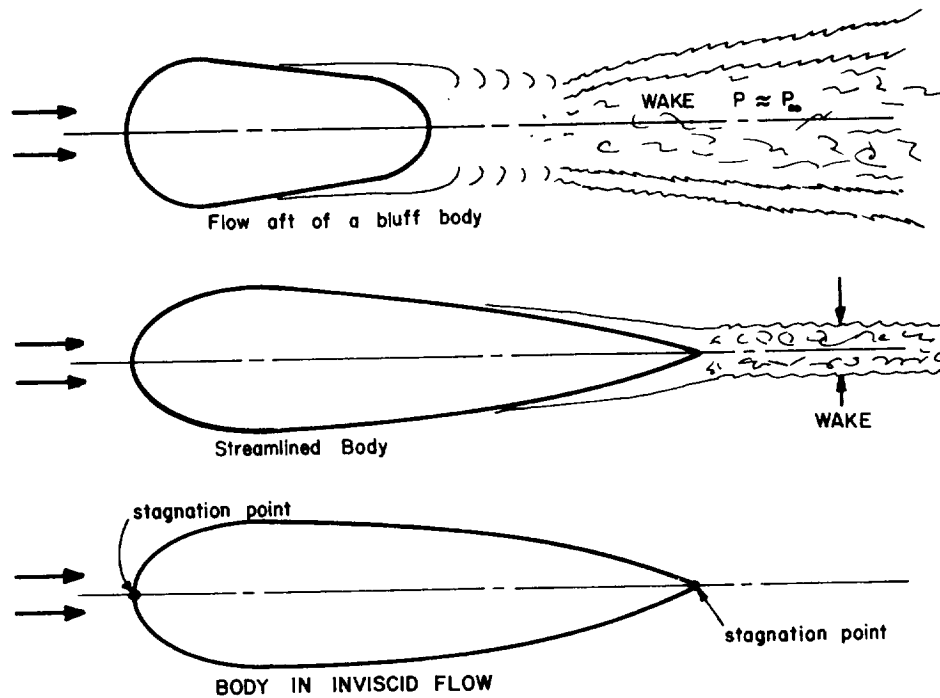
How to include such a representation within the framework of our procedure for determining the inviscid flow field is the hurdle we must clear. Since the wake is a region of "dead" air relative to the body it seems reasonable to represent it by assuming it to be part of the physical body for purposes of calculating the inviscid flow field. Then, by assigning quadrilaterals to the surface of this wake-body which are of approximately the same area as those on the physical body, one can redo the inviscid computation to find the pressures on the wake-body. See the sketch below. The pressures on those panels of the wake-body which lie immediately above equivalent panels (quadrilaterals) on the physical body are then applied to the panels on the physical body along the normals to the physical body. The forces acting on the physical body are summed to find a lift and a drag. Since the pressures on the upstream portion of the wake-body will, if its contours are properly chosen, be less than those on the rear of the physical body according to the inviscid flow computation, the integration of forces will indicate a net drag on the body. Note that the total number of panels considered in the analysis is greater when the wake-body is present.

127
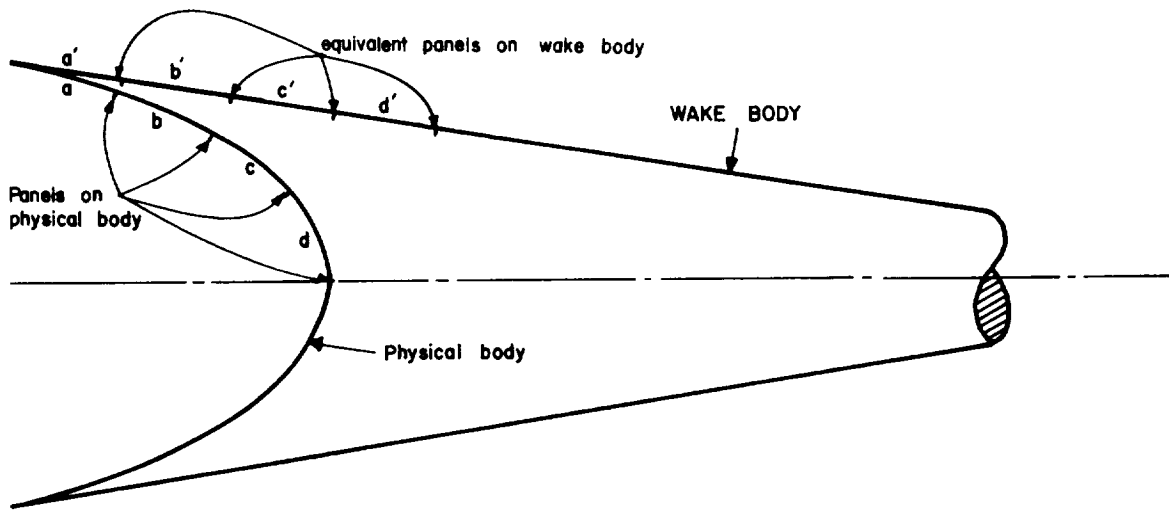
Pressures computed on panels a', b', c', and d' by the inviscid source distribution are applied to panels a, b, c, and d on the physical body before the integration of forces in the axial direction is carried out. Since these pressures are lower than those computed by source distribution on the physical body, a net drag is obtained on integration.

Some difficulty arises in attempting to specify the contours of the wake-body *a priori*. If the contours are significantly concave, the inviscid flow model will always produce a decelerating flow in these regions and, hence, rising pressures. The computed pressure drag will then be less than that actually present. Further, the boundary layer model used cannot predict the location of the actual flow separation point. Thus, there is some ambiguity in locating the upstream origin of the wake-body as well as how it should fair into the physical body.

Consider, however, the drag data (Ref. 112) for airship hulls with geometries given in the following table.

x/ℓ	$2y/t$	x/ℓ	$2y/t$	x/ℓ	$2y/t$
0	0	.150	.887	.600	.885
0.0125	.2	.200	.947	.700	.790
.0250	.335	.250	.982	.800	.665
.050	.526	.300	.998	.900	.493
.0750	.658	.350	.999	.950	.362
.100	.758	.400	.990	.980	.225
.125	.835	.500	.950	1.000	0

128
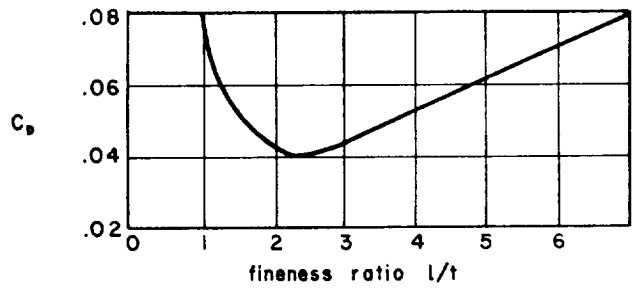
Note that the minimum combined skin friction and pressure drags occur at a fineness ratio of 2.1. The drag coefficient rises steeply for blunter bodies. It is thus, unlikely that fuselages will be blunter than a fineness ratio of 2 to 3. Calculations of the skin friction drag over a 3 to 1.0 ellipsoid (see sketch below) compared with the total measured drag indicate that the pressure

129
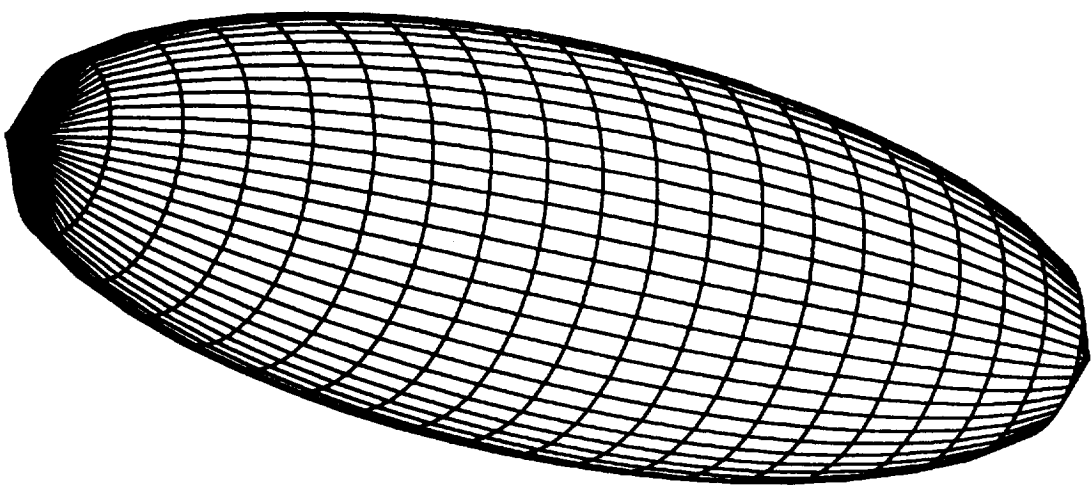
drag generated by such a body is about 25 percent of the total. As the
fineness ratio increases this fraction of course decreases. The total drag
is thus not particularly sensitive to small errors in determining the pres-
sure drag on streamlined bodies. Let us, therefore, arbitrarily assume that
the wake-body always begins at about the 0.9ℓ point on the physical body. We
will also assume that we can determine the initial surface slope of the wake-
body from our calculation of the displacement thickness as indicated in the
sketch below:
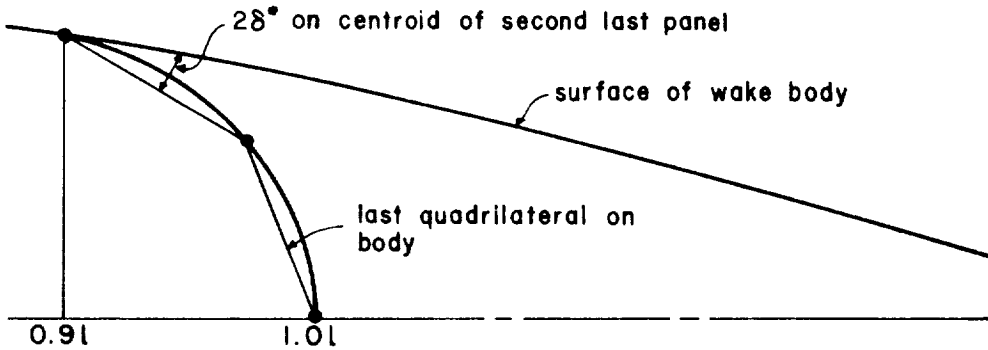
The wake-body surface we will gradually fair, via an exponential function, to
a point sufficiently far downstream that the surface of the wake-body can be
paneled with an even number of sets of panels in the streamwise direction.
The panel areas are to be the same size as those on the physical body. It
will be recognized that for thin wakes the termination point of the wake-body
with this scheme is of necessity very far downstream. For thicker wakes, the
termination point will be somewhat further upstream. The sketch below shows
how the 3 to 1.0 ellipsoid appears with a wake-body appended.

130
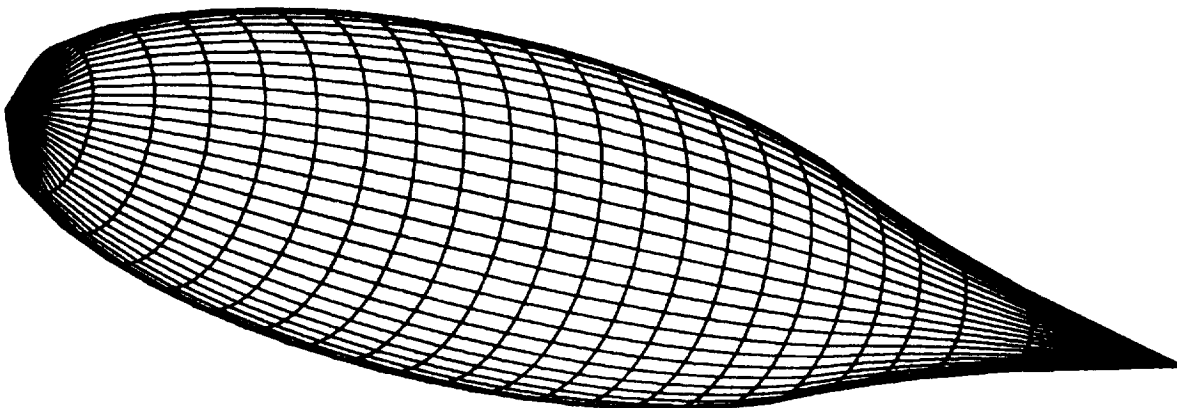
The reader will recognize that an ellipsoid of revolution is one of the simplest bodies possible. Because the procedure outlined above gives reasonable results in such cases, does not mean that it will necessarily give reasonable results for more complex bodies, if applied in an arbitrary and capricious fashion. More complex bodies—of which light aircraft fuselages are excellant examples—can produce local separations as the flow rounds the cowling or encounters the windscreen and more general separations in the area aft of the cabin, for example. None of these situations can really be treated properly by the boundary layer theory employed. All one can do is insure that the fuselage geometry is adequately presented to the inviscid flow field computation and that when boundary layer computations indicate that separation is imminent, force the flow to continue attached (for computational purposes) until a more favorable pressure gradient is encountered. Sometimes these measures will yield acceptable results, sometimes not. The user must in most cases examine the results with caution and in detail, relying on past experience and experimental evidence to determine whether they are reliable. It is unfortunate that the requirement for the exercise of such substantial engineering judgement in this procedure has not yet been eased. However, in defense of this situation one may point out that in contrast to other aspects of the prediction of aerodynamic characteristics of complete configurations, the estimation of fuselage drag in a rigorous analytical fashion is now in its infancy and undergoing considerable flux. The method related herein was assembled by the present writers in the absence of suitable techniques in the literature. With the intense activity in the field, however, it would be surprising indeed if it were not soon superceded by methods more easily applied and using more accurate boundary layer models.

We have discussed above the methods by which we find the pressure and skin friction on individual panels or quadrilaterals. To determine the lift, drag, and pitching moment of the entire body we begin by finding first the component in either the z (lift) direction or the x (drag) direction, of the pressure on each panel. A product of this pressure component and the panel area (assuming of course that the pressure determined for a particular point on the panel exists everywhere on that panel) gives the force contributed by the panel to the total body force along either the z or x-axis. The pressure on a panel is exerted parallel to the normal to the panel. Thus, the component of the pressure in the z-direction is given by

$$P_{z_i} = P_i \left(\frac{\bar{c}}{\sqrt{\bar{a}^2 + \bar{b}^2 + \bar{c}^2}} \right)_i \qquad (164)$$

while the component in the x-direction is

131

$$P_{x_i} = P_i \left(\frac{\bar{a}}{\sqrt{\bar{a}^2 + \bar{b}^2 + \bar{c}^2}} \right)_i \qquad (165)$$

Formally, then, the lift is simply

$$L = \sum_{i=1}^{k} P_{z_i} A_i \qquad (166)$$

where k is the number of panels on the body and A_i is the area of the ith panel. The pressure drag is

$$D_p = \sum_{i=1}^{k} P_{x_i} A_i \ . \qquad (167)$$

The pitching moment is easily seen to be

$$M = \sum_{i=1}^{k} P_{z_i} (x_o - x_i) A_i \ , \qquad (168)$$

where x_o is the moment reference point and x_i is the x-coordinate of the ith panel.

If we call γ_x, γ_y, and γ_z the direction cosines of the streamlines over the body, then the x-component of the skin friction on each panel, when summed over all panels on the body, becomes the skin friction drag:

$$D_f = \sum_{i=1}^{k} \tau_{w_i} \gamma_{x_i} A_i \ . \qquad (169)$$

Skin friction contributions to the lift and pitching moment can be found in an analogous fashion. The total fuselage drag is the sum of the pressure drag and the skin friction drag, i.e.,

$$D = D_f + D_p \ . \qquad (170)$$

132

CONCLUDING REMARKS

In our determination of the lift, drag, and moment characteristics of light aircraft fuselages we have generally employed a procedure which is characterized by the following steps:

1. Represent the surface of the isolated fuselage by a number of quadrilaterals or four-sided panels.

2. In effect move all four corners of the panel into the same plane through the procedure which determines the direction of the normal.

3. Place a source of undetermined strength on each panel and require that the velocity induced by all sources plus that due to the free stream in the direction normal to the surface of each panel be zero.

4. Solve the resulting system of equations for the source strengths.

5. Compute the velocity over the body surface. From this, determine streamlines and surface pressures.

6. Perform two-dimensional, momentum-integral-type boundary layer computations along streamlines to find local values of τ_w and δ^*.

7. Integrate τ_w over the surface to find the skin friction drag of the isolated fuselage.

8. Modify the body shape by attaching a wake-body toward the trailing edge and enlarging, if desired, the remainder of the body to account for displacement thickness effects.

9. Compute a new set of source strengths and surface pressures corresponding to the wake-body shape.

10. Integrate the surface pressures to find the lift, pressure drag, and pitching moment of the fuselage.

This procedure of course applies only to an isolated fuselage and does not account for flow interactions between the fuselage and the wing or tail plane. These are discussed in the following chapter. The procedure also does not account for the effects on drag of small protuberances such as extended landing gear. A suggested means of tying together the results of the various computations in this, the previous chapter, and the chapter on interactions, along with empirical results for small protuberances, is presented in the chapter on Lift and Drag Estimation of Complete Configurations.

We should mention also that the method developed here applies only to fuselages so inclined as to have minimum circumferential pressure gradients. This was done, it will be recalled, to permit the use of a simple boundary layer computation procedure along streamlines. With the need for many panels to adequately represent the body and the desire to have at least one streamline go through each panel near its centroid, the number of boundary layer calculations required to accomplish this is very substantial. In essence we are substituting a large number of two-dimensional computations which we know how to do for a smaller number, perhaps, of difficult and involved three-dimensional computations with which we have had little experience as yet. For this reason and the very large computer capacity required even for such an approach, we have not considered the case of flow approaching the fuselage with a vertical or sidewise component. The reader will recognize that because the formulation of our problem is linear, the effects of such components can be superimposed on the effects of the axial component. To include one of these components, however, is the equivalent, computationally, of repeating the problem. Considering only an axial onset flow limits the applicability of our treatment to cases where the fuselage angle of attack is always near zero, i.e., near cruise. We cannot, therefore, take adequate advantage of the ability of the wing program to yield reliable results up to C_L's of about 0.8 except to account for variations in wing incidence angle. It is to be expected, however, that such restrictions will be relaxed through the use of improved methods which should be available from the major airframe manufacturers within the next few years.

134

INTERFERENCE EFFECTS

In our previous discussion we mentioned that while we perform most of our aerodynamic calculations on isolated fuselages and wings we did realize that when these components are joined together to form practical configurations there are interactions between them not accounted for by the treatment of isolated components. We intimated that we would approach these interaction effects largely on an empirical basis. As may become evident from the discussion below, we are virtually forced to take this approach by the complexity of the problem.

WING-TAIL INTERFERENCE

An aircraft's horizontal tail plane is flying in a very non-uniform flow field at least compared with the one seen by the main wing. In developing lift, the main wing imparts a downward component to the flow over it, a component which is not uniform across the span. Further, the wake produced by the main wing is a sheet-like region of low dynamic pressure which diffuses slowly as it moves downstream. These phenomena make it difficult to determine the orientation and magnitude of the flow striking the tailplane. Ferrari in Ref. 98 describes the situation aptly:

"For wings of aspect ratio greater than 4.0 and sweep angles of less than 30°, the wings' system of bound vortices may be taken to be compressed into one single vortex, the axis of which is normal to the plane of symmetry and passes through the mean aerodynamic center of the lifting surface. The single vortex has a circulation that is variable in strength along the span; a trailing vortex sheet is shed across the whole breadth of this bound vortex core. The actual shape of the vortex sheet is rather difficult to specify inasmuch as it depends upon the particular velocities induced by the vortex system at the points occupied by the trailing vortex sheet.

The problem is complicated even more seriously because the trailing band of vortices is unstable. As a result of this instability the band tends to roll up at the edges in such a way that two distinct vortical corelike nuclei of appreciable size are formed, which are characterized by equal and opposite amounts of circulation....the complete rolling up of the vortices is accomplished only at a certain distance behind the rear edge of the wing, which may be calculated from the relationship $0.56 \frac{b}{2} \frac{AR}{C_L}$ *"* where the symbol AR denotes aspect ratio and b is the wing span.

For most cases of interest however *"the region of the vortex sheet which stands nearest to the stabilizer...is not yet involved in the roll-up phenomena."*

The usual way to treat the problem of determining the direction and magnitude of the flow approaching the stabilizer is to assume first of all that the wake is non-existent. This enables one to use an extension of the lifting line theory (as we shall do below) to find the flow angle anywhere in the downwash field. The effect of the wake is then added in a semi-empirical fashion.

Referring now to the figure following equation (82) we may, as we did in obtaining equation (87), write an expression for the downwash component of the flow velocity at any point aft of the wing, which includes contributions from the bound vortex (that running along the wing span) and the two trailing vortices:

$$w = \frac{\Gamma}{4\pi} \left\{ \frac{-x}{x^2 + z^2} \left[\frac{y + b/2}{\sqrt{x^2 + z^2 + (y + b/2)^2}} - \frac{y - b/2}{\sqrt{x^2 + z^2 + (y - b/2)^2}} \right] \right.$$

$$- \frac{y - b/2}{z^2 + (y - b/2)^2} \left[1 - \frac{x}{\sqrt{x^2 + z^2 + (y - b/2)^2}} \right] \qquad (171)$$

$$\left. + \frac{y + b/2}{z^2 + (y + b/2)^2} \left[1 - \frac{x}{\sqrt{x^2 + z^2 + (y + b/2)^2}} \right] \right\}$$

Positive z-direction is downward and positive x-direction is upstream. The other components of the induced velocity are

$$u = \frac{\Gamma}{4\pi} \frac{z}{x^2 + z^2} \left\{ \frac{y + b/2}{\sqrt{x^2 + z^2 + (y + b/2)^2}} - \frac{y - b/2}{\sqrt{x^2 + z^2 + (y - b/2)^2}} \right\} \qquad (172)$$

$$v = \frac{\Gamma}{4\pi} \frac{z}{z^2 + (y - b/2)^2} \left\{ 1 - \frac{x}{\sqrt{x^2 + z^2 + (y - b/2)^2}} \right\}$$

$$- \frac{\Gamma}{4\pi} \frac{z}{z^2 + (y + b/2)^2} \left\{ 1 - \frac{x}{\sqrt{x^2 + z^2 + (y + b/2)^2}} \right\} . \qquad (173)$$

The foregoing expressions give the velocity components induced by a single horseshoe vortex. In the usual case where it is necessary to represent

136

the actual distribution of circulation by superimposing a number of simple horseshoe vortices, the induced velocity components are calculated by replacing Γ by $d\Gamma/d\eta$ and $b/2$ by η in (171, 172, and 173) and the integrating the resulting expressions from $\eta = 0$ to $\eta = b/2$. $\Gamma(\eta)$ is known beforehand from the procedure which determines the three-dimensional lift distribution on the wing. Since this integration must be carried out for each point of interest, the mapping of the flow direction and magnitude in the region near the tail is a laborious process. The flow direction in the x-z plane is found from the expression

$$\varepsilon = \sin^{-1}\left[\frac{w}{V + u}\right] , \qquad (174)$$

while the magnitude is given by

$$\sqrt{w^2 + (V + u)^2} . \qquad (175)$$

If the stabilizer is placed about three chord-lengths behind an aspect ratio 6.0 wing, it will be found that the downward angle will be about $2SC_L/\pi b^2$ radians. For $C_L = 0.6$ this is $3.64°$.

The center of the vortex sheet describes a path in an x-z plane which can be found by integrating ε with respect to x from the trailing edge of the wing rearward. Ferrari gives an expression, developed from experimental data, for the vertical displacement of this vortex sheet center which occurs because of the influence of the wake:

$$\Delta z = -\frac{1}{2} C_s \sin \alpha , \qquad (176)$$

where C_s represents the length of separated flow over the airfoil (primarily the upper surface). The semi-thickness of the airfoil wake he states as

$$z_{wake} = 1.38 \ c \ C_{d_o}^{\frac{1}{2}} \left[\frac{x}{c} + 0.15\right]^{\frac{1}{2}} , \qquad (177)$$

where C_d is the profile drag coefficient applying to the airfoil section located at the spanwise station at which the wake's behavior is being investigated. The magnitude of the velocity along the wake centerline Ferrari gives as

$$u_{w_c} = V\sqrt{1 - \frac{2.42\left[C_{d_o}\right]^{\frac{1}{2}}}{\frac{x}{c} + 0.3}} \qquad (178)$$

137

while the distribution of velocity vertically through the wake is

$$u_w = V \sqrt{1 - \frac{2.42 \left[C_{d_o} \right]^{\frac{1}{2}}}{\frac{x}{c} + 0.3} \cos^2 \left(\frac{\pi \tilde{z}}{z_{wake}} \right)} \tag{179}$$

$\tilde{z}$ is to be measured from the wake centerline and $\tilde{z}/z_{wake} \leq 1/2$.

To summarize the procedure related above, we described the trace of the center of the vortex sheet shed by the wing in x-z plane at any value of y by an integration with respect to x of equation (174). Since this is an inviscid theory, we apply empirical corrections to account for viscosity effects. The actual wake centerline is displaced from the value given by the integration of (174) by an amount Δz from (176). The actual magnitude of the flow velocity across the wake is given by (179) rather than (175). Outside the wake region, the magnitude of the flow velocity is given by (175). The magnitude given by (175) should also be used in place of V in (179). These relationships are depicted in the figure below:

SECTION A-A

138
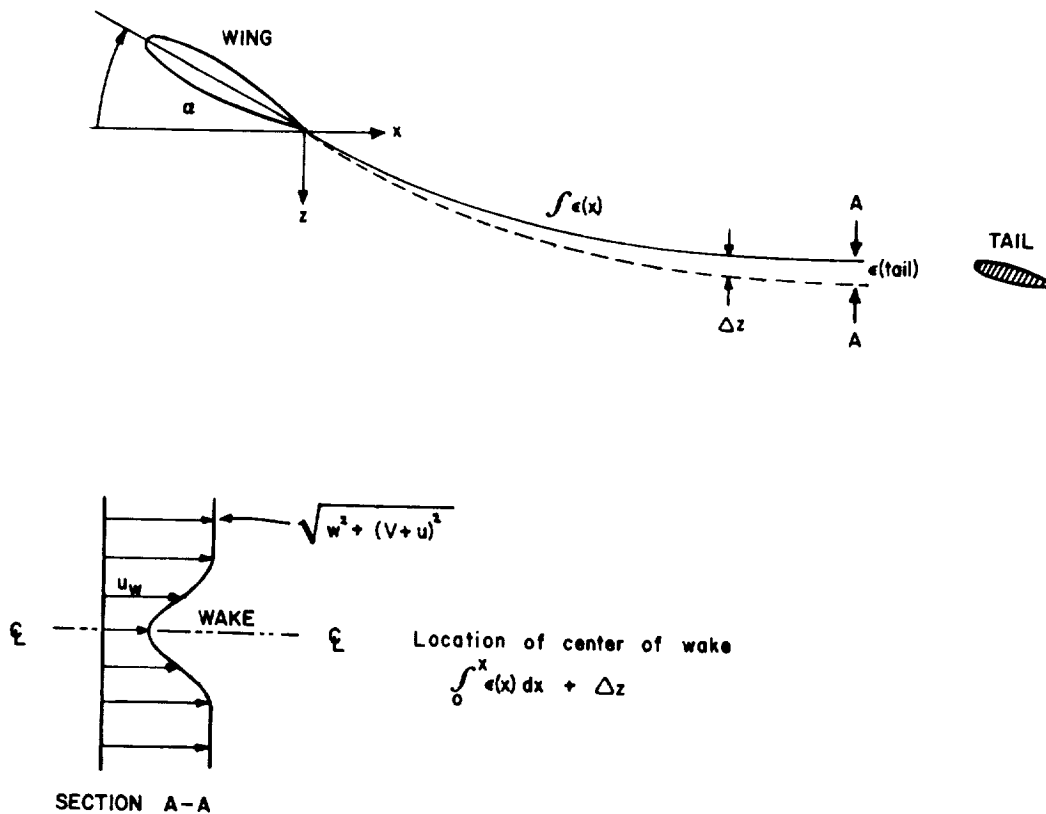
From the foregoing we have sufficient information to develop at least a semi-quantitative view of the disturbance which the wing produces in the flow field approaching the horizontal tail plane. With this view we can then proceed to calculate with reasonable accuracy the aerodynamic characteristics of this surface considering it to be a portion of a complete airframe. We do this by employing the theory for a complete wing in the presence of a fuselage developed in the previous chapter but substituting local values of $(\alpha - \epsilon)$ for α and u_w for V (if the tail plane is flying in the wing wake). Such a procedure ignores the presence of a thick boundary layer over the aft portion of the fuselage, the disturbance to the flow field caused by the vertical stabilizer, and the confluence of the body flow immediately downstream of the stabilizer. Since a thick boundary layer tends to smooth out disturbances and since it tends to make the fuselage appear to the flow to be longer and less rapidly varying in area than it actually is, it seems reasonable to conclude that for most configurations in cruise flight the effects ignored are small.

LIFT OF AN ISOLATED FUSELAGE

It was pointed out in the discussion of the literature that a fuselage at angle of attack in inviscid flow should experience no net lift. Any lift generated is a result of viscous effects. In principle we could obtain the correct result by extending our treatment of forces on an isolated fuselage as presented in the previous chapter to include a vertical component in the onset flow and a boundary layer computation method which accepts circumferential pressure gradients and expanding flows. (It is of course possible that bodies of unusual shape will also generate lift in much the same way that wings do but we have not as yet added planar vortices to our treatment of the pressures on the fuselage to account for this.) Unfortunately, the procedure would assuredly be very lengthy. Since it is desirable in design work to have an indication of the relative importance of certain effects so that we know how precisely we will have to evaluate them, we would like to have a simpler means for establishing the magnitude of the fuselage lift and for identifying its source physically. One such means follows from the argument given below.

A very long circular cylinder whose axis is at angle α with the oncoming flow has a flow about it which can be thought of as being made up of two non-interacting parts: first, an axial flow which cannot affect the lift although it does result in a skin friction drag; second, a crossflow which exhibits the same characteristics as the flow over a cylinder placed normal to a stream of velocity $V \sin \alpha$. For Reynolds numbers between 4×10^2 and 2×10^5 one finds from experiments (Ref. 65) that the drag coefficient of a cylinder normal to a stream is always about 1.1. This is, of course, a result of the separation of the flow from the cylinder near the 90° points and the creation of a large low pressure wake on the lee side. If the cross flow Reynolds number lies within this range, one would expect that a long, isolated cylinder will experience a force normal to its axis given by

$$N = 1.1 \frac{\rho}{2} (V^2 \sin^2 \alpha) \ell d \ . \tag{180}$$

ℓ is the length of the cylinder and d is its diameter. The force normal to the stream direction is then

$$L = N \cos \alpha \ . \tag{181}$$

Normalizing this force by the usual $1/2 \ \rho V^2 \ d\ell$ gives

$$C_L = 1.1 \sin^2 \alpha \cos \alpha \ . \tag{182}$$

Most fuselages are not very long circular cylinders but have instead a nose which, even for the flow model assumed, i.e., noninteracting axial and crossflow components, experience an inviscid lift force. Usually, this force would be balanced by an equal and opposite force on the tail in inviscid flow, resulting in the development of a pitching moment but no net lift. If, however, the body is cut off so that the flow separates from the base and forms a viscous wake, the body appears to the inviscid flow to extend downstream for some distance at essentially constant diameter. The balancing force is then not developed and we are left with a net lift which is said to be potential or inviscid in origin. Following Ref. 101, we shall show how this arises.

We recall that the potential for the crossflow at station x on an infinite circular cylinder in cylindrical coordinates is written

$$\phi = - V \sin \alpha \left(r + \frac{a^2}{r} \right) \cos \theta \tag{183}$$

where a is the radius of the cylinder at station x, r is the radial coordinate, and θ is the angular coordinate measured from the forward stagnation point. The crossflow may also change in time as the flow moves along the axis of the cylinder so that the appropriate form of the Bernoulli equation giving the relation between pressure and velocity along a streamline is

$$\frac{P}{\rho} = - \frac{\partial \phi}{\partial t} - \frac{1}{2} \left[\left[\frac{\partial \phi}{\partial r} \right]^2 + \left[\frac{\partial \phi}{r \partial \theta} \right]^2 \right] + c \ . \tag{184}$$

The three partial derivatives are given by

140

$$\frac{\partial \phi}{\partial t} = -V \sin \alpha \, \frac{\cos \theta}{r} \, 2a \, \frac{da}{dt} = -V \sin \alpha \cos \theta \, \frac{2a}{r} \frac{da}{dx} \frac{dx}{dt}$$

$$\frac{\partial \phi}{\partial r} = -V \sin \alpha \, (1 - \frac{a^2}{r^2}) \cos \theta \; . \qquad (185)$$

$$\frac{\partial \phi}{r \partial \theta} = + \frac{V \sin \alpha}{r} (r + \frac{a^2}{r^2}) \sin \theta$$

Inserting these values into the Bernoulli Equation yields

$$\frac{P}{\rho} = 2V^2 \sin \alpha \cos \alpha \, \frac{da}{dx} \frac{a}{r} \cos \theta - \frac{V^2 \sin^2 \alpha}{2} \left\{ \cos^2 \theta \left(1 - \frac{a^2}{r^2} \right)^2 \right.$$

$$\left. + \left(1 + \frac{a^2}{r^2} \right) \sin^2 \theta \right\} + c \qquad (186)$$

As $r \to \infty$, we must assume that P becomes P_∞, the freestream value at infinity; hence

$$c = \frac{P_\infty}{\rho} + \frac{V^2 \sin^2 \alpha}{2} \; . \qquad (187)$$

On the surface of the cylinder $r = a$. At this location the pressure is, accordingly,

$$\frac{P}{\rho} = 2V^2 \sin \alpha \cos \alpha \, \frac{da}{dx} \cos \theta - \frac{V^2 \sin^2 \alpha}{2} \left[4 \sin^2 \theta \right] + \frac{P_\infty}{\rho} + \frac{V^2 \sin^2 \alpha}{2}$$

$$= 2V^2 \sin \alpha \cos \alpha \, \frac{da}{dx} \cos \theta + \frac{V^2 \sin^2 \alpha}{2} \left[1 - 4 \sin^2 \theta \right] + \frac{P_\infty}{\rho} \qquad (188)$$

The force exerted by the crossflow, per unit length, on the cylinder is, as may be seen in the sketch below,

Cross Flow → | Normal Force →
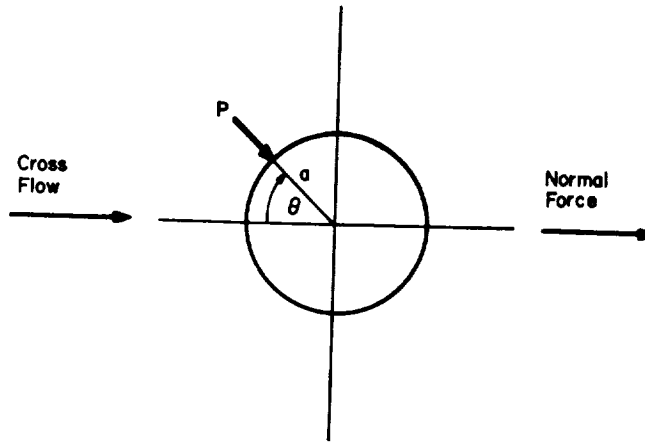
$$\frac{N}{\ell} = \int_0^{2\pi} P \ a \cos \theta \ d\theta \ = \frac{\rho V^2}{2} \left[\int_0^{2\pi} 2 \ a \ \sin 2 \ \alpha \ \frac{da}{dx} \cos^2 \theta \ d\theta \right.$$

$$\left. + \sin^2 \alpha \int_0^{2\pi} a(1 - 4 \sin^2 \theta) \cos \theta \ d\theta \right] + a \int_0^{2\pi} P_\infty \cos \theta \ d\theta \ . \tag{189}$$

Since

$$\int_0^{2\pi} \cos \theta \ d\theta = 0 \quad \text{and} \quad \int_0^{\pi} \cos^3 \theta \ d\theta = 0 \ ,$$

$$\frac{N}{\ell} = \frac{\rho V^2}{2} 4 \ a \ \sin 2 \ \alpha \ \frac{da}{dx} \int_0^{\pi} \cos^2 \theta \ d\theta$$

$$= \frac{\rho V^2}{2} 2\pi a \ \frac{da}{dx} \sin 2 \ \alpha \tag{190}$$

In producing a force on the cylinder the inviscid flow is deflected so that the lift is considered to act at an angle $\alpha/2$ from the normal force:

$$\frac{L}{\ell} = \frac{\rho V^2}{2} 2\pi a \ \frac{da}{dx} \sin 2 \ \alpha \ \cos \frac{\alpha}{2} \tag{191}$$

This inviscid lift is developed by the body in addition to the viscous lift described earlier. The total lift is the sum of the two effects. We write therefore

142

$$\frac{L}{\frac{\rho V^2}{2}} = \int_0^\ell \left[2\pi a(x) \; \frac{da}{dx} \; \sin 2\alpha \; \cos \frac{\alpha}{2} + \; 2a(x) \; 1.1 \; \sin^2 \alpha \; \cos \alpha \right] dx \tag{192}$$

If we choose our reference area to be ℓd then

$$C_L = \frac{2}{\ell d} \int_0^\ell \left[\pi a(x) \; \frac{da}{dx} \; \sin 2\alpha \; \cos \frac{\alpha}{2} + 1.1 \; a(x) \; \sin^2 \alpha \; \cos \alpha \right] dx \tag{193}$$

We note with interest that inviscid lift is present only when $\frac{da}{dx} \neq 0$, that is, primarily in the nose region of the body. We can easily make an approximate comparison between the predictions of this theory and experimental results taken on an ogive-cylinder-boattail (ℓ/d = 12.7, d = 5, nose length = 26.25) by assuming that the ogive can be satisfactorily represented by a cone and the boattailing can be ignored. Then da/dx = 0.125, a(x) = 0.125x for x from 0 to 20; da/dx = 0, a(x) = 2.5 for x from 20 to 63.5. With these numbers

$$C_L = \frac{2}{317.5} \left[\pi \sin 2\alpha \; \cos \frac{\alpha}{2} \; \frac{x^2}{128} \Big|_0^{20} + 1.1 \; \sin^2 \alpha \; \cos \alpha \; \left(\frac{x^2}{16} \Big|_0^{20} + 2.5x \Big|_{20}^{63.5} \right) \right] \tag{194}$$

$$C_L = 0.0619 \; \sin 2\alpha \; \cos \frac{\alpha}{2} + 0.573 \; \sin^2 \alpha \; \cos \alpha \tag{195}$$

The table below presents the results obtained with this formula and the comparable experimental data. Other experimental data on bodies of revolution may be found in Ref. 95.

α	C_L Theory	C_L Experiment Ref. 96
4°	.0114	.0108
8°	.0280	.0233
12°	.0494	.0376
16°	.0744	.0536
20°	.1022	.0713

143

Qualitative agreement is seen to be quite good. The lack of quantitative agreement is no doubt a result of our failure to describe the shape of the experimental body with sufficient accuracy since the comparisons with experiment presented in Ref. 101 show good correlations.

To apply this theory properly to isolated fuselages, we can see from the foregoing development that we need to know

(a) the fuselage geometry,

(b) the crossflow drag coefficient at each axial station on the fuselage,

(c) the crossflow potential function at each axial station, and

(d) the extent of the fuselage over which the axial flow is essentially inviscid.

The latter we need in order to determine the axial extent over which we may need to consider potential forces. Frequently this point is taken to be the fuselage station at which the wing intersects the fuselage. The crossflow drag coefficient and crossflow potential function are often difficult to determine for non-circular bodies. We do know, however, that the crossflow drag coefficient for Reynolds numbers in the range 10^2-10^5 is always less than about 1.4. (It has this value for a flat plate normal to the wind.) Since the inviscid lift term in (195) is linear and approximately equal to the viscous term for $\alpha = 12°$, it seems reasonable to conclude that it would seldom be more than 30 percent greater than for the experimental configuration considered here. Thus, for all angles of attack of interest, the lift of an isolated fuselage can be expected to be less than 5 percent as great as the lift of a wing of the same planform area. Under conditions of cruising flight, the fuselage lift is probably around 1 percent of the wing lift. With the uncertainties involved in determining wing-body interference effects of this magnitude, fuselage lift alone can be justifiably ignored.

INTERACTION BETWEEN WING LIFT AND FUSELAGE LIFT

As long as the fuselage is long and thin and either circular or elliptical in cross section and the flow about the wing-body combination can be regarded as inviscid, Multhopp's transform technique discussed earlier adequately describes the effect of wing-body interference on overall lift. These conditions are seldom obtained in practice, however. It is of interest, therefore, to examine what method practicing designers use to describe this effect. Ref. 97 suggests that one assume a linear variation of lift with angle of attack and consider that the effect of the body on the wing lift, the wing on the body lift, and total wing-body lift are as given by the ratios in the following table:

144

body diameter / wing span	L_{WB}/L_W	L_{BW}/L_W	L/L_W
0	1.0	0	1.0
0.05	0.99	0.057	1.047
0.1	0.972	0.1215	1.094
0.15	0.956	0.170	1.127
0.2	0.936	0.224	1.16
0.25	0.907	0.270	1.18
0.30	0.882	0.308	1.19

L_{WB} = wing lift in presence of body

L_{BW} = body lift in presence of wing

L_W = lift of wing without body

L = Total lift

Examining the quantities in the table shows that as the body grows and begins to shield the wing, the wing lift falls off, but not as rapidly as the fuselage diameter increases. This indicates a substantial "bridging" effect and is about what one would predict from the Multhopp theory. The body's contribution to the total lift grows in direct proportion to its diameter. The total wing-body lift for fuselage width-to-span ratios typical of light airplanes is about 10 percent greater than for an isolated wing because of the flow wake existing over the fuselage. On the basis of our previous argument concerning the lift of isolated fuselages, it would appear that the interaction between the wing and the fuselage enhances the fuselage's contribution to the total lift. Experience with the flow over cylinders inclined to a stream (Ref. 102) would indicate that these effects actually are more non-linear with α than would be suggested by the table above. In particular, if $\alpha < 6°$, a common condition, certainly, for cruising flight, the crossflow wake is weaker than would be expected from linear considerations and the total lift under these conditions is probably about the same as for the wing alone or perhaps as small as that given by Multhopp's transform. Thus, it would seem reasonable to ignore lift interaction effects unless boundary layer calculations on the body alone indicate the likelihood of the formation of a wake on the upper side of the fuselage which would grow with small increases in angle of attack.

DRAG INTERACTION EFFECTS

During preliminary design it is usually assumed (Ref. 53) that the drag of wing-body combinations is the sum of the skin friction drag of the body and wing separately times a mutual interference factor plus the form drag of the combination. This mutual interference factor for the Mach numbers of interest in light aircraft is usually taken to be about 1.06 for Reynolds

numbers below 2×10^7 and 0.93 for Reynolds number based on fuselage length above 6×10^7. A light aircraft with a 30' fuselage moving at 150 mph will have a Reynolds number of about 4.5×10^7. Thus for cruise conditions this interaction criterion would suggest that the drag interference effects are small.

Such interference criteria, however, cannot account for interactions which result in locally adverse pressure gradients on the wing or fuselage. These gradients often lead to flow separation (and accompanying high drag) at places like wing-fuselage junctions. Correction of these drag problems is usually relegated to the flight phase of the development program, the process being called "drag cleanup."

146

LIFT AND DRAG OF COMPLETE CONFIGURATIONS

In the preceding chapters we have set forth theoretical methods by which one can determine the aerodynamic characteristics (lift, drag, and pitching moment) of airfoil sections and isolated fuselages at low and moderate angles of attack. We also presented an analytical method, valid for unswept wings at moderate-to-high aspect ratios, for correcting two-dimensional aerodynamic characteristics for the presence of the fuselage, taper, twist, changes in camber, and the finite extent of the wing. In addition, we discussed an analytical means by which one might determine the magnitude and direction of the flow approaching the horizontal tailplane. Further, we were able to show that for low-to-moderate angles of attack the fuselage contribution to the aircraft lift is usually negligible. We also noted the semi-empirical means by which drag interaction effects are frequently treated.

These various methods, interesting as they may be in themselves, remain largely academic exercises for the designer unless he can employ them effectively in the estimation of the lift and drag of complete configurations. It is really only in tasks of this kind that the advantages of more rigorous estimation procedures can be fully appreciated. For many reasons it is not yet practical to attempt to consolidate the various methods into a single computer program, the inner workings of which need not concern the user. It therefore requires some understanding of the basis for the various methods in order to apply them all to the same configuration and obtain useful results. We shall outline below a step-by-step procedure which should accomplish this objective.

1. Select the basis of the force coefficients. Normally we will take the wing planform area, including the portion covered by the fuselage, as the basic area.

2. From the geometry of the wing, including

    (a) airfoil ordinates at several spanwise stations (at least root and tip)
    (b) taper
    (c) twist
    (d) aspect ratio
    (e) fuselage cross section at wing root and location of wing in relation to fuselage center,

determine the lift, drag, and pitching moment coefficients of the wing and the spanwise variation in circulation. These data come from the methods discussed in the chapter on wing characteristics. The spanwise circulation, it may be noted, is proportional to the final form of the spanwise variation in section

147

lift coefficient for a given α. This information will be used later when we determine the flow conditions approaching the horizontal tailplane. Note also that for good results we should choose flight conditions for which $C_L \leq 0.8$.

3. Determine the drag <u>force</u> on the fuselage using the methods discussed in the chapter on fuselages. The methods will also give fuselage lift and pitching moment. The moment will be of interest if one seeks to perform stability computations at some later time. The lift given by these methods includes both the inviscid and viscous lifts discussed in the chapter on interference but of course does not include wing-body interference effects. This fuselage lift properly can be added to the wing lift; because it should be small compared with the wing lift, its computation can also serve as a check on the fuselage drag computation. If the fuselage lift is large, the fuselage drag values should be suspect.

4. Normalize the fuselage drag <u>force</u> by the wing area and the free-stream dynamic pressure.

5. Compute the magnitude and direction of the flow approaching the horizontal tailplane.

6. With these data, use the methods of (2) above to determine the lift, drag, and pitching moment of the horizontal tailplane.

7. Normalize by the wing area and freestream dynamic pressure.

8. Add the lift coefficients for the wing, fuselage, and horizontal tailplane together to obtain an overall lift coefficient. Do the same for drag and moment coefficients.

9. Multiply the total drag coefficient by a factor ranging from 1.06 at low Reynolds numbers ($< 2 \times 10^7$) to 0.93 at high Reynolds numbers ($< 6 \times 10^7$). Reynolds number is based on fuselage length.

10. Add in drag increments for protuberances as in Ref. 2 (C_{D_π} method) and for those interference effects which experience or test data indicates may be greater than normal. The C_{D_π} method is summarized below.

Protuberances such as handles, hinges, antennas, cover plates, etc., impose additional drag on the aircraft but their effects are not readily treated within the analytical framework discussed earlier. It has been common to account for such effects in the following manner:

The drag produced by some element of the aircraft can be written

$$D_e = C_{D_e} q S_w$$

148

where C_{D_e} represents the drag coefficient of the element based on the wing area. It is, however, difficult to estimate the drag coefficient of many protuberances in terms of the wing area. These estimates are frequently made by comparing the protuberance with some simple shape which has been tested frequently. For example, the drag of a vertical antenna wire would be estimated by considering the wire to be a cylinder normal to a uniform stream. For most Reynolds numbers, the drag coefficient of such cylinders is given in classic texts as 1.2-1.3 based on a product of the wire diameter and its length. If one calls such a "natural" area A_π and the drag coefficient based upon it C_{D_π}, then D_e is also given by

$$D_e = C_{D_\pi} \, q \, A_\pi \, .$$

In terms of C_{D_π} then,
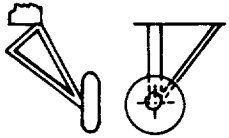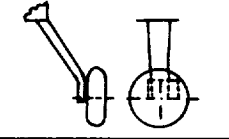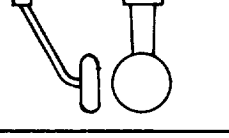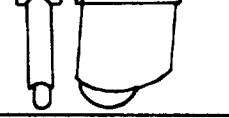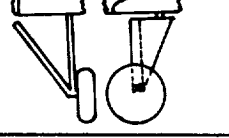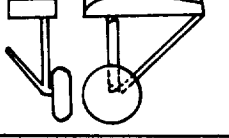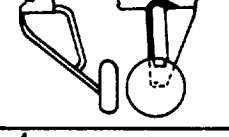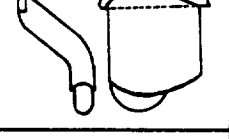
$$C_{D_e} = \frac{C_{D_\pi} A_\pi}{S_w} \, .$$

The net contribution of all protuberances to the total airplane drag coefficient based on the wing area is thus simply

$$C_{D_{Protuberances}} = \frac{\sum_\pi C_{D_\pi} A_\pi}{S_w} \, .$$

The same rationale applies to the contributions to the vehicle drag provided by the fuselage and tail surfaces. The reason for basing all contributions to the overall drag on the wing area is simply one of convenience. The wing area is the basic geometric parameter used in sizing the airplane during preliminary design and is therefore the parameter governing initial estimates of power required and performance.

Given below are C_{D_π} values and the areas upon which they are based for some landing gear designs, nacelles, wing tanks, and wires and struts.

149

Configuration	Remarks	$f=C_{D_\pi}A_\pi$
	8.50-10 wheels, not faired . . . . 8.50-10 wheels, faired . . . . 8.50-10 wheels, no streamline members . .	1.67 1.50 3.83
	8.50-10 wheels, faired . . . . 27-in. streamlined wheels, not faired	0.74 0.98
	27-in. streamlined wheels, not faired . . 8.50-10 wheels, faired . . . . 21-in. streamlined wheels, not faired . .	0.84 0.68 0.53
	8.50-10 wheels	0.51
	8.50-10 wheels, not faired . . . . 8.50-10 wheels, faired	1.52 1.02
	8.50-10 wheels, not faired	1.60
	24-in. streamlined wheels, & intersections filleted . . . . 8.50-10 wheels, no fillets	 0.86 1.13
	8.50-10 wheels	1.05
	Low pressure wheels, intersections filleted Low pressure wheels, no wheel fairing . . Streamlined wheels, round strut, half fork no fairing	0.31 0.47
Nose Gear	For the nose gear C_{D_π} = .5→.8 based on A_π = (wheel diameter)(wheel width)	

COMPONENTS	AREA FOR DRAG CALCULATION	C_{D_π}
Nacelles		
1. above wing, small airplane	Cross section area	.250
2. large leading edge nacelle, small airplane	Cross section area	.120
3. small leading edge nacelle, large airplane	Cross section area	.080
4. Improved nacelle, no cooling flow	Cross section area	.050
5. Improved nacelle, typical cooling air flow	Cross section area	.100
Wing Tanks		
1. centered on tip	Cross section area	.05-.07
2. below wing tip	Cross section area	.07-.10
3. Inboard below wing	Cross section area	.15-.30
Wires and Struts		
1. smooth round wires and struts (per foot)	Frontal area	1.2-1.3
2. standard aircraft cable (per foot)	Frontal area	1.4-1.7
3. smooth elliptical wire (per foot)	Frontal area	
fineness ratio 2:1		0.6-0.4
fineness ratio 4:1		.35
fineness ratio 8:1		.3-.2
4. standard streamlined wire (per foot)	Frontal area	.45-.20
5. square wire (per foot)	Frontal area	.16-.20
6. streamlined struts (per foot)	Frontal area	.075-0.10

For smooth round wire of diameter less than ¼ inch, assume two end fittings equivalent to three feet of wire.

For smooth round struts of diamter greater than 5/16 inch, assume two end fittings equivalent to one foot of strut.

For smooth elliptical wire, assume two end fittings equivalent to 10 to 15 feet of wire.

For square wire, assume two end fittings equivalent to two feet of wire.

For streamlined struts, assume two end fittings equivalent to five feet of strut if faired, ten feet if unfaired.

The contribution of the various protuberances to the total aircraft drag coefficient is commonly on the order of 3% to 5%.

PROGRAM FOR THE CALCULATION OF TWO-DIMENSIONAL

AERODYNAMIC COEFFICIENTS

INTRODUCTION

In order to predict the three-dimensional lift and drag characteristics of a general planform, one usually begins with the two-dimensional characteristics of the airfoils which constitute the wing. As noted in the literature review, a computer program which estimates the aerodynamic characteristics of multiple-component airfoils in subsonic, viscous flow already exists (Ref. 31). This program, originally written for NASA Langley under NASA Contract NAS1-9143, has undergone extensive in-house modification at NASA Langley; this modified version is available to the general public. To take advantage of the rather substantial effort which went into the preparation of this program, it was decided to use this NASA program as the basis for developing a suitable—in terms of accuracy and computational requirements—procedure for estimating the two-dimensional lift and drag characteristics of general airfoils.

Since the present report is concerned only with the prediction of characteristics of single-element airfoils and wings, a single element version of the program was obtained from NASA Langley. Although the NASA program was written for the CDC 6600 computer, only slight modification was required in order to use it on the IBM 370-165 computer at N. C. State University. Several airfoils were then investigated at NCSU to determine how well the program prediction of lift, drag, and pitching moment coefficients compared with the experimental data given in Reference 19. After several comparisons it was concluded that in general the predicted lift coefficient was high by 5 to 8 percent for moderately thick airfoils and 8 to 15 percent for very thick airfoils; the drag coefficient was usually high by at least 25% and sometimes as much as 75%. The program as obtained from Langley required 200K (K denotes 1,000 bytes) of core storage and a scratch disk (for matrix inversion use) for an IBM FORTRAN IV H-LEVEL run on the IBM 370-165 computer. The average execution time was 45 to 60 seconds for each airfoil angle of attack (a large portion of this time is input-output time for the scratch disk). Based on the above results the goals for the new program were set as follows:

(1) modify the NASA program to improve the predicted lift and drag coefficients

(2) reduce the size of the program to facilitate its use on smaller computers (∿100K)

(3) reduce the computational time required to evaluate the coefficients of a particular airfoil.

The extent to which the above goals were achieved is summarized below:

(1) a marked improvement in the accuracy of the predicted lift and drag coefficients is evident by perusing Figure 12 through Figure 30

(2) the modified program required only 106K of core storage with no scratch disk required

(3) the computational time required to evaluate the coefficients of
a particular airfoil was reduced to 20 seconds or less.

In the following sections the reader will find a brief explanation of
the theory upon which the program is based, a discussion of the various
changes and modifications implemented in the NASA program, and a comparison
of the predicted lift and drag coefficients (from both the original NASA
program and the NCSU-modified program) with experimentally-determined
coefficients.

GENERAL PROGRAM THEORY

The only practical method currently available for predicting the viscous flow field about an airfoil involves an iterative procedure. The basic steps for this iterative procedure are:

(1) obtain an inviscid flow solution for the basic airfoil

(2) obtain a boundary layer solution based on the inviscid flow solution

(3) construct a modified airfoil by adding the boundary layer displacement thickness to the original airfoil

(4) obtain an inviscid flow solution for the modified airfoil

(5) repeat steps (2) through (4) until some convergence criterion is satisfied, for example, the difference between two successive values of lift coefficient is less than some specified tolerance.

The above steps represent only an outline of the procedure to be used since there are many techniques available for obtaining inviscid and boundary layer flow solutions. For a discussion of many of the available techniques and how they are applied to particular problems, the reader should consult the literature review.

A detailed description of a logical solution procedure is given in the general theory of the previous section. However, for reasons noted in the general introduction, this procedure differs in some details from the method actually employed in the program. Thus, to insure clarity, a brief description of the actual solution procedures used in the modified NCSU program is given below.

It is of utmost importance to choose a solution procedure which can be relied on to converge in most cases. For the problem of the flow field about an airfoil, convergence depends primarily on the manner in which the displacement effects of the boundary layer are treated in the inviscid flow calculations. The program as supplied to NASA chose to model the influence of the boundary layer on the velocity distribution over the actual airfoil as two separate effects. The effect of the boundary layer on the lift is considered to be a modification of the camber line so as to effectively decrease the angle of attack of the airfoil. The change in camber is given by the difference in the magnitude of the upper and lower surface displacement thicknesses as shown in Figure 1.

156

original airfoil
with camber line

upper and lower surface δ^*
with change in camber
line due to δ^*

Δ camber $= \delta_u^* - \left|\delta_L^*\right|$

modified airfoil with
modified camber line
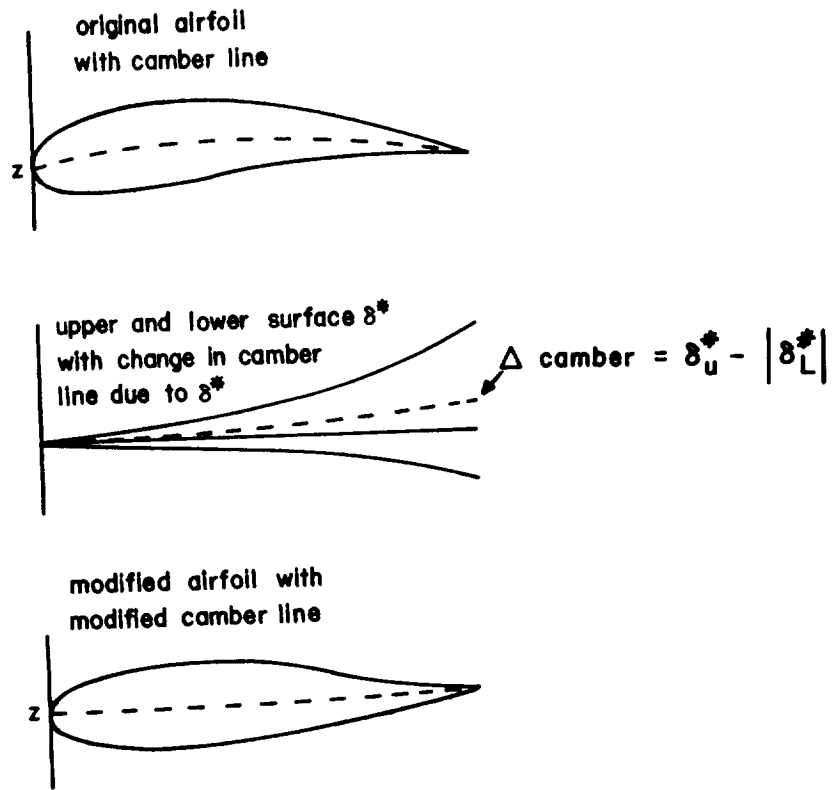
Figure 1.   Modification of camber line due to boundary
layer displacement thickness.

Thickness effects due to the existence of the boundary layer tend to relieve the stagnation condition in the trailing-edge region, giving rise to the pressure or form drag of the airfoil. This effect of the boundary layer on the drag was approximated by the difference of two "thickness" solutions (see Figure 2). The first thickness solution is for a symmetric airfoil at zero angle of attack with the same thickness distribution as the original airfoil plus boundary layer displacement thicknesses. The second thickness solution is also for a symmetric airfoil at zero angle of attack but with the same thickness distribution as the original airfoil. Since superposition is assumed to apply, the velocity distribution over the original airfoil is given by,

$$\left(\frac{V}{V_\infty}\right)_{TOTAL} = \left(\frac{V}{V_\infty}\right)_{CAMBER} + \left(\frac{V}{V_\infty}\right)_{BT+\delta^*} - \left(\frac{V}{V_\infty}\right)_{BT} \quad ,$$

where BT stands for the basic thickness distribution of the original airfoil.

157

original airfoil

symmetric airfoil with
same thickness distribution

symmetric airfoil with
displacement thickness added
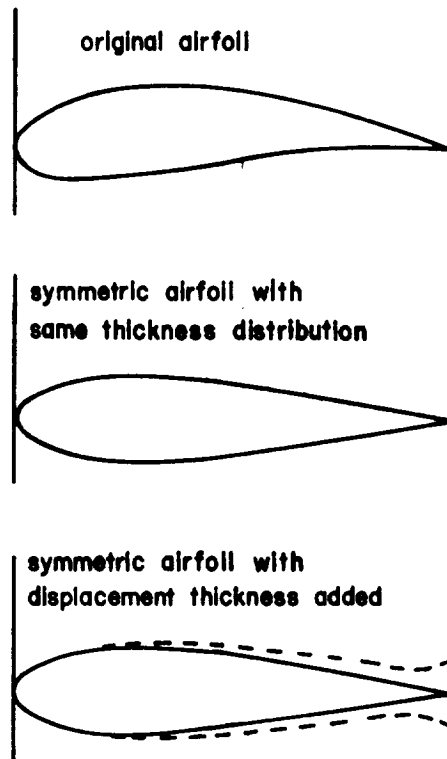
Figure 2.   Thickness effects due to boundary layer
            displacement thickness.

During the initial phase of the investigation, the authors questioned
the necessity of having to use superposition to model the various boundary
layer displacement effects. It seemed that the boundary layer displacement
effect could be modeled by simply adding the displacement thickness to the
original airfoil. However, upon consultation with Mr. Harry L. Morgan, Jr.,
the authors were informed this approach had already been attempted at NASA
Langley but was unsuccessful. Time simply did not permit the authors to
investigate this possible solution procedure further.

The program as originally supplied to NASA Langley used a distributed
vorticity method to calculate inviscid flow solutions. The airfoil was
approximated by a closed polygon, and the distributed vorticity was assumed
to vary linearly along each line segment of this polygon. The sum of the
velocity induced by the distributed vorticity and the free stream velocity
was forced to satisfy the condition that its component normal to the
airfoil surface must be zero at the midpoint of each line segment. NASA
replaced this inviscid solution procedure with a different distributed
vorticity method based on Oeller's work (Ref. 26 and 27).* Using the new

---

*   For the user's convenience, a derivation of the equations for Oeller's
method is given in Appendix G.

158

procedure the airfoil is again approximated by a closed polygon, but the
distributed vorticity is assumed to have a constant value per line segment.
The solution is obtained by requiring that the stream function have the
same constant value at each of the midpoints of the line segments (see
page 405). Consider the two line segments shown in Figure 3. To obtain a
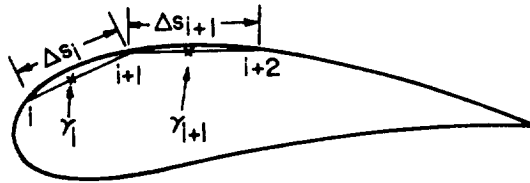
Figure 3. Schematic of control point averaging.

value for the local velocity at the point (i + 1) on the airfoil surface,
it is necessary to use some type of averaging technique on the distributed
vorticities γ_i and γ_{i+1}. The NASA program used an averaging procedure
of the form,

$$\left(\frac{v}{V_\infty}\right)_{i+1} = \tfrac{1}{2} (\gamma_i + \gamma_{i+1}) \quad .$$

Strictly speaking this procedure is correct only if all the line segments
have the same arc length. A somewhat better averaging procedure is obtained
by using the following form,

$$\left(\frac{v}{V_\infty}\right)_{i+1} = (\gamma_i \Delta s_i + \gamma_{i+1} \Delta s_{i+1})/(\Delta s_i + \Delta s_{i+1}) \quad .$$

This improved averaging technique was therefore incorporated into the
NCSU-modified program.

To facilitate the understanding of the remainder of the programming
theory, a program flow chart has been included and is presented in Figure 4.
The remainder of this section will be concerned with the explanation of
this flow chart.

As indicated in the flow chart, the program begins by calling subroutine
READIT. As its name implies READIT is responsible for reading the input
geometry and the ambient conditions. A call is then made to subroutine GEOM
to obtain a set of 65 distributed solution points. A basic thickness
velocity distribution $\left(\frac{v}{V_\infty}\right)_{BT}$ is calculated next using subroutine VOVBT.
This velocity distribution is calculated only once for each airfoil because
it is the basic thickness solution at zero angle of attack. Various arrays
are then initialized and MAIN2 which controls the calculation of the inviscid
flow solution is entered. The camber solution velocity distribution $\left(\frac{v}{V_\infty}\right)_{CAMBER}$

159

is calculated in subroutine CAMBER. For all iterations after the first (zeroth in the program output), the "basic thickness plus displacement thickness" velocity distribution $\left(\dfrac{V}{V_\infty}\right)_{BT+\delta^*}$ is calculated in subroutine VOVBT. The three velocity distributions are combined to give the total velocity distribution for the airfoil, $\left(\dfrac{V}{V_\infty}\right)_{TOTAL}$. Using this velocity distribution subroutine COMPR is called to obtain the compressible pressure coefficients using the Karman-Tsien correction law. The location of the forward stagnation point is obtained from subroutine STAG and subroutine MAIN2 returns control to the mainline.

Subroutine MAIN3 is next entered to control the boundary layer calculations. A call is made to subroutine LAMNA2 to obtain the laminar boundary layer solution for the upper surface. At each point on the airfoil surface subroutine LAMNA2 makes a call to subroutine BLTRAN to see if boundary layer transition has occurred. Control remains in subroutine LAMNA2 until either laminar separation, boundary layer transition, or the end of the surface is encountered. If boundary layer transition has occurred, TURB2 is called to calculate the turbulent boundary layer solution. To insure stability in the computation the last seven boundary layer displacement thicknesses are replaced using a least squares fit obtained from subroutine LSQ*. A calculation for the lower surface boundary layer is performed in a similar manner. The pressure coefficient and skin friction coefficients are integrated over the airfoil in subroutine LOAD, and the various aerodynamic and force coefficients are computed in MAIN3 before control is returned to the mainline. A check is then made to see if five iterations have been completed, a check is made to see if all the angles of attack have been used for the specified Mach number, and a check is made to see if the airfoil has been investigated at each of the specified Mach numbers. Once all the Mach numbers have been investigated, the program returns to subroutine READIT where it attempts to read another data set.

Although this description of the flow chart concludes the General Program Theory section, the reader is reminded that additional aspects of the theory must, of necessity, be discussed in explaining many of the modifications incorporated into the NCSU program. The next section will discuss program modifications of a general nature while later sections will be concerned with modifications of a more specific nature.

---

* See page 79 of Reference 31 for the description of this displacement thickness smoothing procedure.
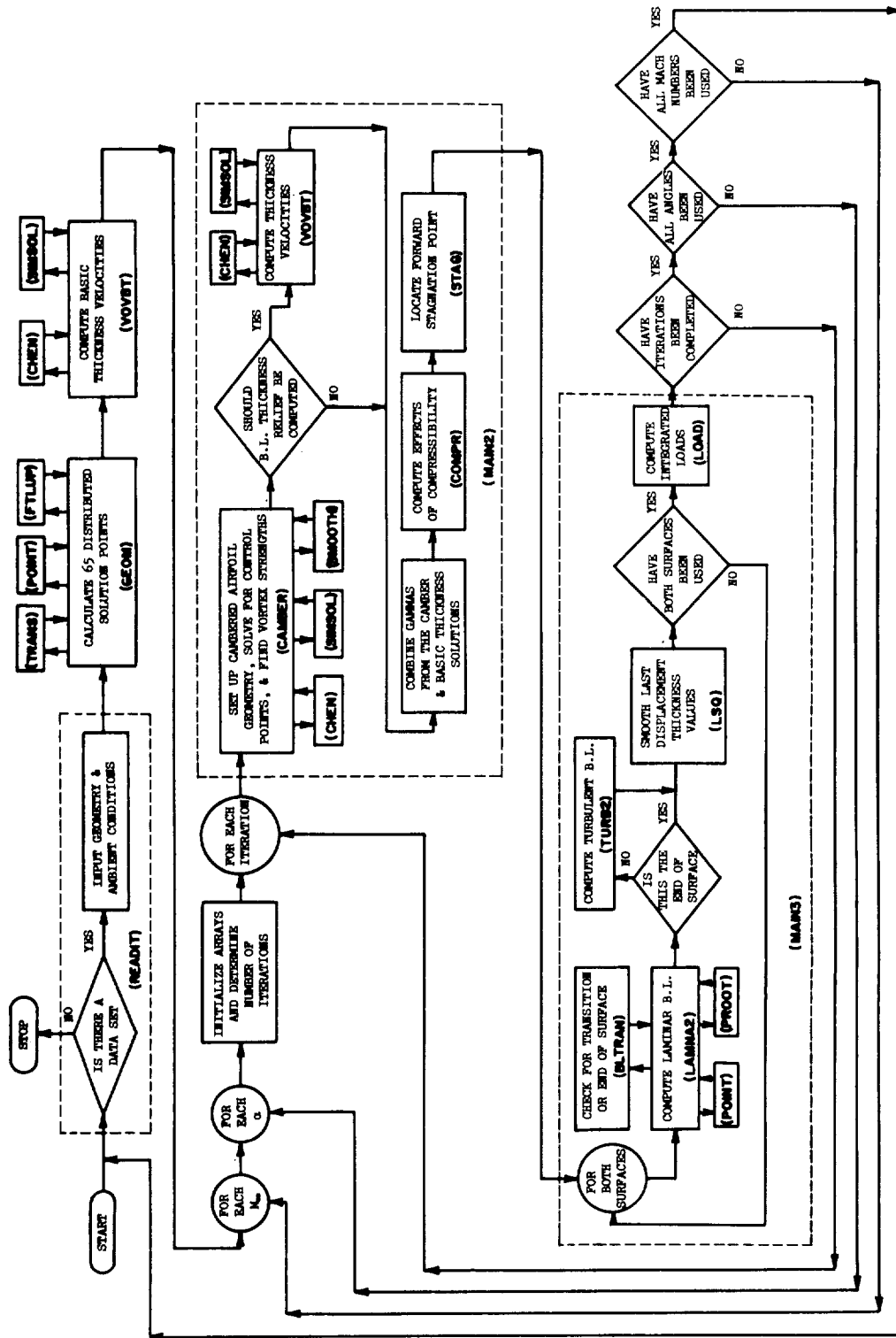
Figure 4. NCSU computer program flow chart.

161

GENERAL PROGRAM MODIFICATIONS

Many of the changes made in the NASA program may be classified as general programming modifications applicable to the entire program rather than a particular subroutine; these general modifications are discussed below. Throughout the remainder of the text the program obtained from NASA Langley will be referred to as the NASA program while the modified program generated at N. C. State University will be referred to as the NCSU program.

The NASA program was run on the CDC 6600 computer using segmentation to reduce the core requirement. Since segmentation as such is not supported by IBM machines, the segmentation control cards were deleted. The NASA program also contained a PLOT subroutine which was used to plot the pressure coefficients. Inasmuch as plotting software may vary drastically from installation to installation, this subroutine was deleted from the program.

Although only a multi-component version of the program was originally supplied to NASA Langley (Ref. 31), a single element version was produced at NASA by deleting the confluent boundary layer and slot flow subroutines. This single element version was the program made available to N. C. State University. The program as supplied still contained variables in other subroutines which were used only for multi-component calculations and variables which were double subscripted to facilitate multi-component calculations (the second subscript was used to denote the particular component). These unnecessary variables and second subscripts are eliminated in the NCSU version of the program.

In the interest of clean coding, the COMMON statements were aligned and modified so that the Common variables have the same variable name in each of the subroutines in which they appear. Also, whenever possible the NCSU program makes use of Common transfer of information thus eliminating many of the subroutine arguments.

In the NASA program a Mach number extrapolation and smoothing process* was applied to upper and lower surface trailing edge pressure coefficients in subroutine MAIN3 (statements 90 through 99 of NASA program). Since this pressure coefficient modification was found to produce a significant non-zero lift coefficient for a symmetrical airfoil at zero angle of attack, it was deleted in the NCSU program.

The NASA program also allowed the user to specify the total number of solution points desired for the airfoil investigated, up to 100 points. In order to minimize storage, several test cases were investigated to find a minimum value for the number of solution points which would still give valid aerodynamic coefficients. Calculations with 65 points produced

---

* See pages 78 and 79 of Reference 31 for the description of this procedure.

coefficients which were within one or two percent of those predicted
using 100 points; also, less than 65 points showed the trend of producing
a larger percentage difference as the number of points were decreased.
It was therefore decided that 65 solution points were the optimum number,
and the NCSU program specifies the use of 65 solution points.

Several of the unnecessary WRITE and PRINT statements contained in the
NASA program were deleted in the NCSU program to reduce total program
size. To insure compatibility with other installations all input/output
statements were changed to the forms READ(JREAD,C), WRITE(JWRITE,C), and
WRITE(JPUNCH,C) where JREAD, JWRITE, and JPUNCH are the appropriate
input/output numbers and C is some FORMAT statement number. The values
of the input/output unit numbers are assigned by three specification
statements at the beginning of the mainline of the NCSU program (JREAD=1,
JWRITE=3, and JPUNCH=2). For installations having different input/output
unit numbers these specifications may be changed. Only Hollerith field
literal data specifications were permitted in the NCSU program to insure
IBM-CDC machine compatibility.

The maximum number of angles of attack, specified as 5 in the NASA
program, was increased to 10 in the NCSU program. Also, many of the
subroutines contained in the NASA program were combined or deleted in
the NCSU program to make it simpler and more efficient. A table comparing
the subroutines contained in each of the programs is given below.

NASA PROGRAM	NCSU PROGRAM
MAIN	MAIN
READIT	READIT
GEOM	
DISTP	GEOM
ROTRAN	
MAIN2	MAIN2
VOVBT	VOVBT
THICK	
COEFF	
VORTEX	
CONTPT	CAMBER
EQUIV	
CHEN	CHEN
COMPR	COMPR
STAG	STAG
MAIN3	MAIN3
LAMNA2	LAMNA2
BLTRAN	BLTRAN
TURB2	TURB2
TURB	
LOAD	LOAD
LSQ	LSQ
TRANS	TRANS
PROOT	PROOT
POINT	POINT

163

```
NASA PROGRAM                NCSU PROGRAM

      SLOPE
      SMOOTH               SMOOTH
      FTLUP                FTLUP
      DIF
      SIMSOL               SIMSOL
      FUNCTION LOCF(X)
```

Table 1. Comparison of subroutines contained in the NASA
program and the NCSU program.

In order to give the user the option of reducing the size of the NASA
program output for a particular airfoil, an IWRITE control parameter was
incorporated into the NCSU program. IWRITE is a program input variable
which should have the value 0, 1, 2, or 3. IWRITE = 0 is the default
write option which basically gives the complete output presented in the
NASA program. An IWRITE of 1, 2, or 3 will yield reduced portions of the
total output with IWRITE = 3 producing the minimum output for each airfoil
tested. An example of the output generated for each of the IWRITE options
is given in Figures A-3, A-4, A-5, and A-6 of Appendix A.

The angle of attack of an airfoil is generally defined as the angle
between the free stream flow direction and the airfoil chord line. The
program must therefore know the location of the chord line of the airfoil
with respect to the x-axis of the reference system in order to calculate
the correct aerodynamic coefficients as a function of angle of attack.
In most cases the location of the chord line is known, and the airfoil
coordinates are referenced with respect to this chord line. However, in
some instances the chord line location may not be precisely defined, thus
requiring the program to calculate a chord line to use as a reference
line for the angle of attack. In general the most logical choice for
the calculated chord line would be the longest line from the midpoint of
the trailing edge to the nose of the airfoil. Thus, the program would
choose line A as the chord line for the airfoil depicted in Figure 5.
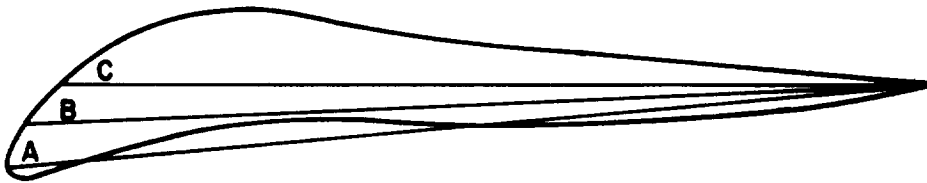
Figure 5. Sample airfoil with longest chord line.

Throughout this report a chord line calculated by the program in this manner
will be referred to as the longest chord line. An input control parameter
IALPHA was incorporated into the NCSU program to allow the user to either
specify the airfoil chord line or to allow the program to calculate the
longest chord line.

164

IALPHA = 0 implies that the chord line of the airfoil is parallel to the x-axis of the input reference system of specified airfoil data points. Although it would be more convenient to specify the airfoil coordinates with the chord line as the x-axis of the reference system as shown in Figure 6a, it is only necessary to input the coordinates with the chord line parallel to the x-axis of the reference system as shown in Figure 6b.

(a)   **(b)**
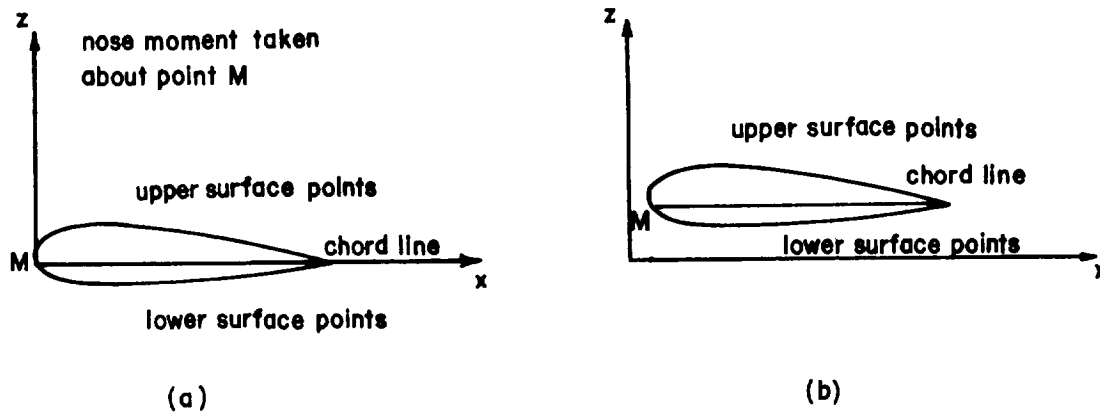
Figure 6.   Location of chord line for IALPHA = 0.

The authors strongly urge that only the convention in Figure 6a be used. As indicated in the User Instructions of Appendix A, a set of lower surface points and a set of upper surface points are read for the airfoil. The upper and lower surface points are defined as those points above and below the specified chord line respectively as indicated in Figure 6. It should also be noted that when calculating the moment coefficient about the nose, the nose point is taken to be the point where the chord line cuts the leading edge of the airfoil. It should be reemphasized that the above discussion applies only to the IALPHA = 0 option.

IALPHA = 1 implies that the chord line of the airfoil is the longest chord line calculated by the program. For this particular option the user may choose any convenient reference line for dividing the airfoil into upper and lower surface input points. The only restriction placed on this reference line is that it must go through the trailing edge midpoint. Again however, the authors strongly recommend that this line be the x-axis of the input reference system as shown in Figure 7. After calculating the longest chord line, the program will then rotate and translate the airfoil coordinates so that the longest chord line lies along the x-axis with the nose of the airfoil at the (0,0) point (see Figure 8). The lift and drag coefficients are then calculated with the angle of attack referenced to this longest chord line, while the moment coefficient about the nose is calculated about the (0,0) point. Similarly the moment coefficient about the quarter-chord is calculated using the point (.25C,0), where C is the length of the longest chord line.
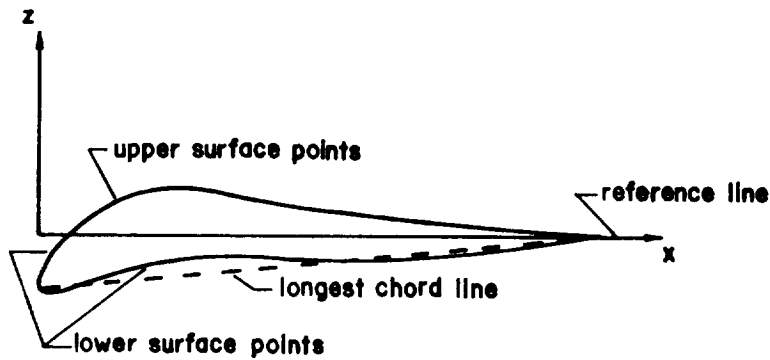
Figure 7.   Example airfoil in reference system
            for IALPHA = 1 option.
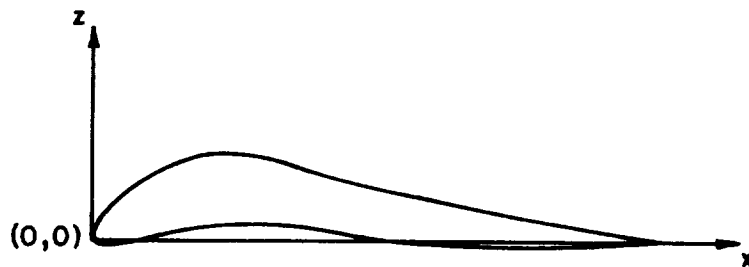
Figure 8.   Rotated airfoil after longest chord line
            has been calculated.

In order to make the NCSU airfoil program compatable with the companion program (given in Appendix C) for calculating three-dimensional aerodynamic coefficients, an input IPUNCH control parameter was added.   The control parameter IPUNCH gives the user the option of obtaining punched data (lift, drag, and quarter-chord moment coefficients for each angle of attack) which may be used in the three-dimensional program.   IPUNCH = 1 gives punched output while the default IPUNCH = 0 gives none.

Many comment cards were added throughout the entire program to help explain program theory and logic.   In general, the purpose of each subroutine is defined at the beginning of the subroutine, and also many of the important areas of the program were denoted using these comment cards.

As a final note to the general program modifications the user is strongly advised to read the User Instructions concerned with input data to the NCSU program since it differs in some respects with that of the NASA program.   For example, the input variable RN in the NCSU program denotes the value of the Reynolds number in millions.   It does not denote Reynolds number in millions per foot as it did in the NASA program!!!

166

While many of the revisions of the NASA program could only be classified as general modifications, there were also modifications which were more specific in nature. The revisions which are concerned with airfoil geometry are discussed in the following section.

CHANGES CONCERNING THE AIRFOIL GEOMETRY SPECIFICATION

In order to compute the lift and drag characteristics of any airfoil, the airfoil coordinates must be specified. The NASA program was designed so that 200 points could be used to specify the shape of the airfoil surface. Since most airfoils can be specified adequately using less than 100 points, the size of the NCSU program was reduced by requiring that the airfoil be specified by 100 points or less. These specified points must be distributed about the airfoil in a manner such that if the airfoil were rotated to the longest chord line system there would be no more than 65 upper or lower surface points (where the upper and lower surface points are defined with respect to the longest chord line). The problem of having more than 65 upper or lower surface points after rotation can usually be avoided if the user specifies no more than 50 input points on either surface. However, if more than 65 points are encountered an appropriate error message will be printed in the program output, and execution of that particular airfoil will be terminated.

As noted under general program modifications the aerodynamic coefficients are calculated based on 65 airfoil solution points. However, these 65 solution points are <u>not</u> specified airfoil input points as such; they are a set of points generated <u>from</u> the specified points by a distribution procedure in subroutine GEOM. Only one call is made to subroutine GEOM since the airfoil can be investigated for several angles of attack and/or Mach numbers using the same set of distributed points. The distribution procedure requires many intermediate arrays for computational purposes, but only two arrays containing the coordinates of the distributed airfoil points are required for the remainder of the program computations. The intermediate arrays are therefore equivalenced to other arrays used in later program computations. This equivalencing procedure gives a reduction of the number of large arrays required by the program.

Subroutine GEOM in the NCSU program is a combination of three of the subroutines in the NASA program (subroutines ROTRAN, GEOM, and DISTP). This subroutine first calculates the longest chord line of the airfoil regardless of the IALPHA option used. It then calculates the angle β between this longest chord line and the x-axis of the input reference system (see Figure 9) and rotates and translates the specified input points so that the longest chord line lies on the x-axis with the nose at the (0,0) point. This is done so that a new set of upper and lower surface points defined with respect to the longest chord line will have monotonic increasing values of x. It was decided that the NASA program could be greatly simplified if the x-values of the distributed upper surface points and the x-values of the distributed lower surface points were the same along the longest chord line. Based on this decision the NCSU program was modified so that it would produce 32 upper and 32 lower surface distributed points along with a common leading edge point (*i.e.* the (0,0) point), giving a total of 65 distributed points.

168
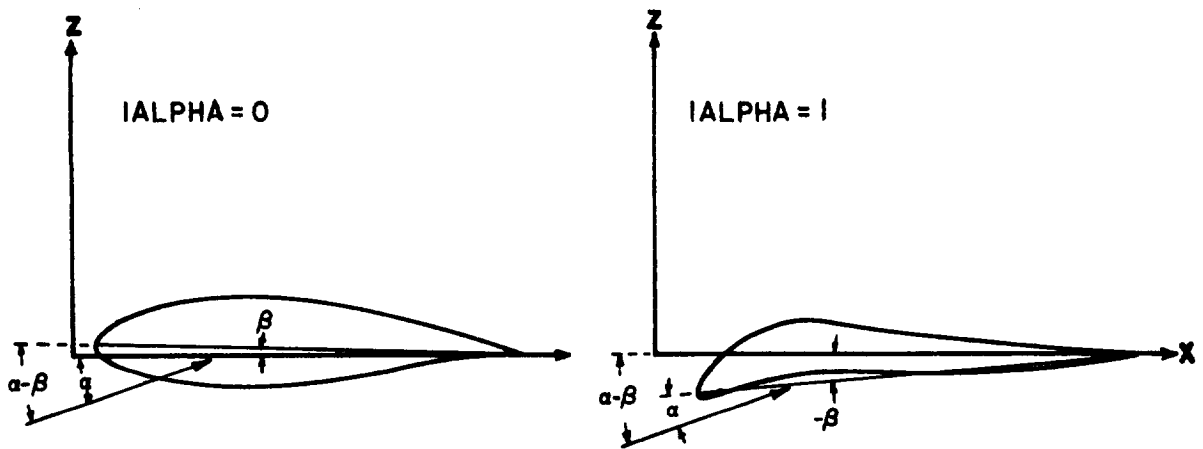
Figure 9.   Reference line for the angle of attack
for both IALPHA options.

The distribution of the points Is made a function of local surface
curvature so that more points are distributed In regions of high curvature.
The distribution procedure Is as follows:

(1) the program uses the upper surface x-values as a longest chord
line scanning array starting with the leading edge point

(2) the curvatures* of both the upper and lower surfaces are computed
at each of these x-values

(3) the absolute value of the curvature for both surfaces are
compared at each x station, and the maximum curvature Is kept

(4) the Integral $SUMA_i = \int_0^{s_i} |K|^{0.25}\, ds$ Is evaluated using the

maximum curvatures at the x stations and the corresponding surface
arc lengths s_i along the upper surface (note that a value of this
Integral Is stored for each s_i station to facilitate backward
Interpolation)

---

* In the nose region the curvature K Is calculated from

$$K = [cz^2 - (cx + b/2)^2]/[z^2 + (cx + b/2)^2]^{3/2}$$

where c and b are taken from a curve fit of the airfoil points of the form

$$z^2 = a + bx + cx^2 \quad .$$

Note that this curve fit gives a slope $dz/dx = (cx + b/2)/z$ which becomes
Infinite when z becomes zero. Because the above curve fit gives an Infinite
slope when z = 0, It obviously cannot be used near the trailing edge of the
airfoil. Therefore, In the trailing edge region the curvature Is calculated
from $K = 2c/[1 + (2cx + b)^2]^{3/2}$, where c and b are taken from a curve fit of
the form $z = a + bx + cx^2$.

169

(5) the maximum value of the integral is divided into 32 equal portions

(6) an s value corresponding to each of these increments is found by backwards interpolation between the s_i array and $SUMA_i$ array

(7) similarly an x value corresponding to the s value is found by backward interpolation

(8) finally corresponding z values for both the upper and lower surfaces are found by backward interpolations.

Although the curve fit formula used in the nose region is designed to approximate an infinite slope, a better value for the curvature integral is achieved when there are more points specified in the nose region of the input data. The better value of the curvature integral will lead to a better set of distributed x's.

In the NASA program the distributed points are rotated and translated back to the reference axis system. For the NCSU program the distributed points remain in the longest chord line system in order to take advantage of the same upper and lower surface x values. For the IALPHA = 1 option the longest chord line system is the ideal system in which to calculate the aerodynamic coefficients. For the IALPHA = 0 option the program calculations are performed using an angle of attack with respect to the longest chord line of $(\alpha - \beta)$ since this corresponds to an angle α with respect to the actual airfoil chord line (refer back to Figure 9).

Following this discussion of the specification of the airfoil geometry, the next two sections present the revisions made to the camber, thickness, and boundary layer solutions of the NASA program.

170

THICKNESS AND CAMBER SOLUTION MODIFICATIONS

It was pointed out in the section on general program theory that the solution for the complete airfoil is a sum of a thickness solution and a camber solution. The modifications which were made in both the thickness and camber solution portions of the NASA program will now be considered.

Subroutine VOVBT is designed to calculate the γ's or surface velocities for a symmetric airfoil with the same thickness distribution as the original airfoil plus boundary layer displacement thickness. Remembering that the upper and lower surface x-arrays are the same in the longest chord line system, the upper surface ordinates of the symmetric airfoil are generated from the following equation,

$$(z_{sym.})_{up} = \tfrac{1}{2}((z_{air})_{up} - (z_{air})_{low}) + \tfrac{1}{2}((\delta^*)_{up} + (\delta^*)_{low}) \ ,$$

for $0 \leq x \leq C$,

where:

z_{air} = original airfoil ordinate
δ^* = boundary layer displacement thickness
C = length of longest chord line.

It also follows that $(z_{sym})_{lower} = -(z_{sym})_{upper}$. Now, once the boundary layer displacement thicknesses have been added, the symmetric airfoil will have a finite trailing edge thickness which is denoted by z_{te}. In the actual physical flow, the wake acts as a displacement body in which the thickness decreases very rapidly from z_{te} to a finite value $z_\infty = \tfrac{1}{2}C_D C$ according to Reference 30, where C_D is the airfoil drag coefficient based on the chord length C. In subroutine VOVBT the wake displacement body of the symmetric airfoil is modeled by:

$$(z_{sym})_{up} = \tfrac{1}{2}[(z_{te} - z_\infty) \ e^{(-6.9XX)} + z_\infty](1.0 - XX)$$

where

$$XX = x/C - 1.0 \quad \text{for} \quad C \leq x \leq 2C.$$

Based on actual measurements given in References 72 and 73 the present authors found that the extrapolation function inside the brackets in the equation above closely approximated the boundary layer displacement thickness of the wake in the trailing edge region. The multiplicative term inside the parentheses is included to close the wake displacement body at x = 2C since the inviscid procedure used for the solution gives best results when applied to closed bodies. The authors feel that this approximation to the wake displacement body gives a better representation of the actual physical flow than does the wake extrapolation procedure in the NASA program while also reducing program size and computation time.

171

Since the thickness solution is calculated only at a zero angle of attack, the flow field about the symmetric airfoil gives an upper surface γ distribution which is simply the negative of the lower surface γ distribution, $i.e.$ $(\gamma_{up})_{x_i} = -(\gamma_{low})_{x_i}$, where x_i denotes a particular x station of the airfoil along the longest chord line. Thus, only the γ's for the upper surface of the symmetric airfoil need to be found.

Subroutine CAMBER is designed to calculate the γ's or surface velocities for an "equivalent" airfoil which has the same thickness distribution as the original airfoil but a modified camber line. This modification is due to the effect of boundary layer displacement thickness on the original airfoil as noted in general program theory. The upper and lower surface ordinates of the "equivalent" airfoil are given by the following equations,

$$(z_{eq})_{up} = (z_{air})_{up} + \tfrac{1}{2}\,\Delta z$$

$$(z_{eq})_{low} = (z_{air})_{low} + \tfrac{1}{2}\,\Delta z$$

where $\qquad \Delta z = (\delta^*)_{up} - (\delta^*)_{low}$.

172

BOUNDARY LAYER MODIFICATIONS

The NASA program contains four subroutines for boundary layer calculations: LAMNA2, BLTRAN, TURB2, and TURB. LAMNA2 performs the laminar boundary layer calculations on the airfoil surfaces. At each point along the airfoil surface, LAMNA2 calls BLTRAN to determine if either transition from laminar to turbulent boundary layer flow or laminar separation has occurred. If transition has occurred, TURB2 calculates the turbulent boundary layer solution over the remainder of the airfoil surface. After the last iteration between the potential and boundary layer flow calculations is performed, an additional turbulent boundary layer calculation is performed using subroutine TURB. This subroutine performs a refined turbulent boundary layer computation to predict the point of turbulent separation.

A brief description of the methods used in LAMNA2 is given in Reference 31, and they are derived in detail in Reference 32. The method for predicting transition used in BLTRAN is described briefly in Reference 31 and derived in detail in Reference 32, while the method for predicting laminar stall is derived in Reference 31. TURB2 uses Goradia's Turbulent Boundary Layer Method which is derived in Reference 31. Goradia's method is designed to remain stable under the influence of extreme gradients, both favorable and adverse, and provide reasonable momentum and displacement thicknesses downstream of the turbulent separation point. TURB uses Nash's Turbulent Boundary Layer Method which is described briefly in Reference 31. Nash's method provides a prediction of the point of turbulent separation, but his method is too sensitive to adverse gradients to be used in the initial iterations.

The NCSU program retains subroutines LAMNA2, BLTRAN, and TURB2. Subroutine TURB (Nash's method) was removed to reduce program size since it played no active part in the iterative process between the potential and boundary layer flow solutions. However, the user should be able to return it to the NCSU program if he so desires.

The three boundary layer subroutines retained in the NCSU program are virtually identical to the original routines in the NASA program except for some coding clean-up performed mainly in the COMMON statements. However, the authors have made two important changes in subroutine LAMNA2:

(1) The NASA program input parameter RN specified the Reynolds number in millions per foot. In the NCSU program, the input parameter RN now specifies the Reynolds number in millions. Since subroutine LAMNA2 requires a value for the Reynolds number in millions per foot, the authors have added a variable RNPFT (equal to RN/CREF) to LAMNA2. This variable is calculated in statement LAM 61 and is used only in statement LAM 62 (see the listing given in Appendix A).

173

(2) The Reynolds number is modified to improve the comparison between calculated and experimental drag coefficients. The modification is justified in the following section of this report (Coefficient Modification). It consists of calculating the boundary layer flow using a Reynolds number which is twice that of the specified Reynolds number. Now the Reynolds number is used only once in LAMNA2 to calculate the kinematic viscosity VO which is then used in all three of the boundary layer subroutines. Increasing the Reynolds number by a factor of two is equivalent to dividing the kinematic viscosity by a factor of two. Therefore, the Reynolds number modification procedure is carried out in statement LAM 65 by dividing the kinematic viscosity by a factor of two.

174

COEFFICIENT MODIFICATION

The various modifications discussed up to this point have resulted in a reduction in both program size and run time. While the authors had hoped that some of the modifications would significantly improve the predicted aerodynamic coefficients, comparison with test results indicated that only slight improvement had been obtained. Since attempts at improving the actual program theory had failed the authors sought to determine whether semi-empirical and empirical corrections might improve the results. The corrections which were incorporated into the program are discussed below.

During the course of the investigation the authors noticed that integration of the pressure coefficients from the inviscid solution gave a non-zero pressure drag. This result was obviously contrary to potential flow theory which says there are no drag forces in inviscid flow. It was first believed that this non-zero pressure drag was the result of round-off and loss of significance in the program computations. Calculations were thus made in both single and double precision arithmetic with various integration procedures to test this hypothesis. The results of these tests demonstrated that the problem was not one of loss of significance in program computations but rather a result of the boundary condition applied at the trailing edge of the airfoil. In true potential flow the Kutta condition requires that a stagnation point be recovered at the trailing edge thus yielding a zero pressure drag. However, the boundary condition employed at the trailing edge in the program's potential flow calculations (subroutines VOVBT and CAMBER) is based on Howarth's criterion that the total flux of vorticity shed into the wake from the upper and lower airfoil surfaces must be zero (References 74 and 75). This condition sometimes referred to as a modified Kutta condition is satisfied by requiring that the upper and lower surface velocities at the trailing edge be equal and thus not necessarily zero as would be the case with a stagnation point at the trailing edge. As a result, a potential flow calculation using this modified Kutta condition yields a velocity distribution which has a non-zero pressure drag.

As discussed in General Program Theory, the superposition technique used in this program implies that the lift forces are represented by the camber solution while the pressure drag due to boundary layer displacement effects is represented by the difference of the two thickness solutions. The difference in the thickness solutions should contain all of the pressure drag; however, because the camber solution is calculated using the modified Kutta condition, it also has an inherent pressure drag. As a result, when the three separate velocity distributions are superimposed, the resulting velocity distribution has, in effect, an "extra" pressure drag in it from the camber solution. The obvious "cure" would be to use a camber solution which has a zero pressure drag. This can be accomplished by using the true Kutta condition as the trailing edge boundary condition for the camber solution while still employing the modified Kutta condition in the two thickness solutions. However, when this procedure was attempted the

175

program's boundary layer routines were unable to cope with the resulting steeply-rising pressure distribution and failed. The next most obvious "cure" would be to try to estimate this "extra" drag and subtract it from the final drag. The authors found that the best way to estimate this "extra" drag was to integrate the pressure coefficients of the inviscid camber solution calculated in the zeroth* iteration which contains no boundary layer effects. A modification was therefore incorporated into the NCSU program to calculate the pressure drag of the zeroth iteration and then subtract this pressure drag from the final predicted drag coefficient.

The modified drag coefficients compared much more favorably with experimental data than did the unmodified coefficients, however they were still a little too large. While computing the aerodynamic coefficients for several airfoil sections, the authors noticed that if the coefficients were calculated using a Reynolds number twice as large as the experimental value an even greater improvement in drag coefficient could be achieved. For example, it was found that if the coefficients of the 23012 airfoil were computed at a Reynolds number of 6,000,000 the predicted drag coefficient compared very well with experimental tests at a Reynolds number of 3,000,000. The trend observed with the 23012 airfoil was also evident in the other airfoils which were investigated; thus, it was decided that a modification would be made so that the aerodynamic coefficients would be calculated using a Reynolds number which is twice that of the specified Reynolds number. This procedure can be justified at least qualitatively by the fact that momentum integral boundary layer methods such as that used here give boundary layer thicknesses which are somewhat large when compared with exact results. Using larger Reynolds numbers tend to reduce the boundary layer thickness.

The effect of both of the modifications discussed above is illustrated in the graph on the following page.

Up to this point none of the modifications have affected the lift coefficients to any extent; they still remained in general 3 to 8 percent too high with the larger errors occurring at the larger angles of attack. For some of the thicker airfoils the lift coefficients may differ from experiment by as much as 10 to 12 percent at high angles of attack. One reason for the large lift coefficients resides in the fact that the effect of the wake on the circulation about the airfoil is not included in the theory. In the actual physical situation the presence of the wake tends to reduce the circulation about the airfoil leading to a reduction of its lift coefficient (Ref. 76). In Reference 77 Spence and Beasley developed an expression which gives the reduction of the lift coefficient due to the wake. This expression, incorporated into the NCSU program, is

$$(C_L)_{\text{with wake}} = (1.0 - 0.214 \sqrt{C_D})(C_L)_{\text{without wake}} \; .$$

In the NASA program the lift and drag coefficients were calculated using the normal axial force coefficients obtained by integrating the pressure and skin friction coefficients over the surface of the airfoil.

---

* In the program the initial or first iteration is referred to as the zeroth iteration.
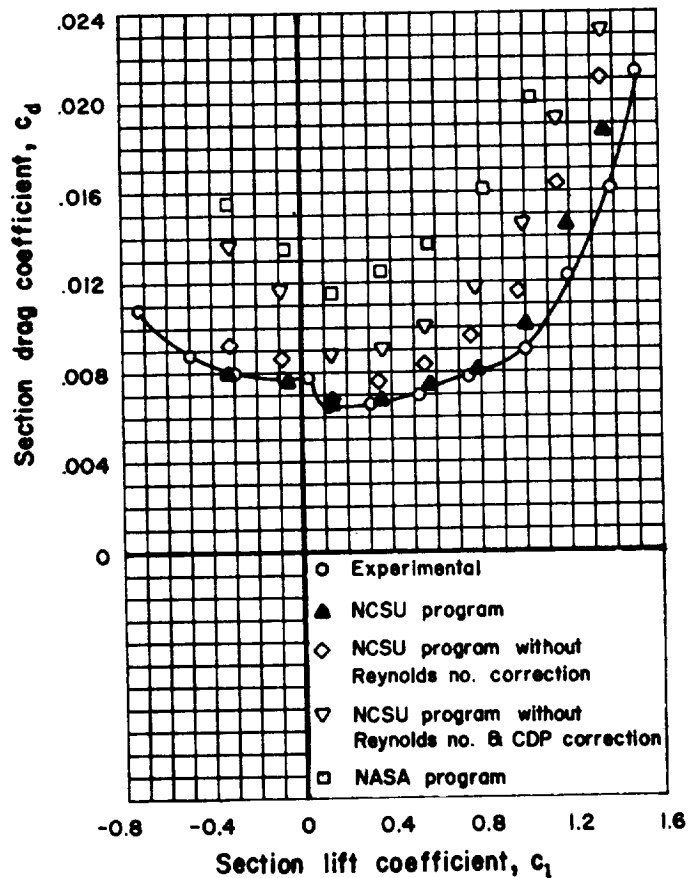
Figure 10. Effect of each of the drag modifications on
the drag coefficients of the 23012 airfoil at
a Reynolds number of 3.0 million.

In the NCSU program the lift and drag coefficients were calculated in the
same manner and then corrected using the modifications discussed above.
In order to make the normal and axial force coefficients consistent with
these new lift and drag coefficients, the force coefficients were re-
computed using the following relations:

$$C_N = C_L \cos \alpha + C_D \sin \alpha$$

$$C_A = - C_L \sin \alpha + C_D \cos \alpha \quad .$$

It is important to note that while the aerodynamic and force
coefficients can be modified to give values which agree better with
experimental data, a similar correction procedure cannot be applied to
modify the pressure coefficients, and therefore they are printed as
calculated.

The results of these program modifications when applied to the
calculation of the aerodynamic characteristics of a number of airfoils
are discussed in the next section.

177

DISCUSSION OF PROGRAM RESULTS

This section is concerned with quantitative comparisons between the predicted coefficients of both the NASA program and the NCSU program and experimental data (taken for the most part from Ref. 19). In order to obtain generally valid comparisons, fifteen airfoils of various thickness and camber distributions were included in the investigation. The fifteen airfoil contours are depicted in Figure 11. The reader can see by examining Figures 12 through 30 that the lift coefficients predicted by each program are quite similar while the drag coefficients are quite different. (Although the plots show only lift and drag coefficient, the reader is reminded that the normal force, axial force, and moment coefficients for any airfoil investigated may also be obtained from the program if desired.) For most of the airfoils the comparisons are self-explanatory and require little discussion; however, some of the more interesting results warrant individual identification.

In general, the lift and drag coefficients obtained using the NCSU program compare very well with experimental data at the lower angles of attack for airfoils with thickness ratios of less than 18 percent. The drag coefficients for the lower angles of attack are slightly high and therefore conservative. For the higher angles of attack the lift is over-predicted, and the drag coefficient is usually under-predicted, because the boundary layer routines used cannot treat flows with large adverse pressure gradients, separated flows, or large wakes such as are present when the airfoil approaches stall.

For the thicker airfoils (thickness ratios of 18 percent or greater) the lift predicted by the NCSU program is more optimistic than for the thinner airfoils. For example, the 2424 airfoil really has a non-linear lift-curve slope which tends to reduce the lift, but the predicted lift curve slope is quite linear, resulting in the over-prediction of lift. The drag coefficients for the thicker airfoils are also seen to agree less well with experimental data than the drag coefficients of the thinner airfoils.

The NASA program usually gave a slightly higher predicted lift coefficient than did the NCSU program. The predicted drag coefficients for the NASA program, however, are in very poor agreement with experimental data. While the comparisons are generally poor, the NASA program drag coefficients do match experimental data better for the thicker airfoils than for the thinner airfoils.

Neither program does a very good job of predicting the aerodynamic characteristics with leading edge surface roughness as can be seen in the case of the 4412 airfoil (Figure 17). The roughness used in the test condition was approximated in the programs by specifying fixed transition at the leading edge for both the upper and lower surfaces of the airfoil. While both programs did a fairly good job of predicting the lift coefficient at the lower angles of attack, neither program did well for the drag

178

coefficient; the NASA program over-predicted the drag coefficient by 25 to 40 percent while the NCSU program under-predicted the drag by 20 to 40 percent.

The reader will notice that there are no predicted NASA coefficients for the 63_2-615 airfoil (Figure 23). While the NCSU program prediction is very good the NASA program was not able to produce an acceptable set of distributed solution points for this airfoil, and thus no correlation can be shown. In a similar situation, the NCSU program was not able to produce a set of distributed points for the 63_3-618 airfoil until two extra points were added in the nose region to those specified for the airfoil in TR 824 (Ref. 19). The extra points were needed because some of the specified points were extremely close together, leaving a large portion of the airfoil nose with no points to specify its shape. The same procedure was used for the 2424 airfoil whose coefficients are shown in Figures 13 and 14. For the 2424 four extra points were added in the region of the nose. As mentioned above, it is important to specify regions of high airfoil curvature properly. While there were a few exceptions, in most cases the ordinates given in TR 824 were sufficient to define the shape of the airfoil.

The Whitcomb airfoil which is currently being investigated by NASA Langley for possible application to general aviation craft is another airfoil which deserves some special mention. While the drawing of this airfoil in Figure 11 does not show it, the Whitcomb airfoil has its minimum thickness slightly ahead of the trailing edge. While the comparison of the NCSU program with experimental data is good for a Reynolds number of 1.9 million, this unusual trailing edge shape is probably one reason for the poor comparison with the previously unpublished experimental data at a Reynolds number of 9.2 million (Figure 29). The poor comparison at 9.2 million may also be due to the erratic nature of the experimental data at this Reynolds number.

In order to investigate another of the latest airfoil designs, that presented in TN D-7071 (Ref. 52) was investigated using the two computer programs. The airfoil (referred to as the TN D-7071 airfoil) was originally designed to optimize the maximum lift coefficient. Experimental results are given for the pressure coefficients and the lift coefficients in Reference 52, but no drag data is presented. This airfoil was of particular interest because of its unusual shape (see Figure 11) and because the original NASA program was used in Reference 52 for a theoretical correlation with the experimental results. Figure 30 indicates that the lift coefficients for the TN D-7071 airfoil obtained from the NCSU program match the experimental values a little better than do the NASA program values. The drag coefficients for both programs are also included for the sake of completeness even though no experimental drag coefficients are available. Figures 31, 32, and 33 give the pressure distributions over the airfoil for experimental measurement, the NASA program, and the NCSU program for this airfoil at angles of attack of 3.4, 12.4, and 18.7 degrees respectively. These pressure coefficients were included to show that the pressure coefficients obtained from the NCSU program

are still basically the same as those predicted by the NASA program. As indicated in Reference 31 the computed pressure coefficients for conventional airfoils usually compared quite well with experimental data.

In order to study some of the bounds of the computational methods presented herein, an attempt was made to predict the aerodynamic characteristics of the last airfoil shown in Figure 11. This is a very thin, highly cambered airfoil recently tested by Milgram (Ref. 108) over a very wide range in Reynolds Numbers and angles of attack. The very non-linear lift curve and high drag coefficients evident in the experimental data indicate extensive flow separation which changes radically with the changes in angle of attack. Figure 34 compares the predictions with the experimental data. The lift predictions are close to experimental values only for an angle of attack of 2-3°. At large angles of attack where the thin nose of the airfoil points into the wind, more or less, there is extensive separation over the aft portion of the upper surface so that the predicted lift does not materialize. The best that can be said for the drag predictions are that the NCSU program is qualitatively correct but predicts only 25% of the actual drag. This experience leads one to suggest extreme caution in applying the program to airfoils which possess regions of surface concavity. Not only does the boundary layer routine used here fail for what in these circumstances will usually be a separated flow, but the inviscid method also has difficulty with such geometries.

As received from Langley, the program contained a provision to modify both the inviscid and viscous computations of the pressure distribution and aerodynamic force coefficients for the effects of changes in free stream Mach number. This capability was retained although it is expected to be of limited utility for most light aircraft designs. For this reason, the applicability to varied airfoils and the accuracy limits of the routine were not investigated extensively. Plotted in Figure 35 are the results for a 23012 airfoil obtained with the original Langley version and the modified NCSU version. It will be seen that for this case and for $\alpha = 0$, at least, the NCSU version gives qualitatively reasonable lift results for all Mach numbers while the original version is reliable to about $M \simeq 0.50$. The reasons for the apparent superiority of the NCSU version were not examined in detail. No explanation for this behavior can, therefore, be offered. Experimental data for the 23012 airfoil were not at hand for comparison, but one may note that generally the Prandtl-Glauert rule under-predicts the increase in C_L near M_{CR} by about 10%. Thus, the NCSU prediction may yield results quantitatively below those found experimentally for $M > 0.5$. The user should therefore approach results for $M > 0.5$ cautiously.

The results of the drag predictions are shown in Figure 36. The Langley results appear to be completely meaningless. Again, the reason for this is not known. The NCSU predictions, on the other hand, appear to be qualitatively correct for all Mach numbers. Quantitatively, the Squire-Young formula seems to give the more reasonable results for $M > 0.55$.

180

Based on the airfoils investigated the authors conclude that:

(1)   the NCSU program gives about the same pressure distribution over the airfoil as does the NASA program

(2)   the lift coefficients for the NCSU program while still a little too large for some airfoils, compare more favorably with experimental data than do the NASA program coefficients

(3)   the predicted drag coefficients using the NCSU program compare much better with experimental data than the drag coefficients obtained from the NASA program.
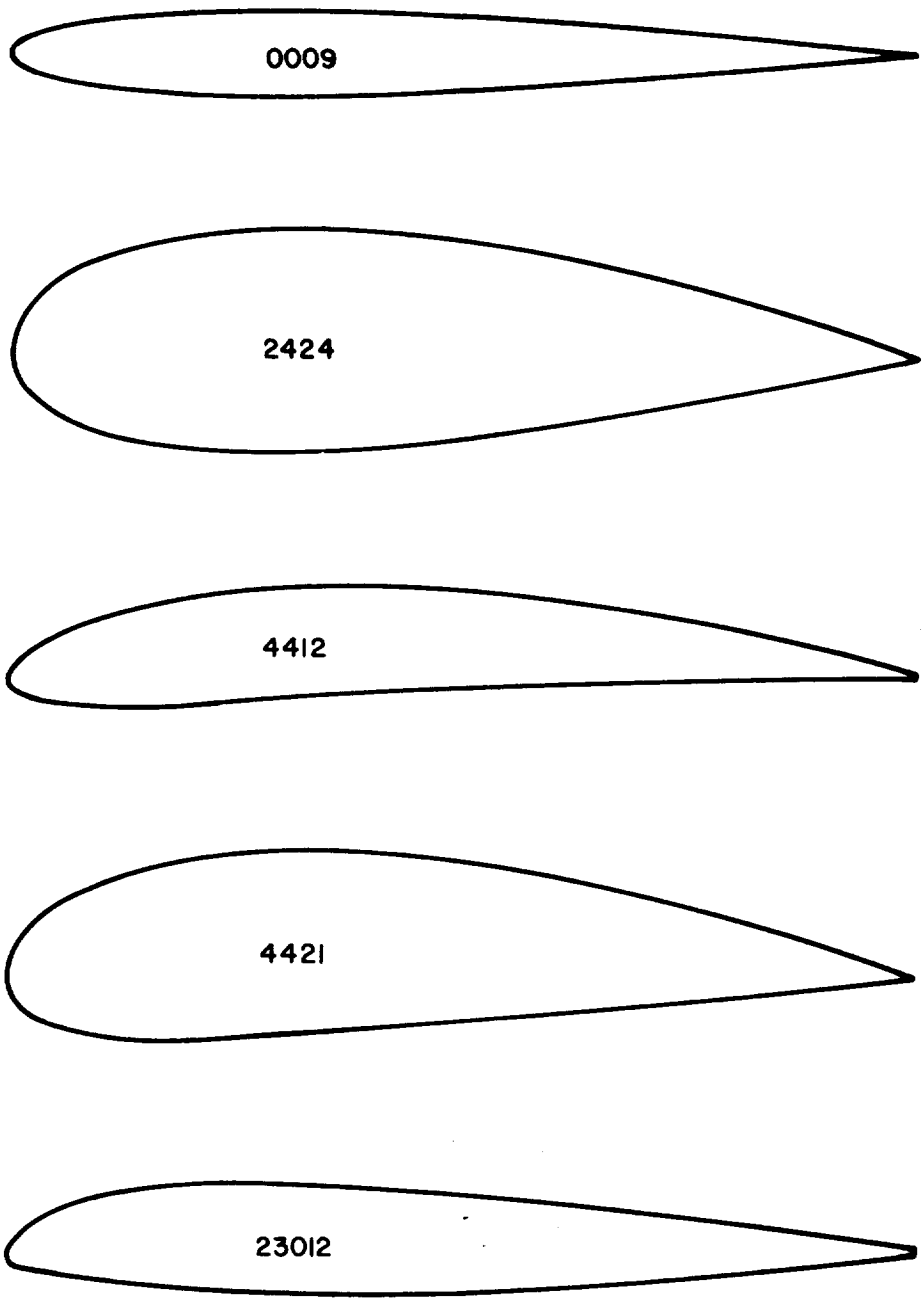
Figure 11.   General shape of the [15] airfoils investigated for lift and drag characteristics.

23021

63-006

63_2-615

63_3-618

63_4-421

Figure 11. Continued.

64₁-412

64₄-421

WHITCOMB

TN D-7071

S-14.1

Figure 11. Continued.

184
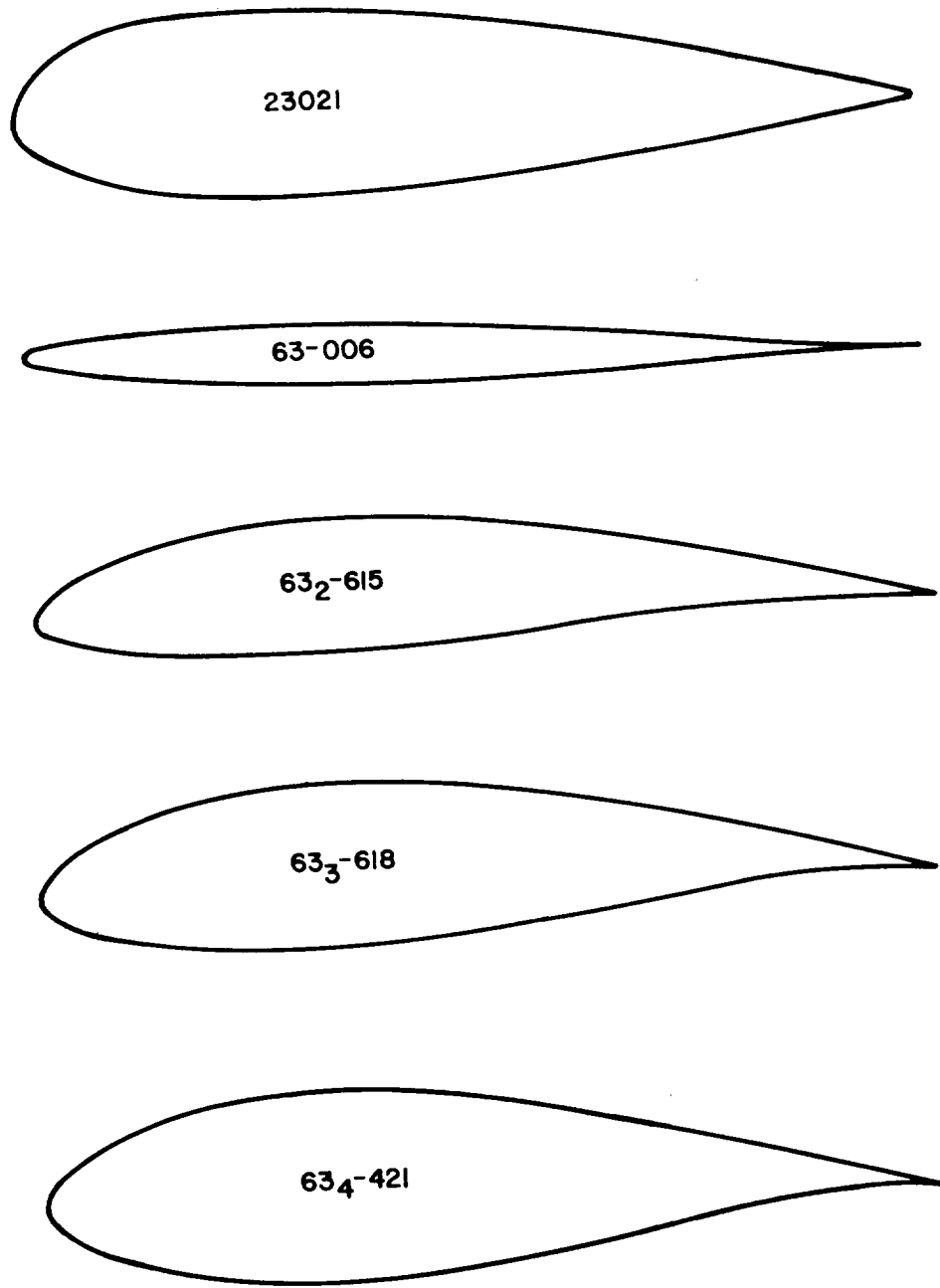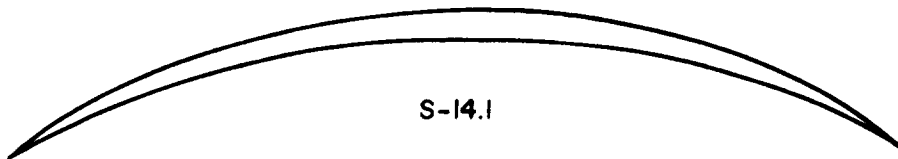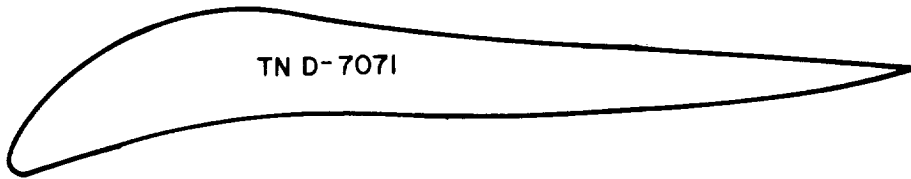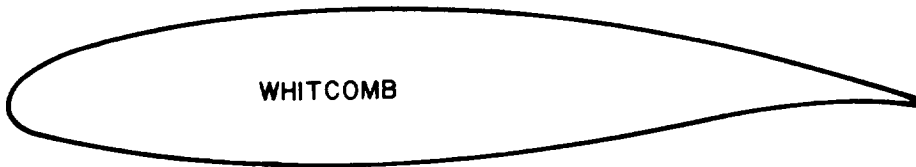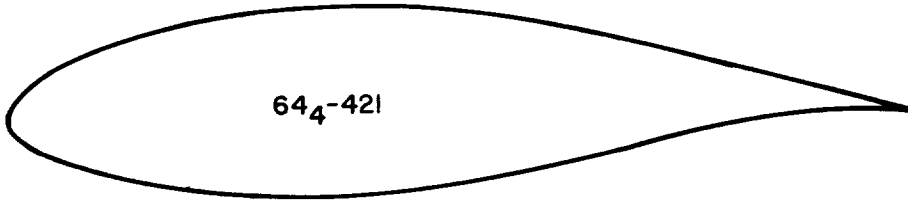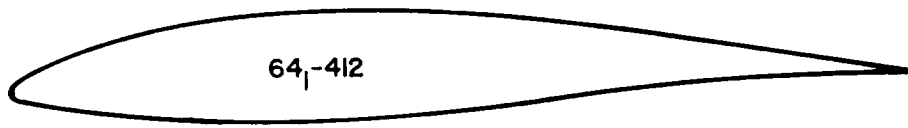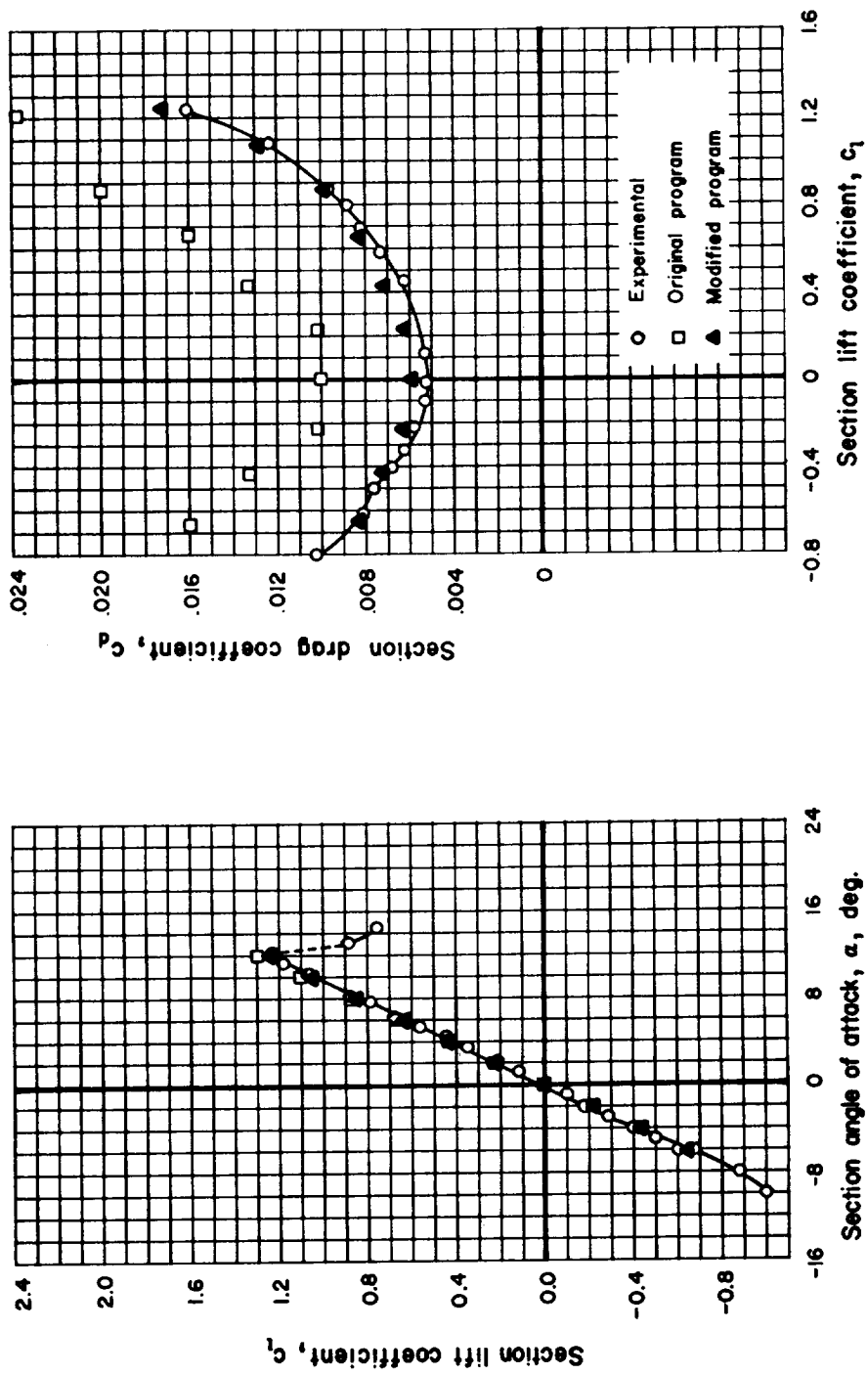
Figure 12. Comparison of 2-D lift and drag coefficients of the 0009 airfoil at a Reynolds number of 3,000,000.

185
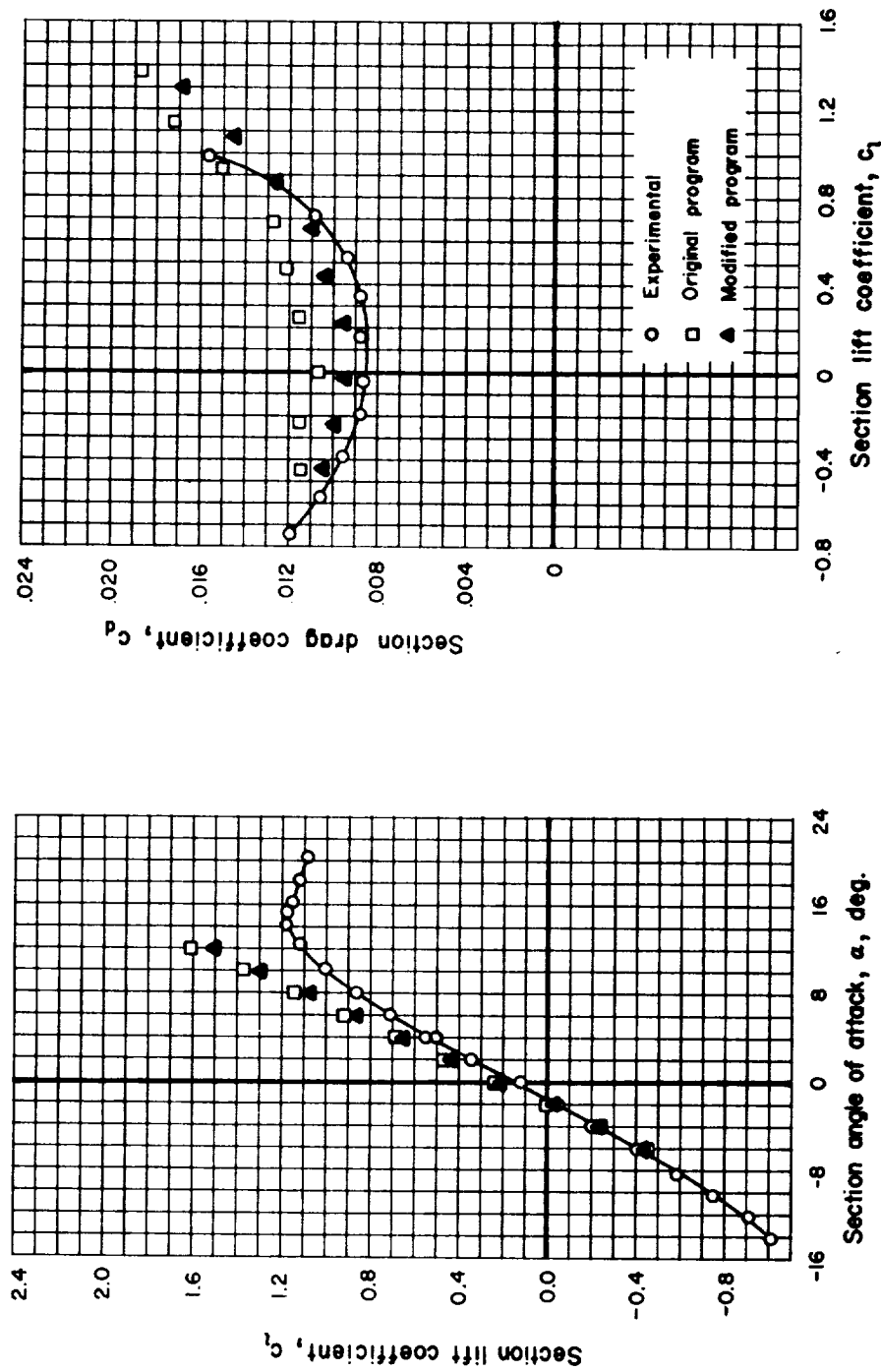
Figure 13. Comparison of 2-D lift and drag coefficients of the 2424 airfoil at a Reynolds number of 2,900,000.

186
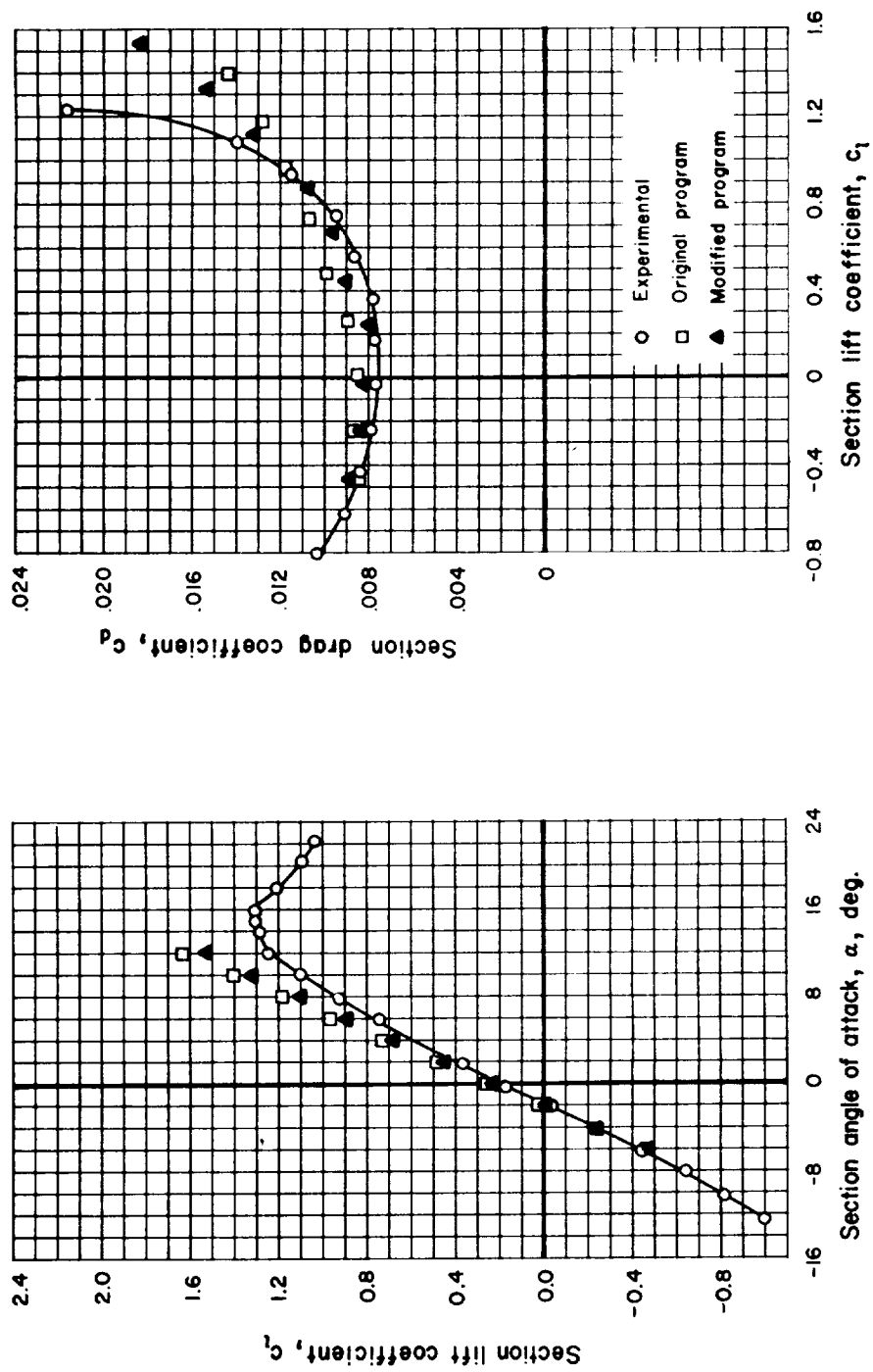
Figure 14. Comparison of 2-D lift and drag coefficients of the 2424 airfoil at a Reynolds number of 9,000,000.
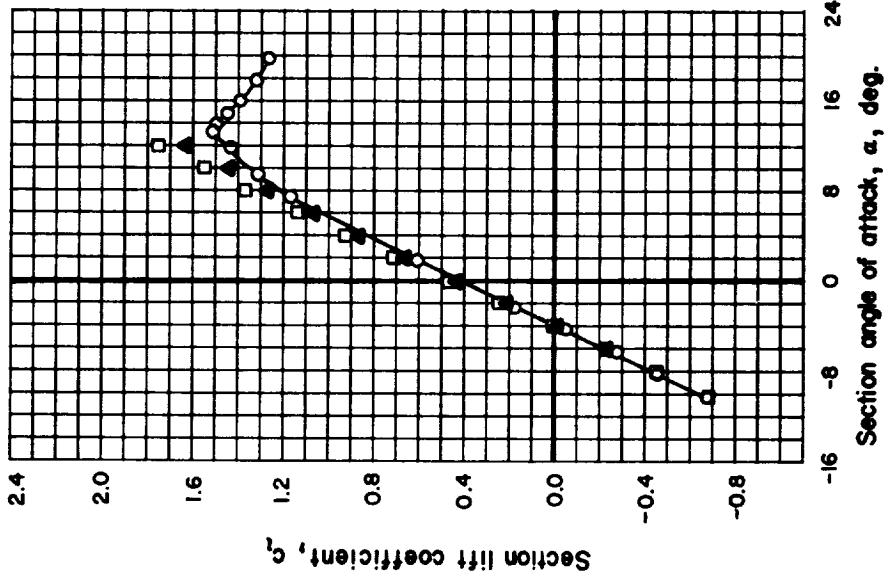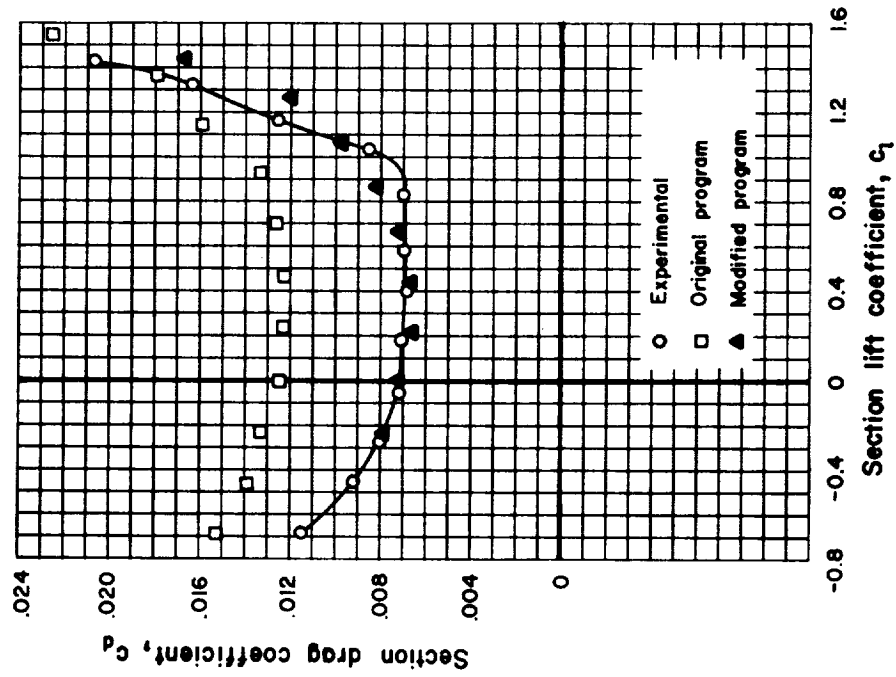
Figure 15. Comparison of 2-D lift and drag coefficients of the 4412 airfoil at a Reynolds number of 3,000,000.

188
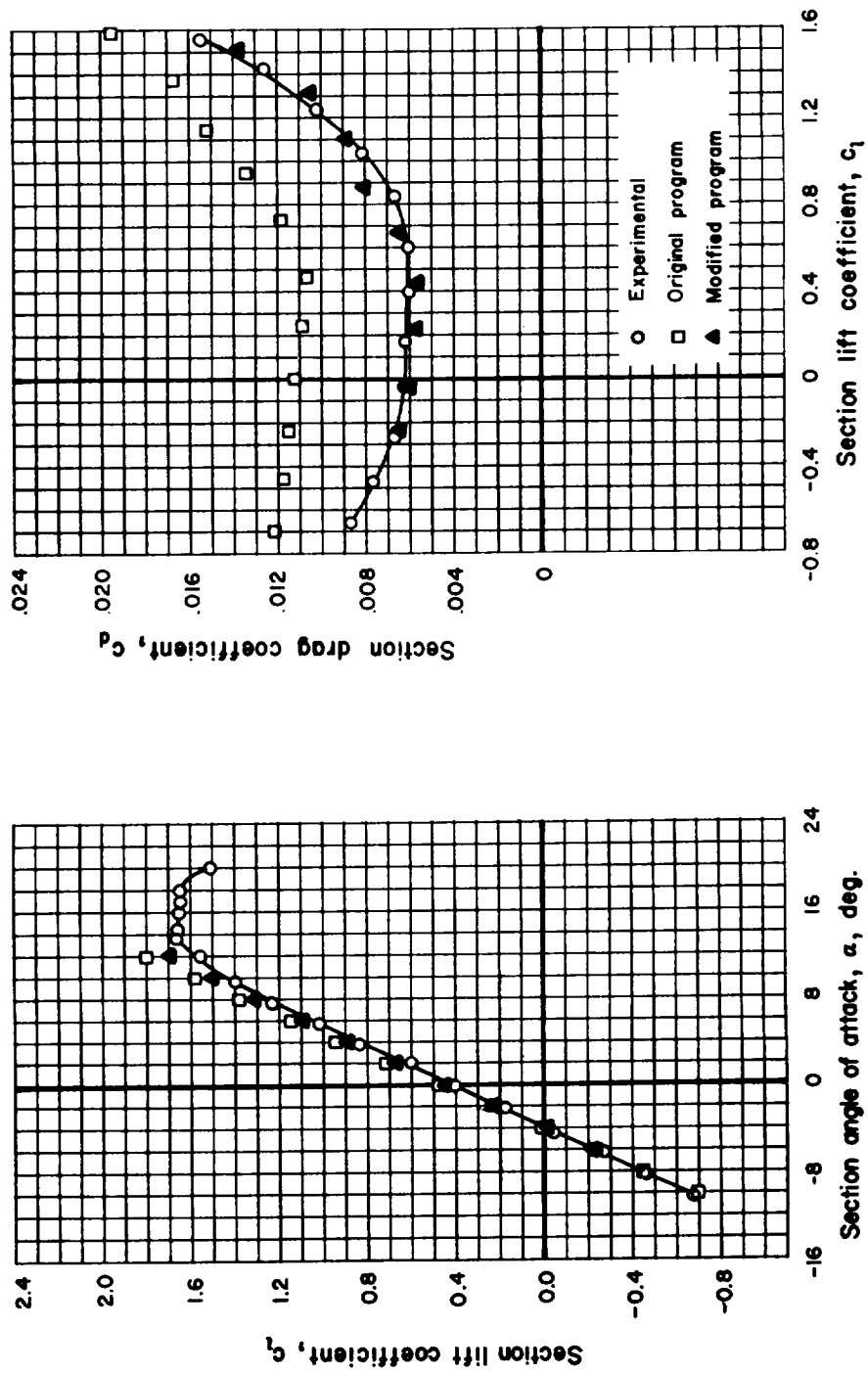
Figure 16. Comparison of 2-D lift and drag coefficients of the 4412 airfoil at a Reynolds number of 9,000,000.

189
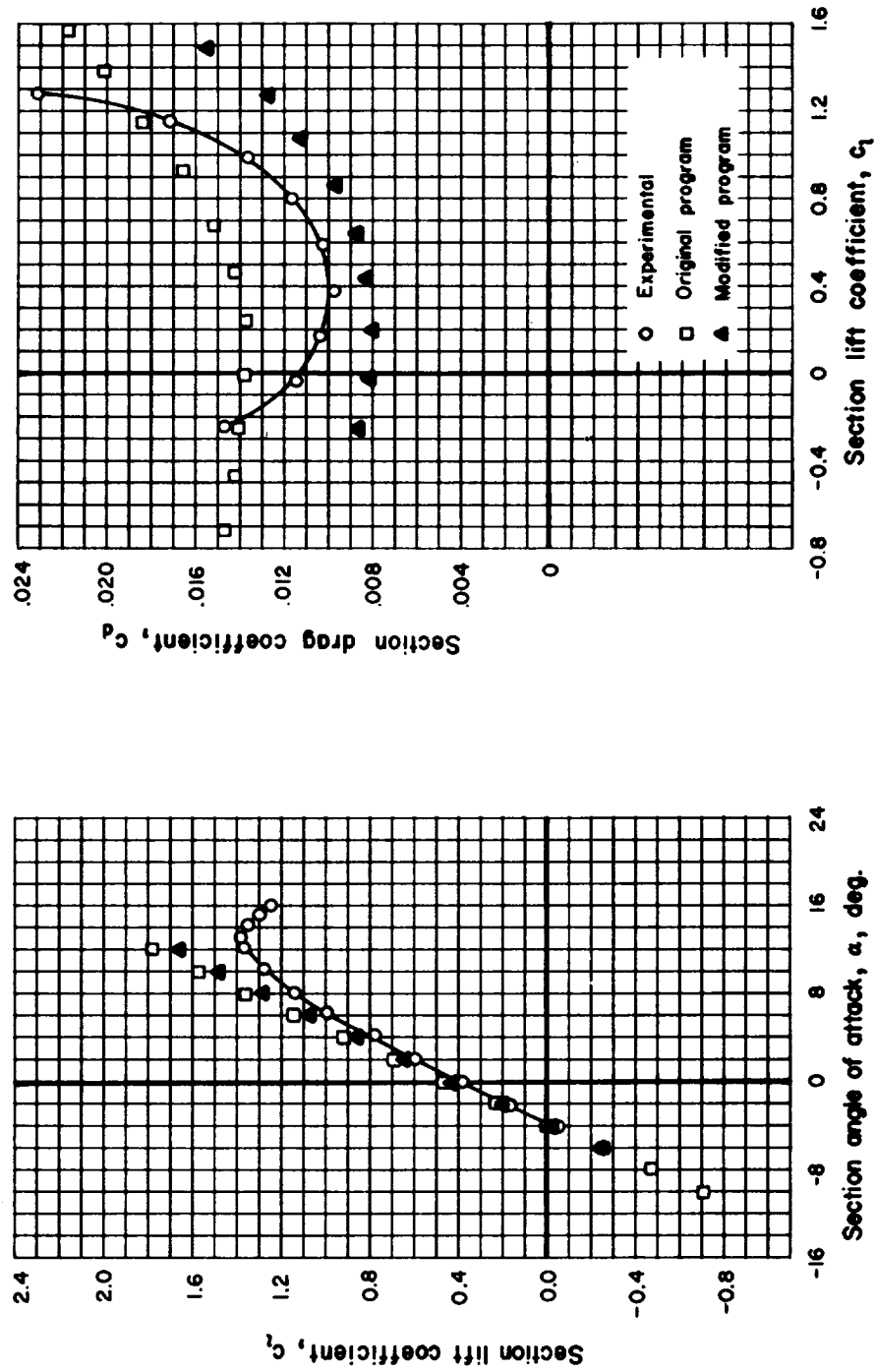
Figure 17. Comparison of 2-D lift and drag coefficients of the 4412 airfoil at a Reynolds number of 6,000,000 with fixed upper and lower surface leading edge transition to approximate standard roughness.
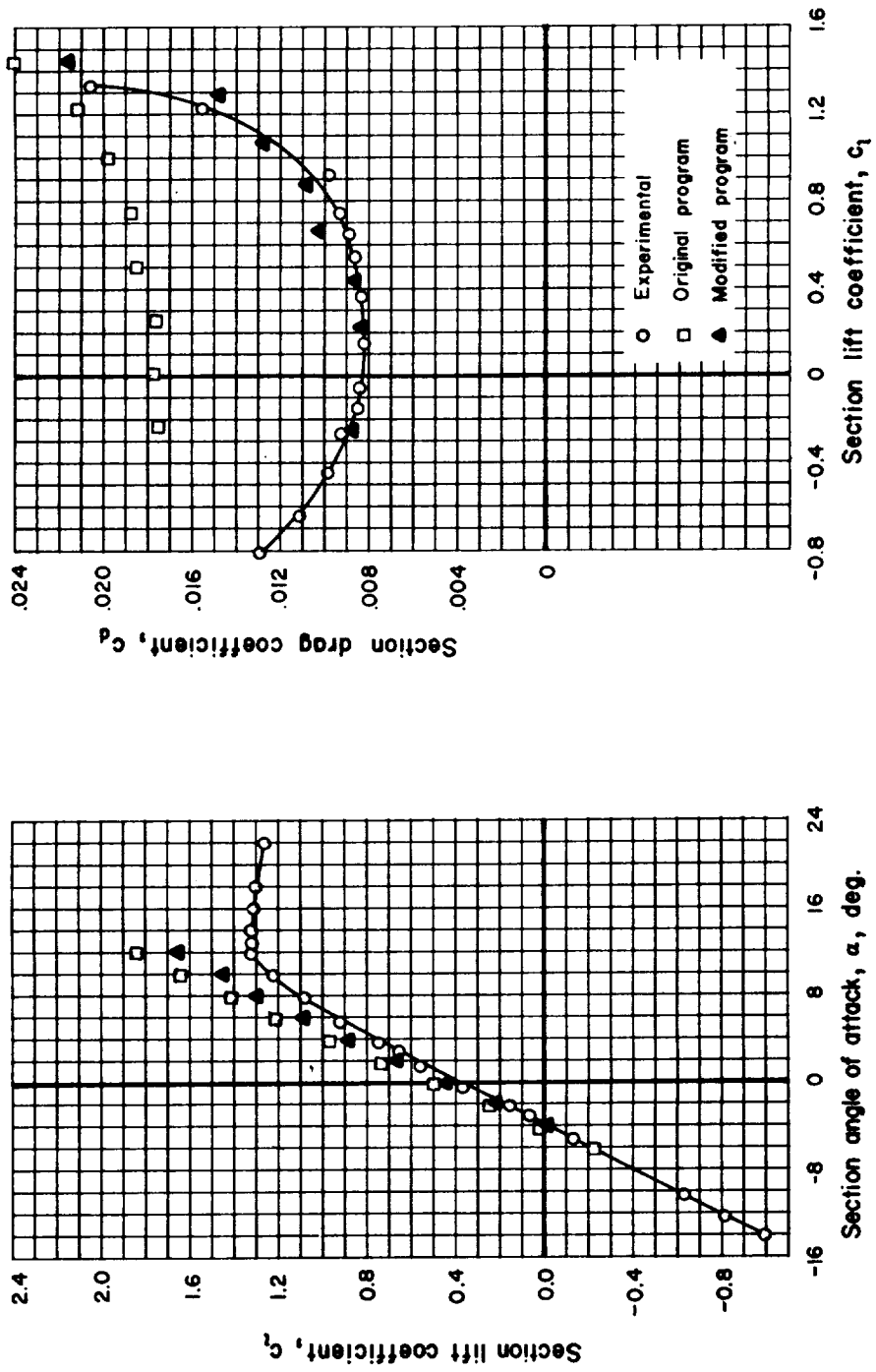
Figure 18. Comparison of 2-D lift and drag coefficients of the 4421 airfoil at a Reynolds number of 3,000,000.

191
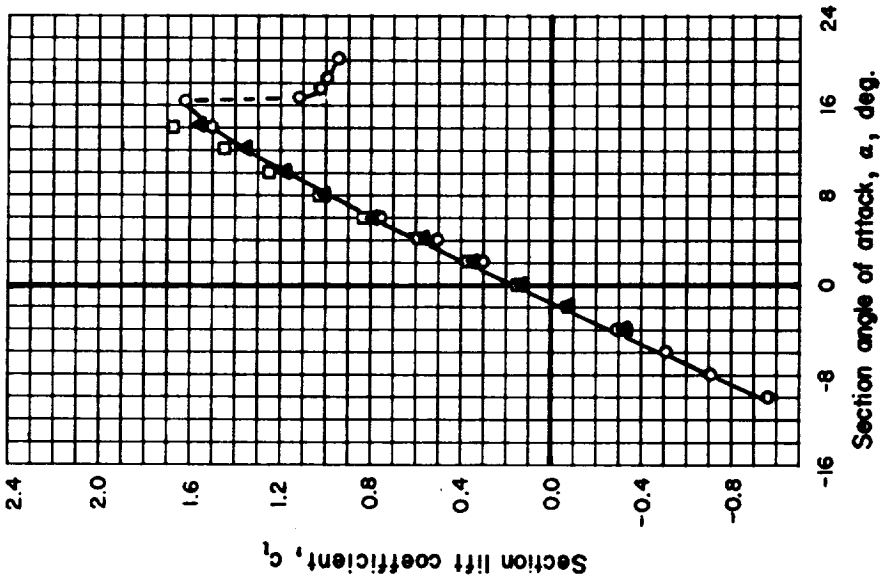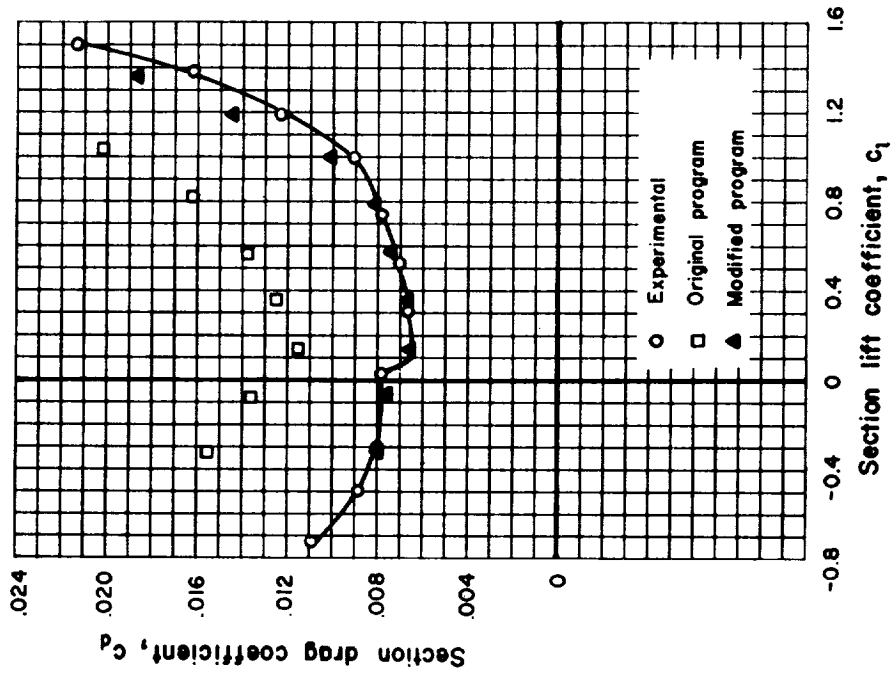
Figure 19. Comparison of 2-D lift and drag coefficients of the 23012 airfoil at a Reynolds number of 3,000,000.

192
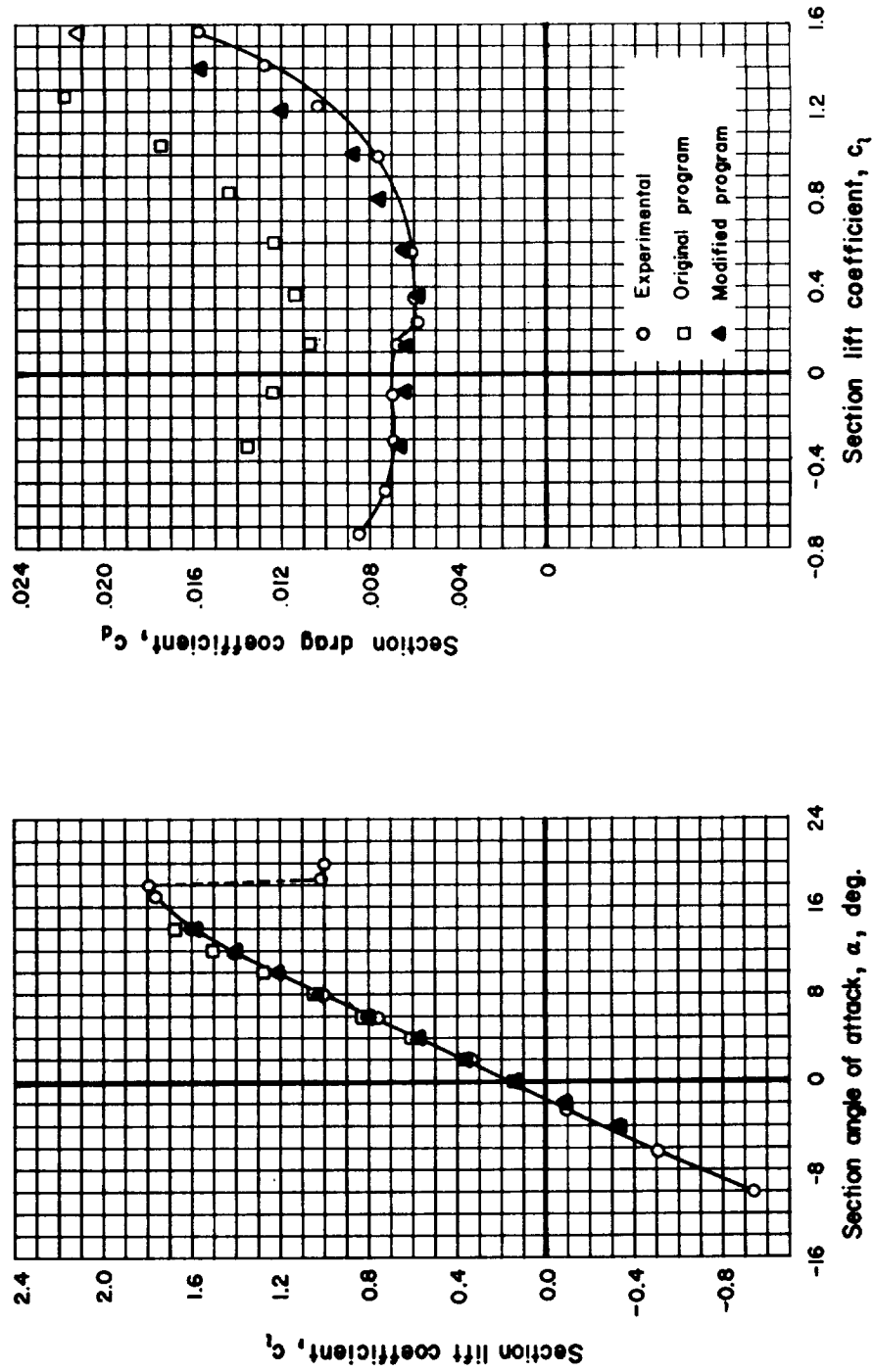
Figure 20. Comparison of 2-D lift and drag coefficients of the 23012 airfoil at a Reynolds number of 8,800,000.

193
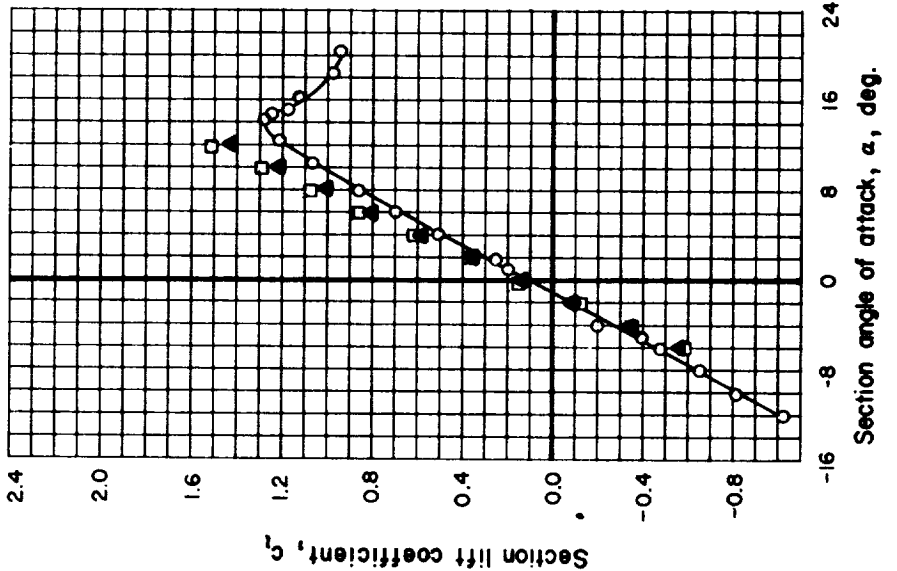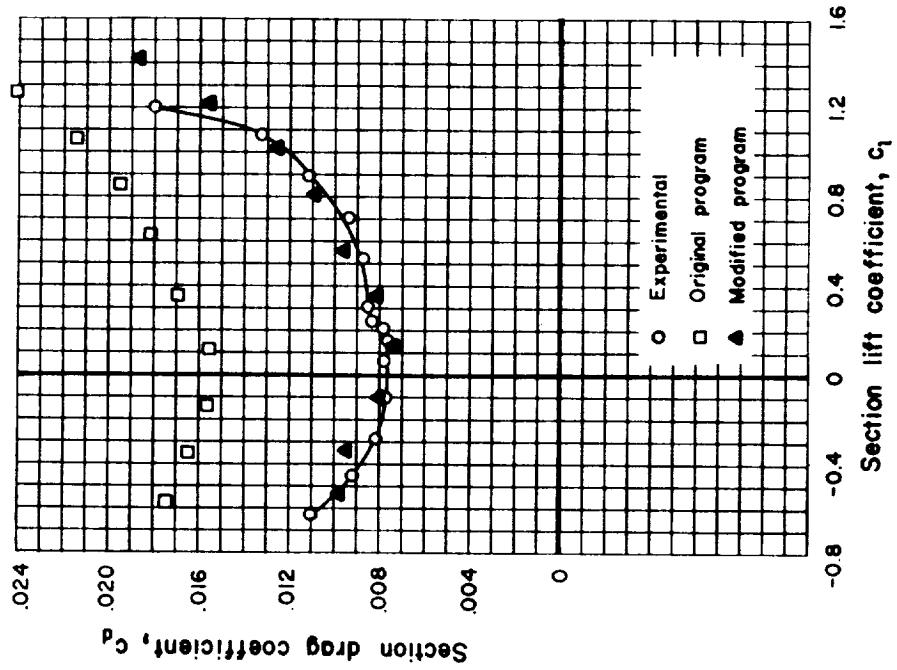
Figure 21. Comparison of 2-D lift and drag coefficients of the 23021 airfoil at a Reynolds number of 3,000,000.

194
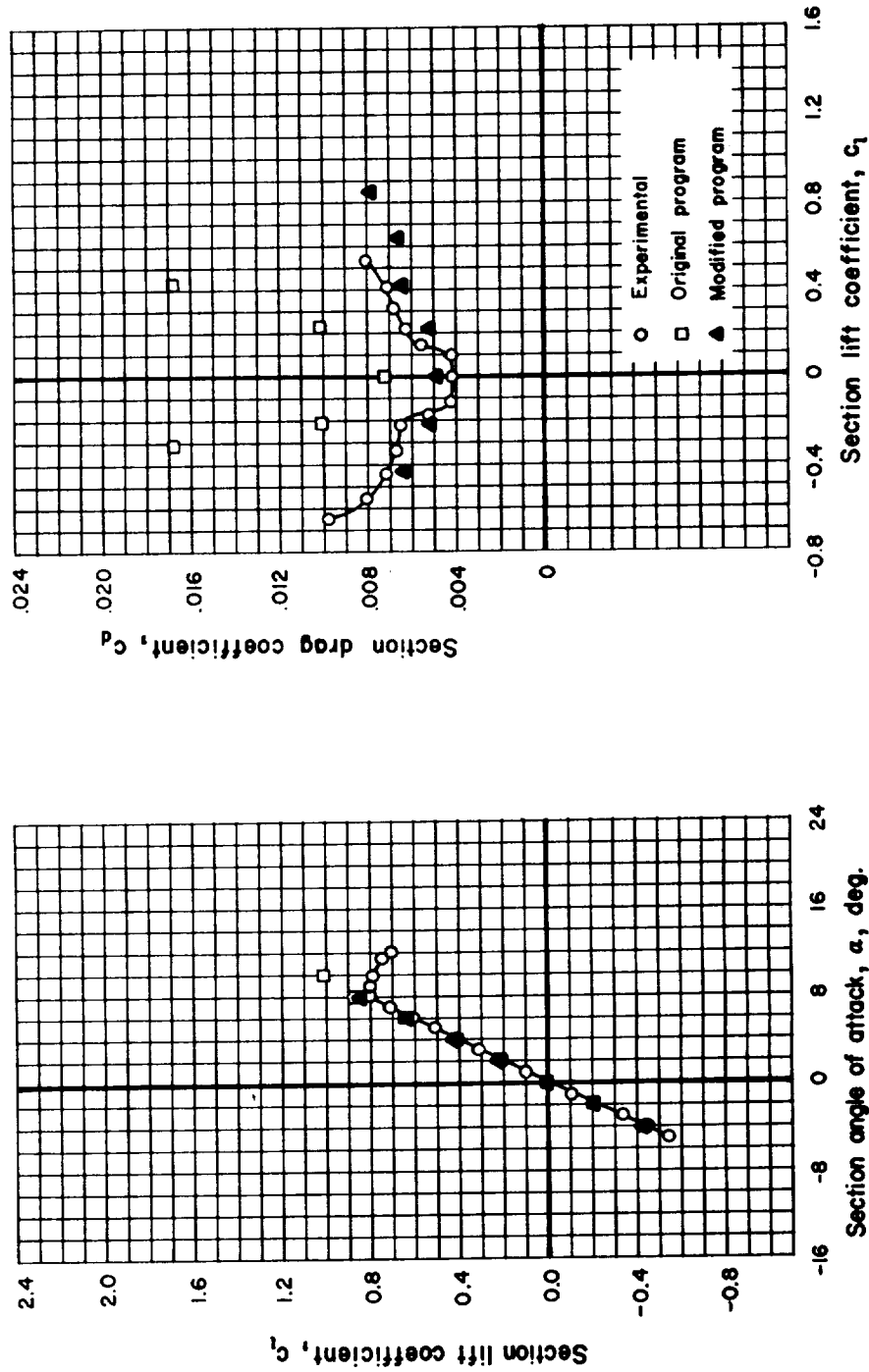
Figure 22. Comparison of 2-D lift and drag coefficients of the 63-006 airfoil at a Reynolds number of 3,000,000.

195
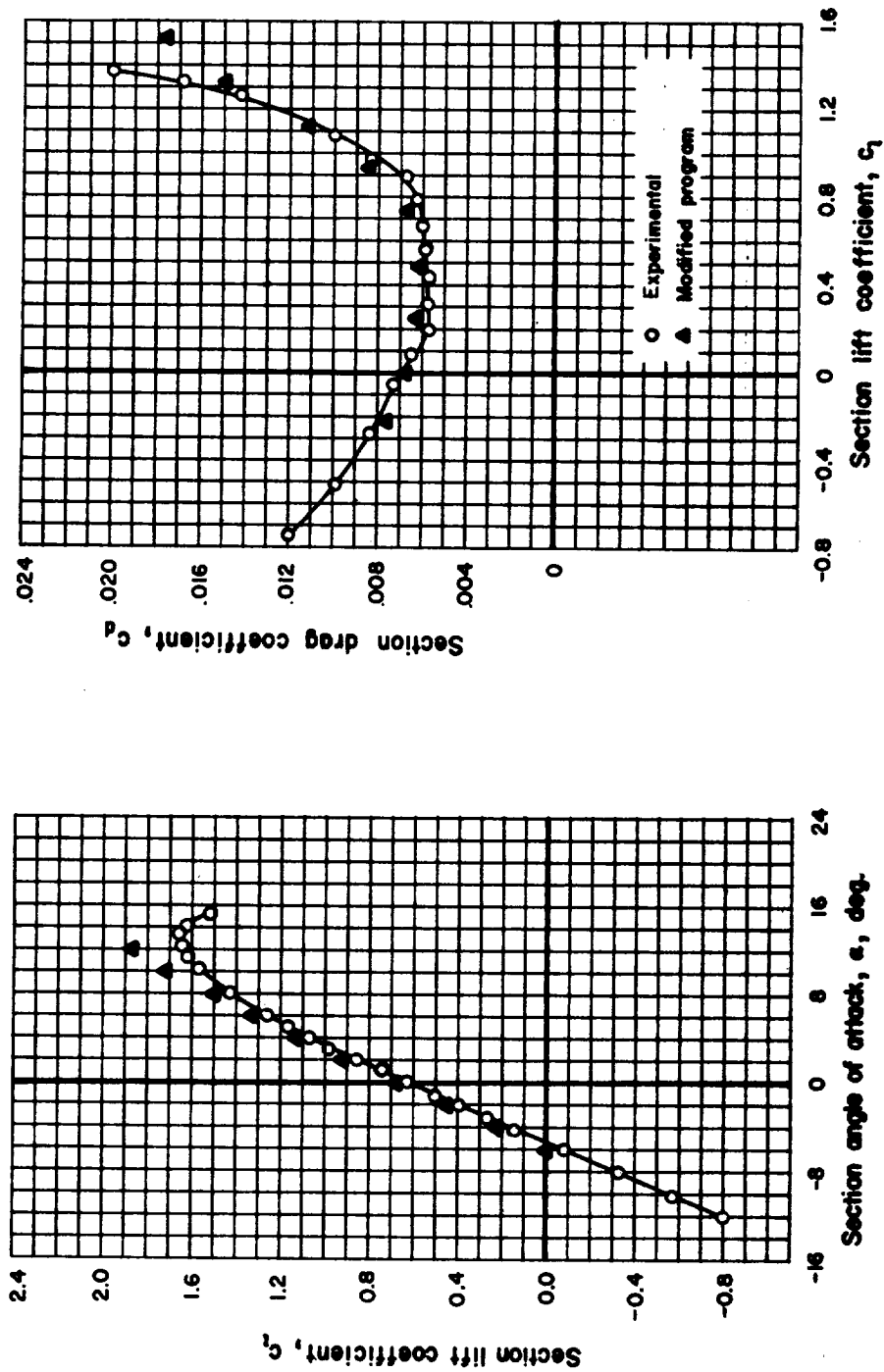
Figure 23. Comparison of 2-D lift and drag coefficients of the 63_2-615 airfoil at a Reynolds number of 3,000,000.
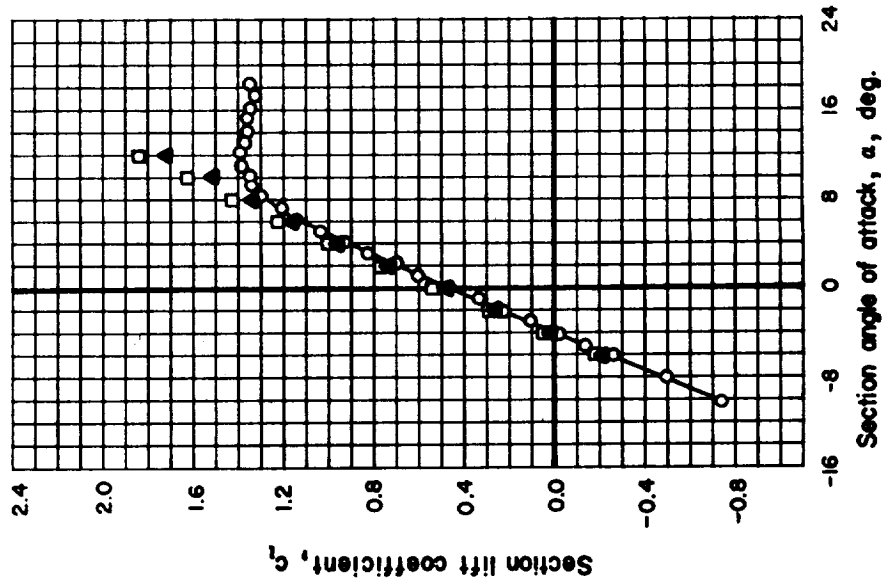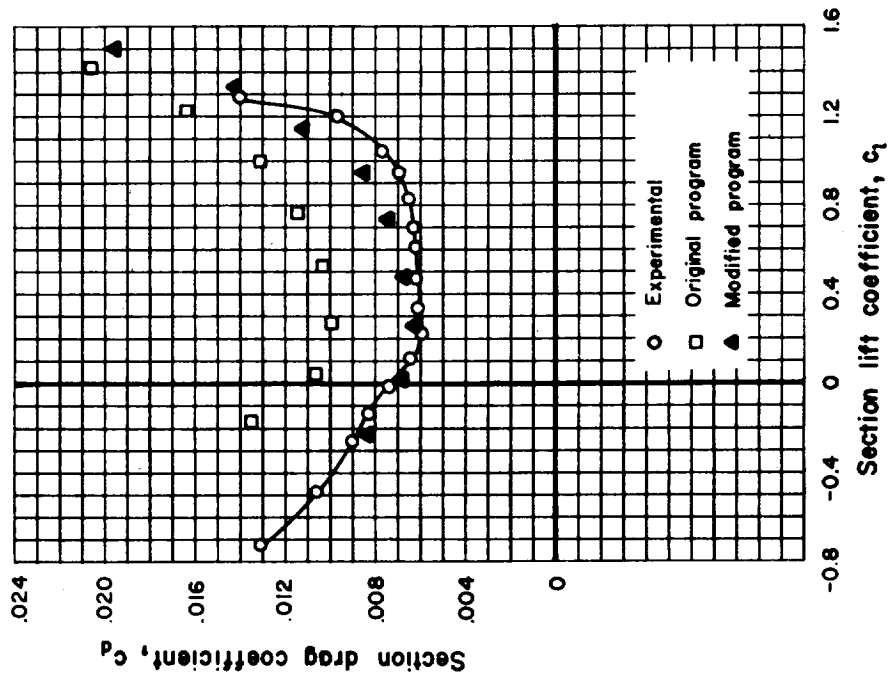
Figure 24. Comparison of 2-D lift and drag coefficients of the 63_3-618 airfoil at a Reynolds number of 3,000,000.
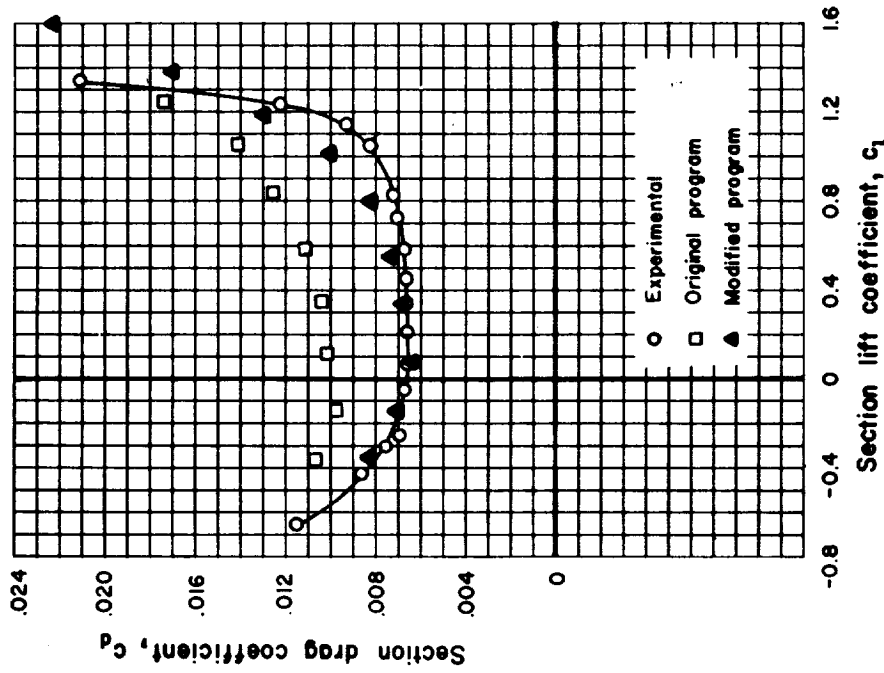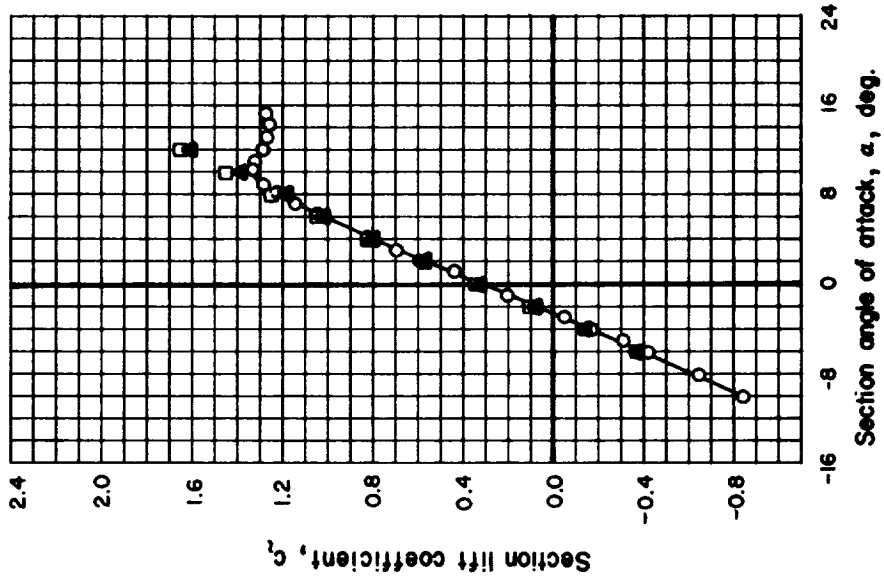
Figure 25. Comparison of 2-D lift and drag coefficients of the 63_4-421 airfoil at a Reynolds number of 3,000,000.

198
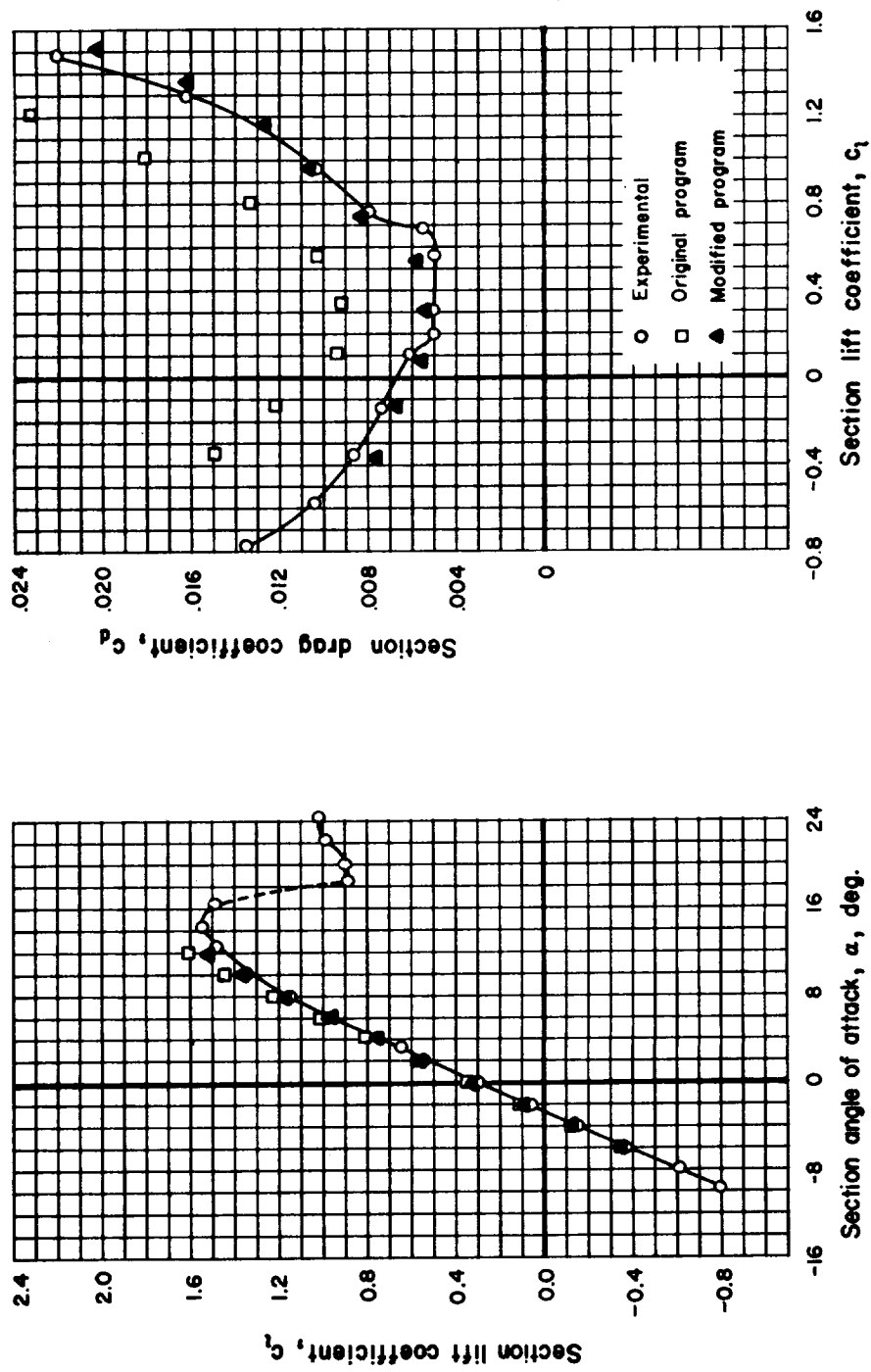
Figure 26. Comparison of 2-D lift and drag coefficients of the 64_1-412 airfoil at a Reynolds number of 3,000,000.

199
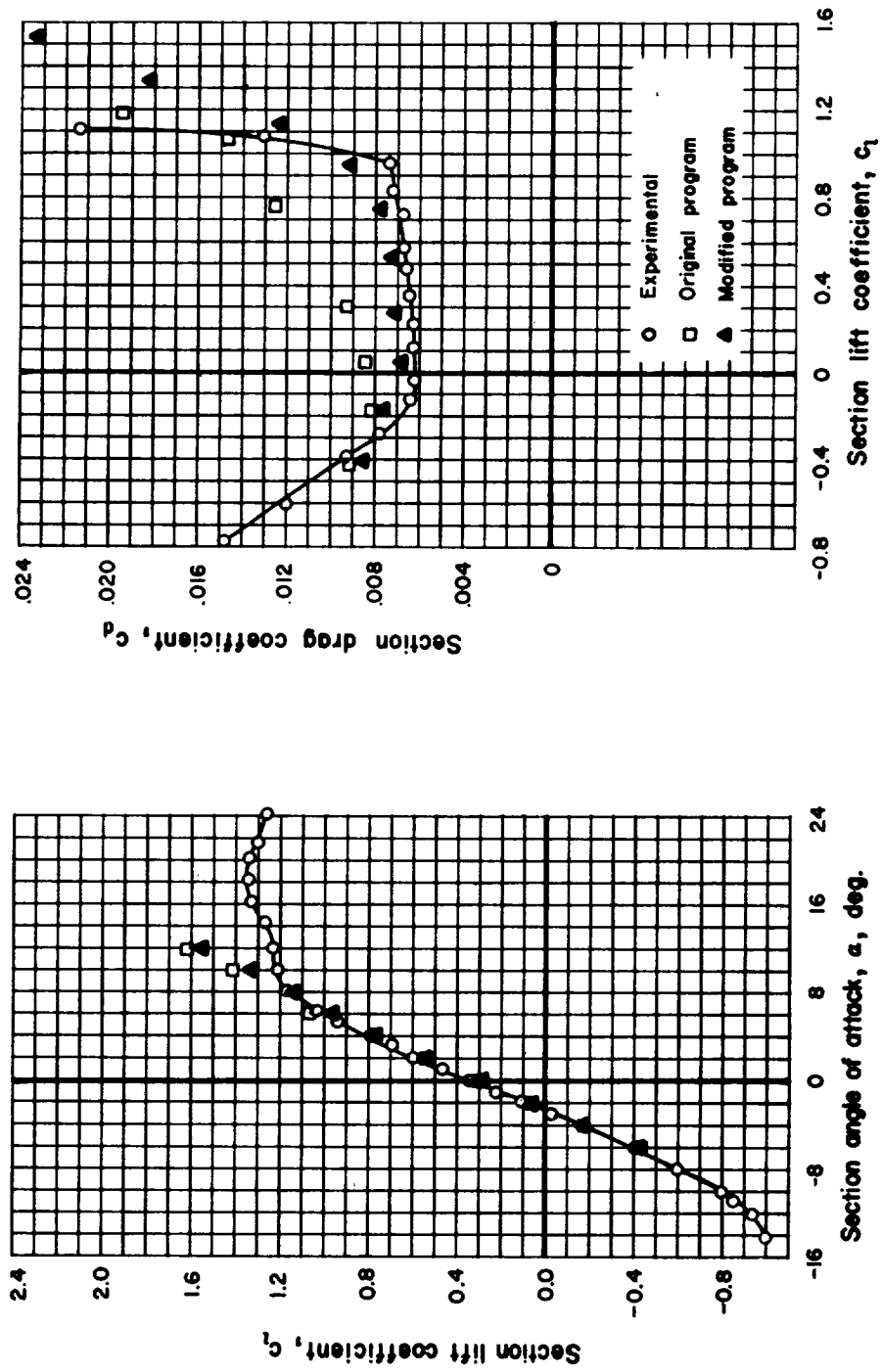
Figure 27. Comparison of 2-D lift and drag coefficients of the 64_4-421 airfoil at a Reynolds number of 3,000,000.

200
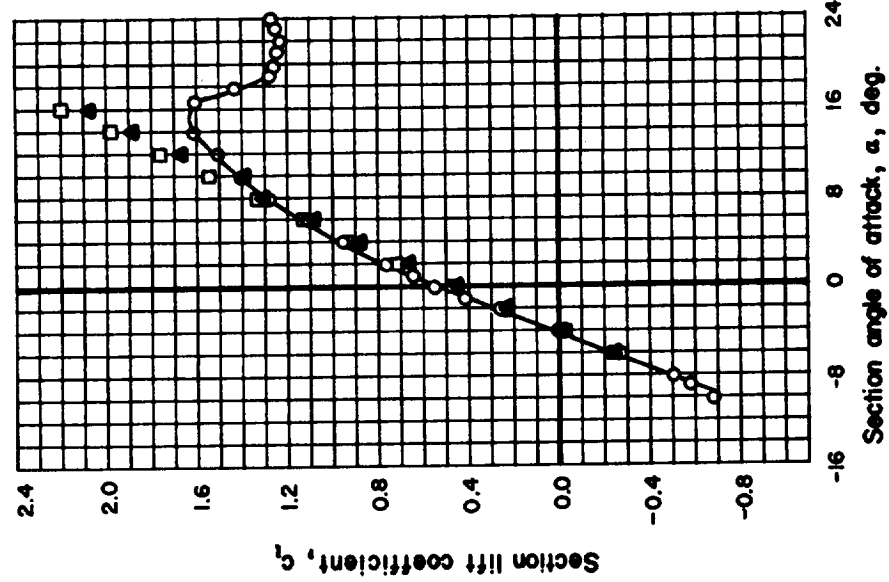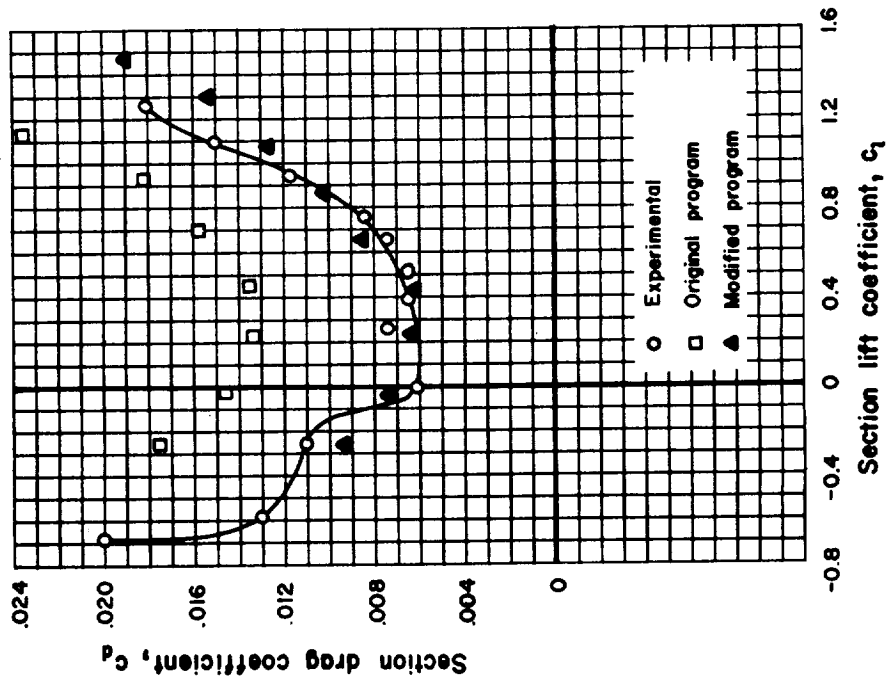
Figure 28. Comparison of 2-D lift and drag coefficients of the Whitcomb airfoil at a Reynolds number of 1,900,000.

201
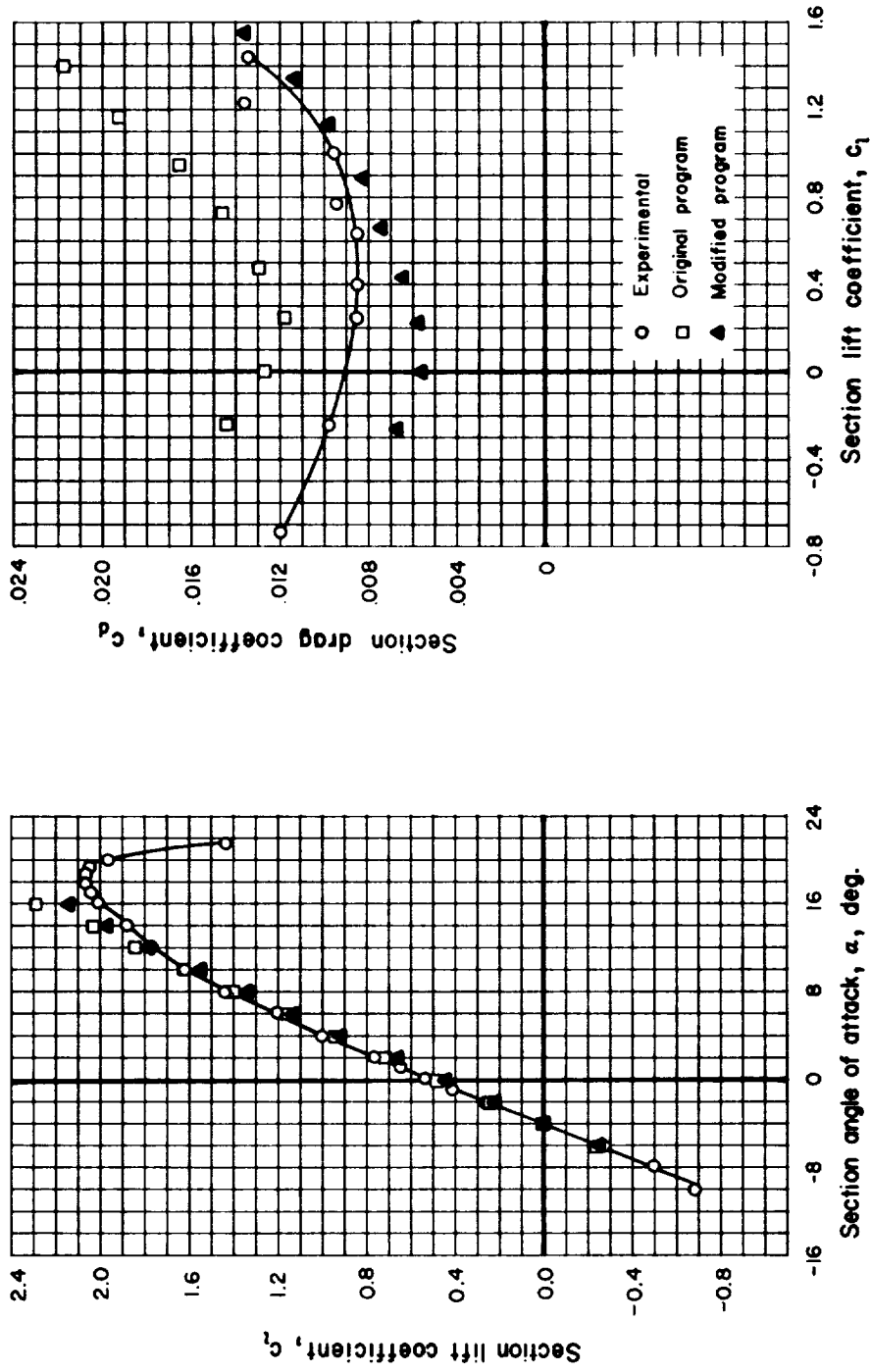
Figure 29. Comparison of 2-D lift and drag coefficients of the Whitcomb airfoil at a Reynolds number of 9,200,000.

202
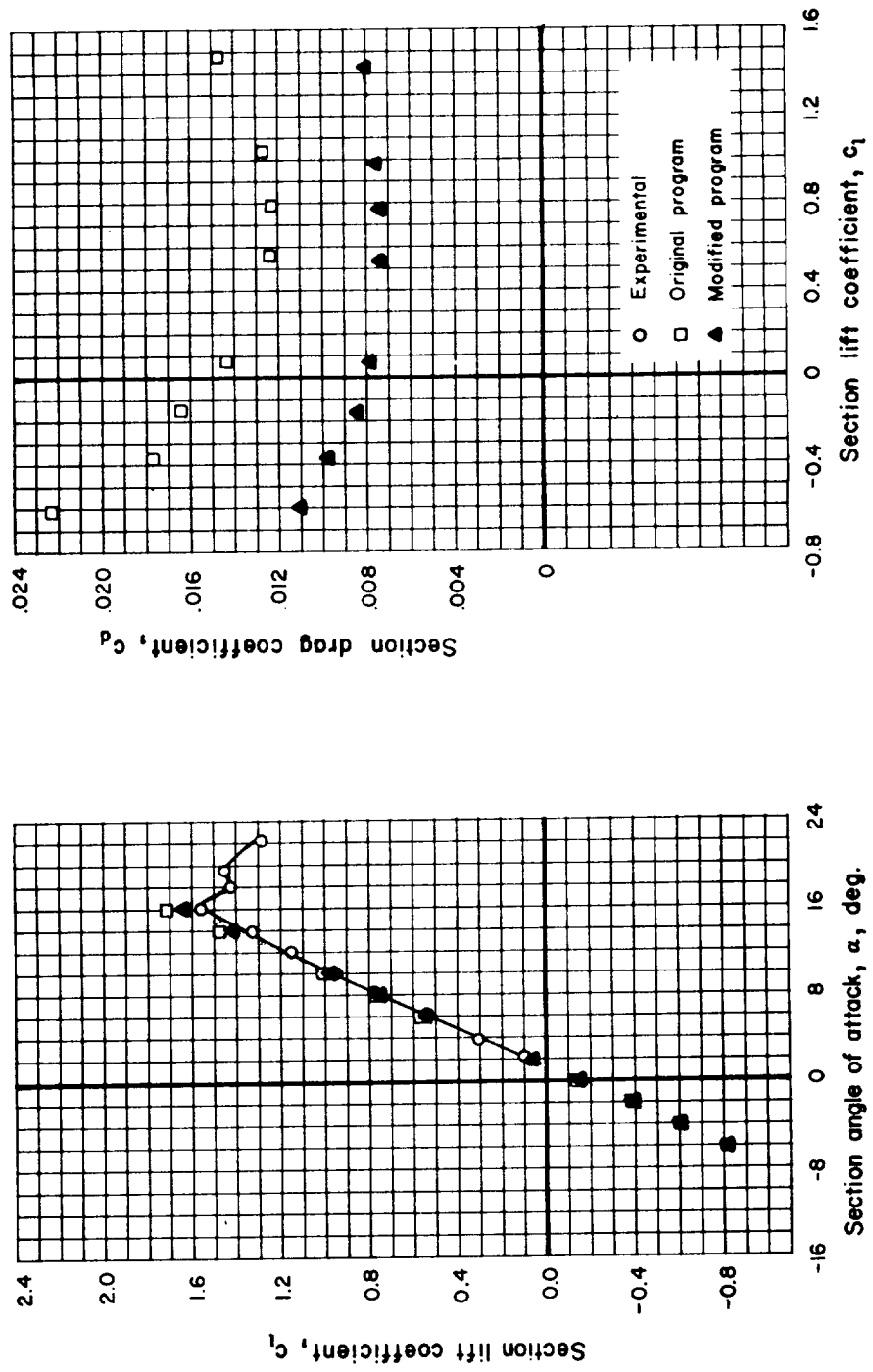
Figure 30. Comparison of 2-D lift and drag coefficients of the TN D-7071 airfoil at a Reynolds number of 9,000,000 (note experimental drag coefficients are not available).

203
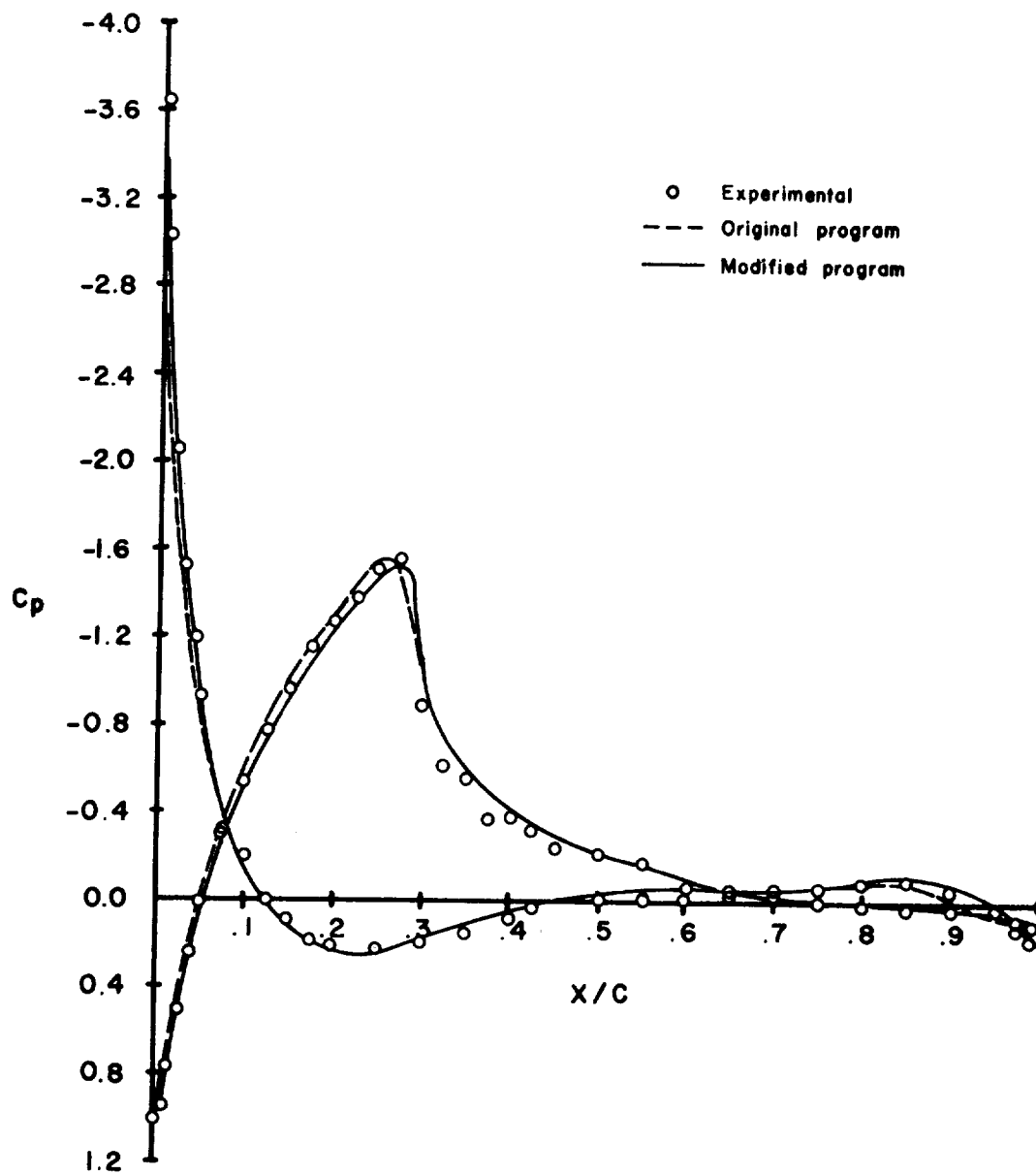
Figure 31. Comparison of pressure coefficients of the TN D-7071 airfoil at a Reynolds number of 9,000,000 and angle of attack of 3.4 degrees.
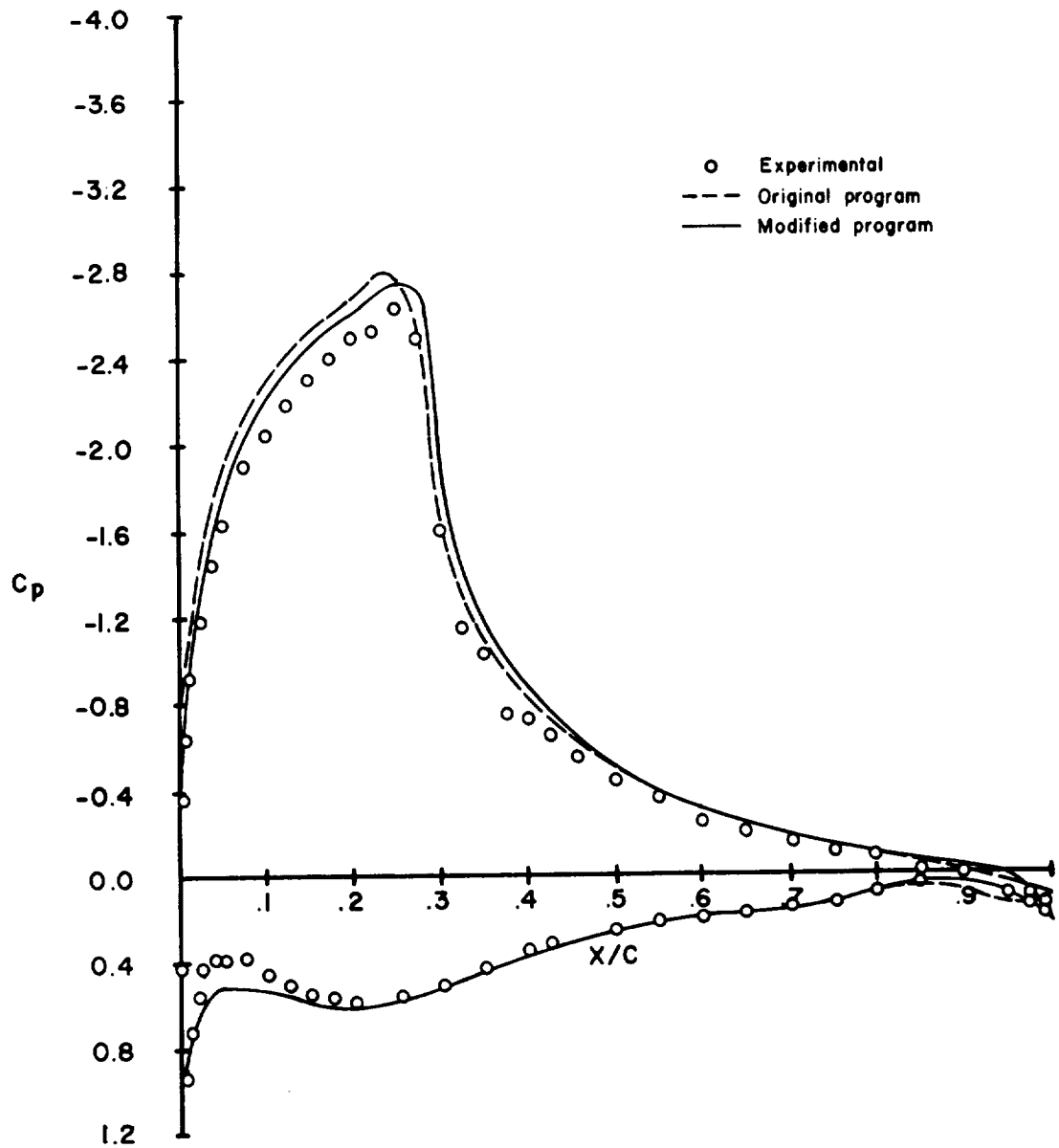
Figure 32. Comparison of pressure coefficients of the TN D-7071 airfoil at
a Reynolds number of 9,000,000 and angle of attack of 12.4
degrees.
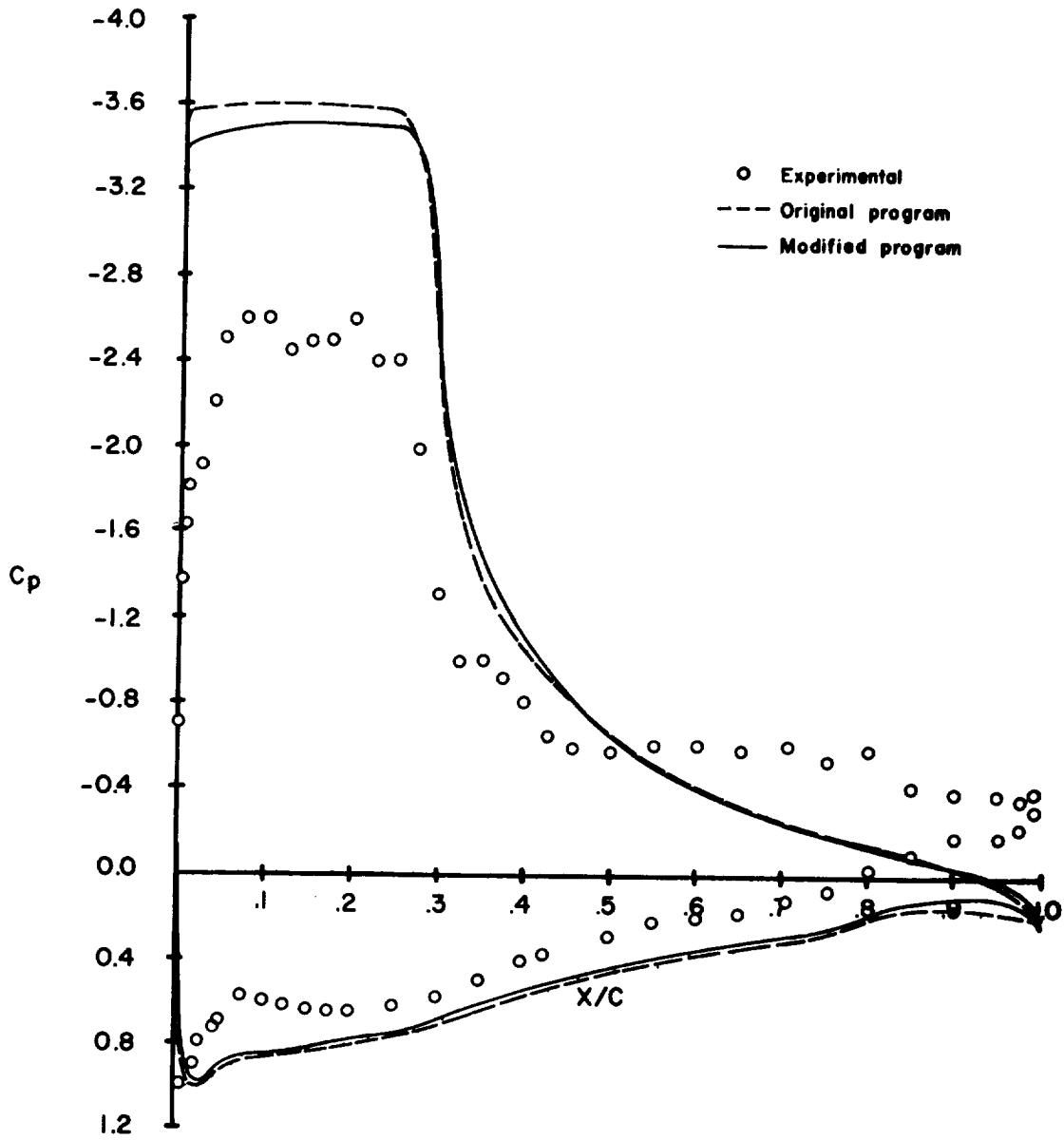
Figure 33. Comparison of pressure coefficients of the TN D-7071 airfoil at a Reynolds number of 9,000,000 and angle of attack of 18.7 degrees.

206

Figure 34. Comparison of 2-D lift and drag coefficients of the S-14.1 airfoil (Ref. 108) at a Reynolds Number of 600,000.

207
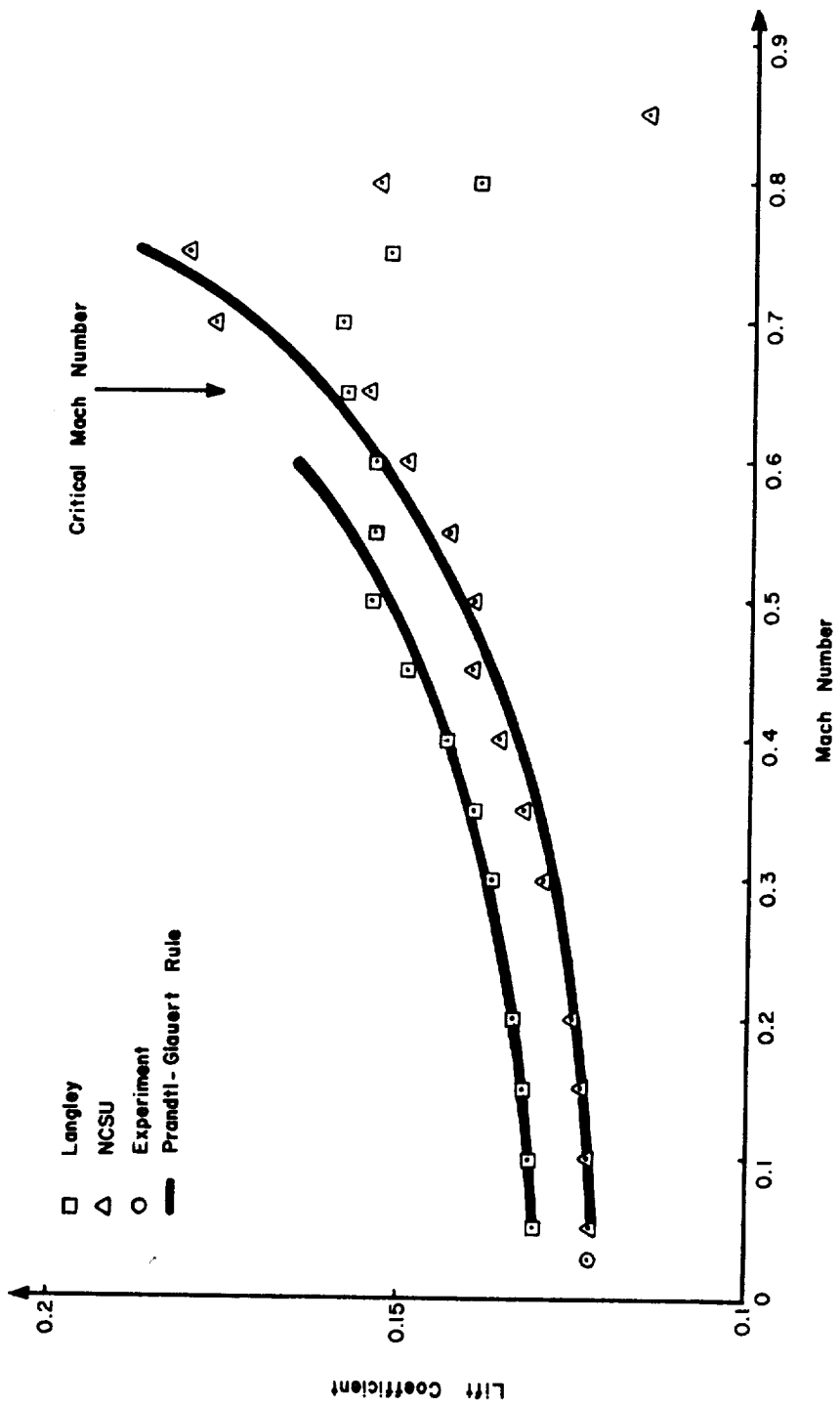
Figure 35. Predictions of C_L variation with Mach Number at $\alpha = 0°$ and a Reynolds Number of 3,000,000 for the 23012 airfoil.

208
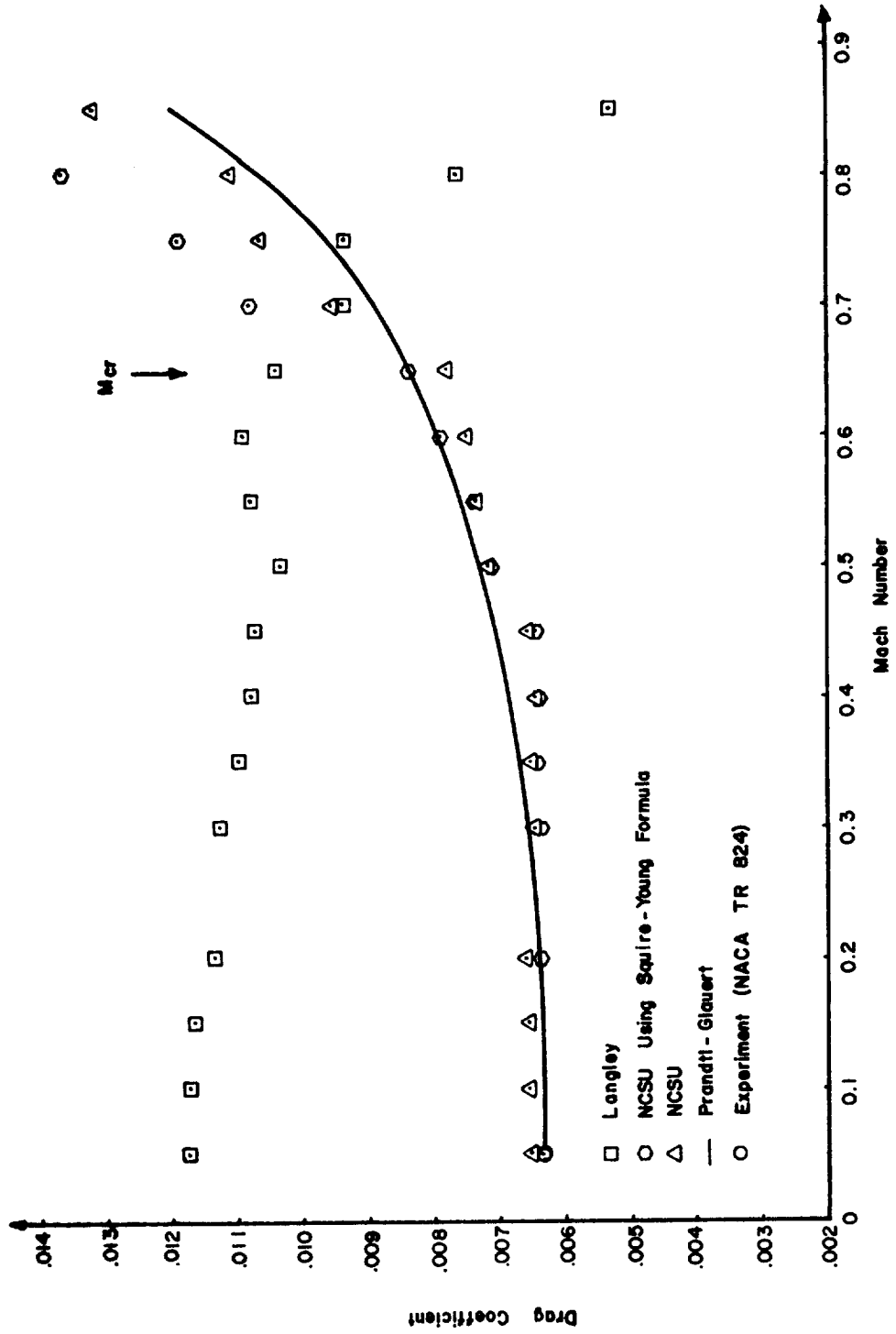
Figure 36. Predictions of C_D variation with Mach Number at $\alpha = 0°$ and a Reynolds Number of 3,000,000 for the 23012 airfoil.

209

EXTENSION TO THREE DIMENSIONS

INTRODUCTION

As elaborated in the theory section of the present work, one may employ lifting line theory to find the effective angle of attack at every point along the span of straight, moderately high-aspect-ratio wings. This effective angle of attack in conjunction with two-dimensional lift and drag data may subsequently be used to calculate three-dimensional lift and drag distributions. Integrating these distributions then gives the perceived lift and drag characteristics of the complete wing. A computer program utilizing this theory to predict stall characteristics of straight wing aircraft presently exists and is published in NASA CR-1646 (Ref. 36). It was selected to serve as the basis of a program for generalizing two-dimensional data to three dimensions in order to take advantage of the very considerable effort which went into its development. The program, however, contains many features which are not needed in the present work. Only that portion of the program pertaining to the prediction of the three-dimensional lift, drag, and pitching moment of a wing-fuselage combination is of immediate interest.

Originally written in FORTRAN IV for the CDC-6600 series computer, the program required substantial changes in order to execute effectively on North Carolina State University's IBM 370-165 machine. A list and discussion of the necessary changes is given on page 223.

The program in its original version was written to predict the angle of attack at which a wing-fuselage combination stalls and to aid the designer of a new or modified aircraft in stall-proofing his flight vehicle. This feature of the program cannot be utilized in the present effort because of the inability of the current two-dimensional program to predict boundary layer separation and therefore section aerodynamic characteristics in the near-stall region adequately. As a result, all portions of the program pertaining to the stalling wing, the iterative calculation of the value of the stall angle of attack, and location of the initial stall point were deleted. The resulting program is therefore applicable only to the linear portion of the lift curve.

Since the two-dimensional program treats only unflapped configurations, that portion of the program concerned with prediction of three-dimensional characteristics of flapped airfoils was deleted. Major reductions were also realized in the storage of airfoil data files. Tables of the two-dimensional aerodynamic characteristics were stored in the form of sets of polynomial coefficients with the Reynolds number inherent in the root and tip data sets in order to reduce input-output time and eliminate the need for nine peripheral storage devices which in the original program were used to store the airfoil data in a detailed table-look-up form. (The tables were constructed from the experimental data in NACA TR 824.)

In this modified form, the NCSU program, called FUNC, operates totally within the machine core. It will produce a three-dimensional table of lift, profile drag, induced drag, total drag and moment coefficients for

nine angles of attack (see Figure C-3) in six seconds with sixty thousand bytes of core, considerably less than the 60 seconds and 104 thousand bytes plus ten peripheral storage devices required by the original program, STALL. The specific changes are discussed in detail below.

The only obvious danger in using this fast and efficient polynomial coefficient technique is that the fit may exceed some tolerable level of error. A least squares routine, FITIT, has been provided to determine the polynomial coefficients. With it the fit error encountered on smooth two-dimensional data, as measured by the average mean square deviation, was normally between 10^{-3} and 10^{-5}, certainly less than three percent of the aerodynamic coefficient value. This may be seen in Figure B-3 which gives the fit coefficients and plots of fitted curves for the various aerodynamic coefficient functions of a 23012 airfoil. Although the fits provided by FITIT have always been within an acceptable tolerance for all runs performed to date, it is suggested that one always check the plot of the curve fit before proceeding to supply FUNC with the fitted data. FITIT performs a run of curve fits for the two-dimensional aerodynamic coefficients at nine angles of attack in eight seconds and fifty-four thousand bytes of core storage.

In order to insure that this fit error and the methods employed in modifying the original program were valid, a test case was run which compared the three-dimensional lift and drag produced by STALL using experimental two-dimensional data and the three-dimensional lift and drag as predicted by FITIT and FUNC using experimental two-dimensional data. The results of the test case is shown in Figure 37. This drag error noted is directly attributable to the inability of FITIT to fit the "drag bucket" of these airfoils using only a fourth degree polynomial. Generally, this routine as used with the two-dimensional predictions of the previous chapter do not encounter these "drag buckets" and, as mentioned earlier, operate with much higher accuracy.

Additional test cases were run to compare the three-dimensional data produced by experiment and that produced by the FITIT-and-FUNC-combination using two-dimensional data supplied from the program of the previous chapter. Results of these tests are reported below. Generally, the errors encountered were less than five percent for lift and eight percent for drag (Figure 38 through Figure 40) when compared with experiment. The experimental data and geometric configuration used in these calculations were taken from Reference 78. A drawing of this planform is given in Figure 41.

In the following the reader will find a brief explanation of the theory upon which the programs are based, and a discussion of the modifications implemented in the NCSU program.

213
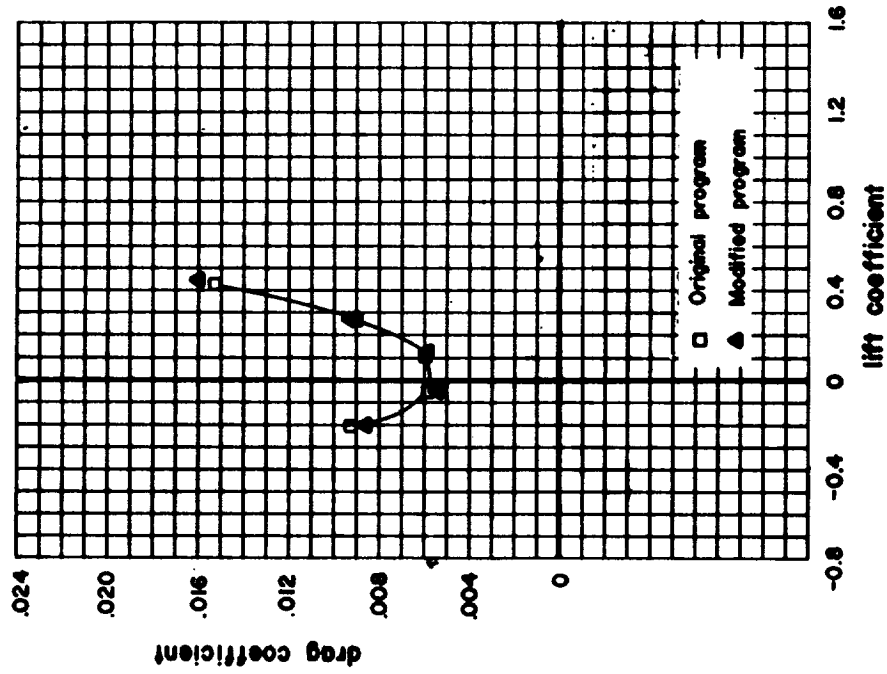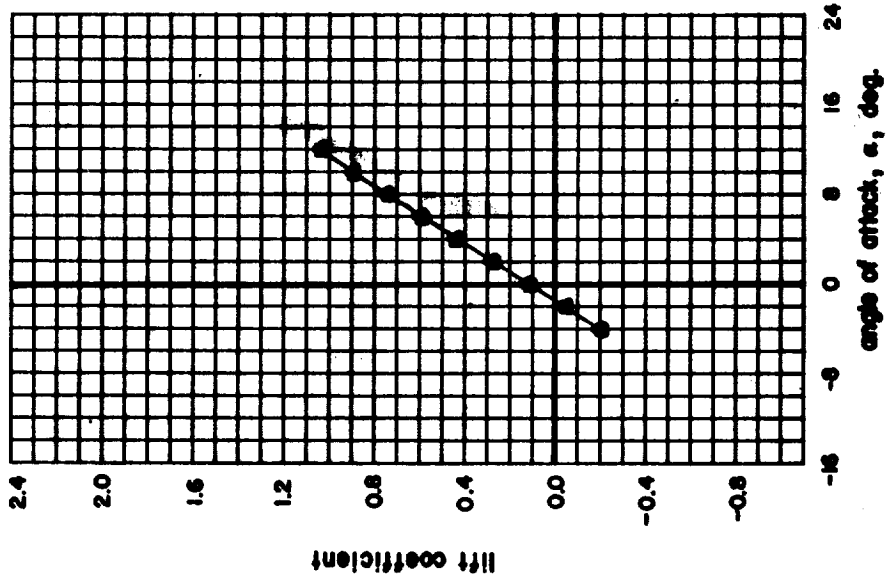
Figure 37. Comparison of original and modified program three-dimensional coefficient values for the 64218 root-64412 tip case.

214
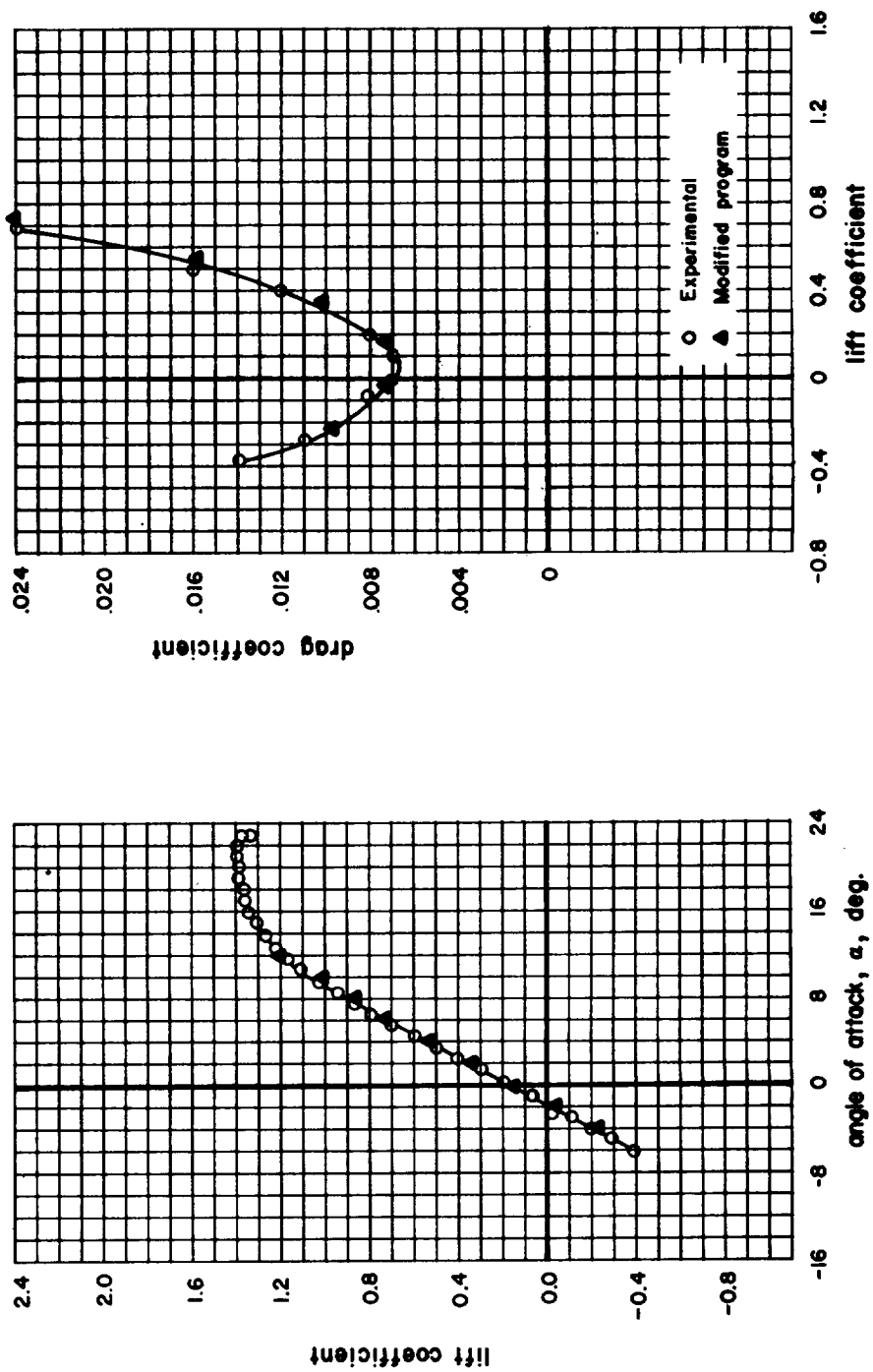
Figure 38. Comparison of experimental and calculated three-dimensional coefficient values for the 64420 root-64412 tip case.

215
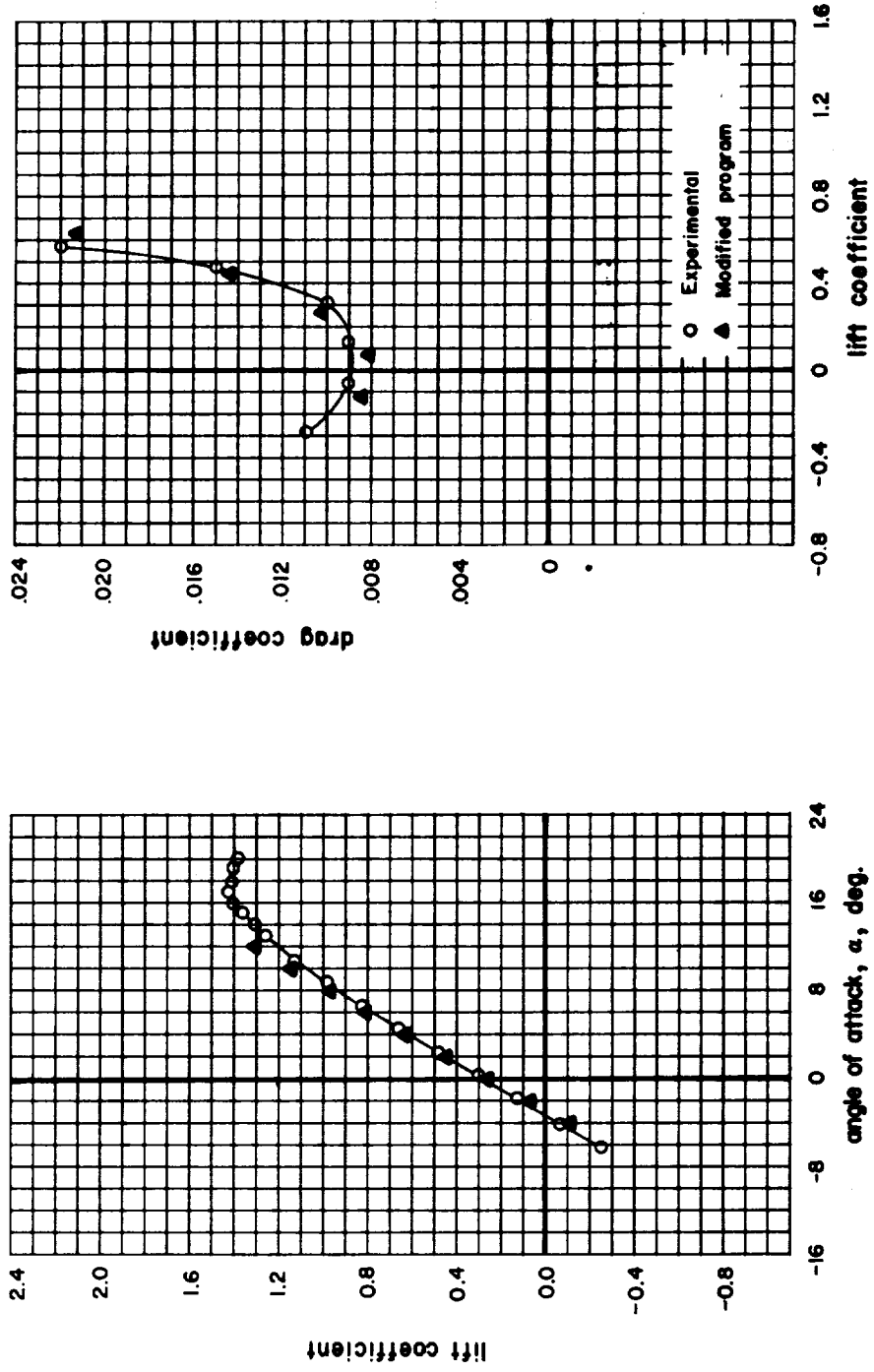
Figure 39. Comparison of experimental and calculated three-dimensional coefficient values for the 4420 root-4412 tip case.
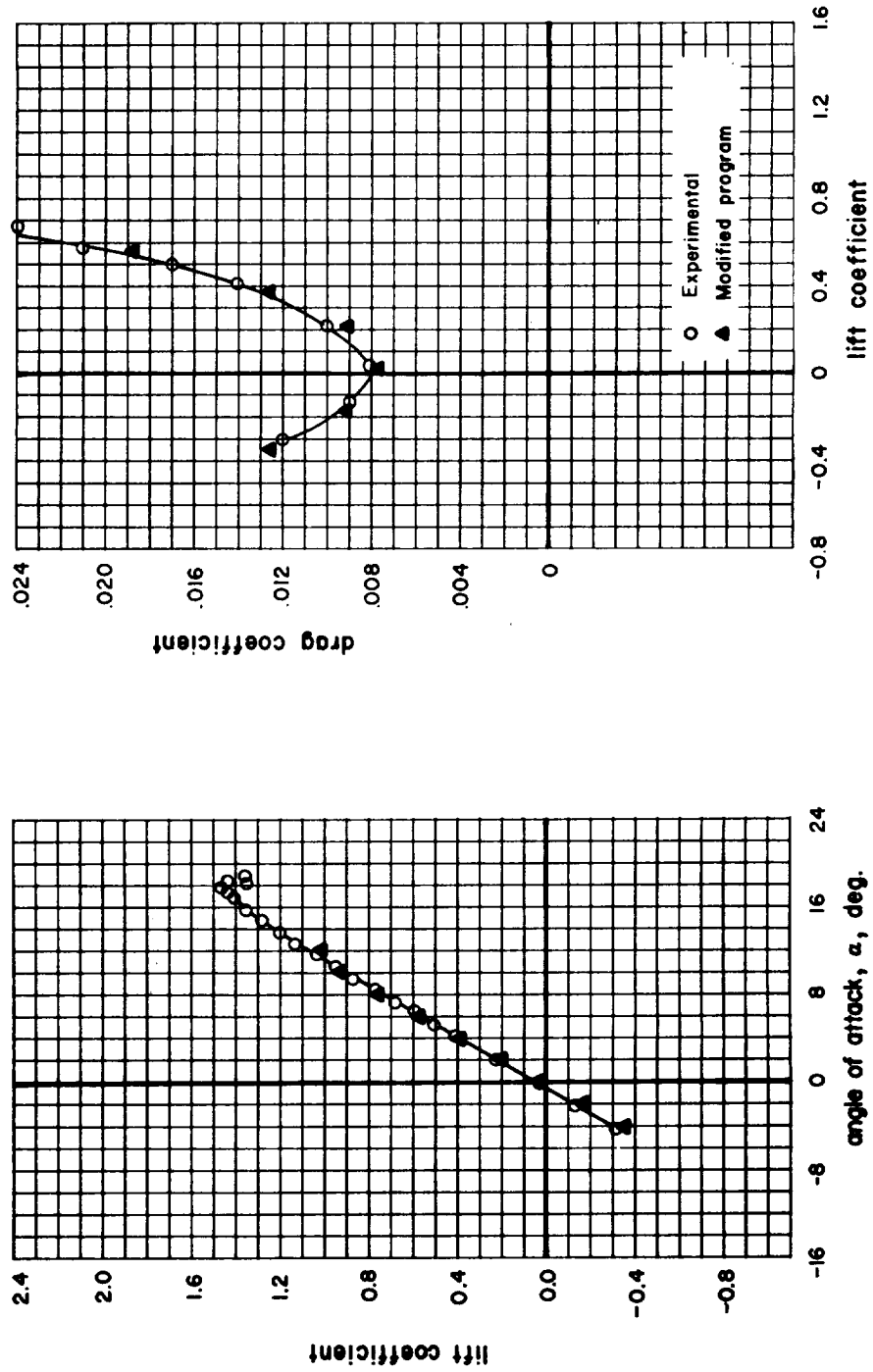
Figure 40. Comparison of experimental and calculated three-dimensional coefficient values for the 23020 root-23012 tip case.
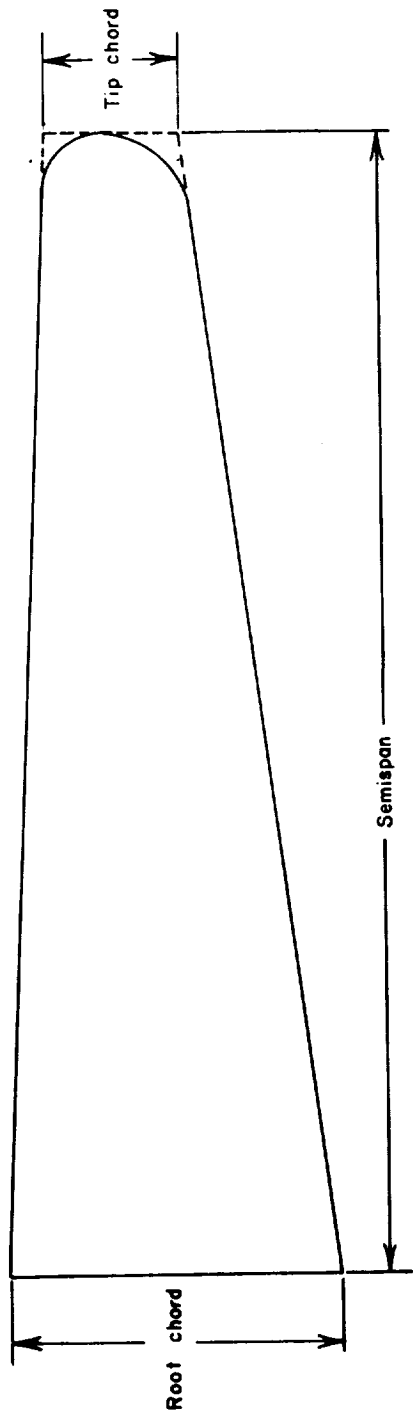
Figure 41. Planform of the three-dimensional test cases from Reference 78.

GENERAL PROGRAM THEORY AND OPERATION

The mathematical technique for extending two-dimensional data to three dimensions, presented in detail in the theory section of this report, is performed by a modified version of the program (listed in NASA CR-1646) known as STALL. The following is a description of the basic logic of the modified version, FUNC, and its support program FITIT. FUNC is supplied with two-dimensional data in the form of polynomial coefficients by a special program called FITIT. Two-dimensional data (experimental or calculated by two-dimensional programs in the previous chapter) for a specific airfoil is generally given in the form of tables of lift coefficient versus angle of attack, drag coefficient versus lift coefficient, and pitching moment versus lift coefficient for a given Reynolds number. FITIT is designed to take these tables (see Figure B-2) and produce polynomial coefficients (see Figure B-3) for this specific airfoil by a least squares curve-fitting technique (see Figure 42). The polynomial coefficients are then punched into the form acceptable to FUNC. After sufficient data sets have been generated in this fashion, the execution of FUNC is ready to be initiated.

At this point the perceptive reader might suspect that FITIT should be incorporated as a preliminary routine within the program FUNC. This was not done because: (1) The fits should always be checked for intolerable error; (2) The core requirements of FUNC would probably be extended to an intolerable level; and (3) Once a set of data for a given airfoil is run and polynomial coefficients obtained, the set need not be run again, allowing the engineer to run many geometric variations within FUNC without recalculating these polynomial coefficients each time.

As can be seen by the flow chart in Figure 43 operation of FUNC is initiated by reading in (see Figure C-2) all geometric parameters associated with the configuration under consideration. Appropriate polynomial coefficient data are then read in* and calculation of the geometric quantities associated with the transformation from the u to the $\bar{u}$ plane is initiated. If the option to read in irregular geometric shapes has not been initiated, the program proceeds to calculate the chord, thickness, camber, geometric twist and Reynolds number distributions for a wing with linear taper in both chord and thickness. Next, the multipliers β_{mk} from Equation 105, the multiplier employed in Simpson's rule enclosed in brackets in Equation 117 through Equation 120, and the inverted G matrix of Equation 114 are calculated. At this point the values of body angle of attack are read, the first value selected and a case heading printed. After a first approximation to the lift distribution has been estimated by Equation 115, the basic iterative loop searching for a convergent lift distribution is entered.

---

* Note from Figure C-2 that the thickness ratio of each airfoil data set has been inserted so that FUNC will have explicit knowledge of its value.

The first step in the search for convergence is to calculate the corresponding values of induced angles of attack and determine the effective wing angles of attack in the real plane. Now the equivalent angles of attack from Equation 116 are computed for use with the two-dimensional data. Using the section data, the values of lift coefficient corresponding to these equivalent angles of attack are obtained. If the value of $C_{\ell}c/\bar{b}$ using these values of lift coefficient is not sufficiently close to the initial estimate of $C_{\ell}c/\bar{b}$, the correction factor given by Equation 114 is calculated and added. The process discussed in this paragraph is followed until the calculated values are sufficiently close to the previous values of $C_{\ell}c/\bar{b}$.

Having obtained the lift distribution, section values for the profile drag coefficient, the induced drag coefficient, and the pitching moment coefficient are obtained. These are then integrated using Equation 117 through Equation 120.

At this point in the execution, another value of body angle of attack is assumed and the program reenters the search for the convergent lift distribution associated with this body angle of attack. The program continues in this cycle until a body angle of attack of 99.0 is encountered at which time control is returned to the beginning of the program and the next set of geometric parameters is read.

220
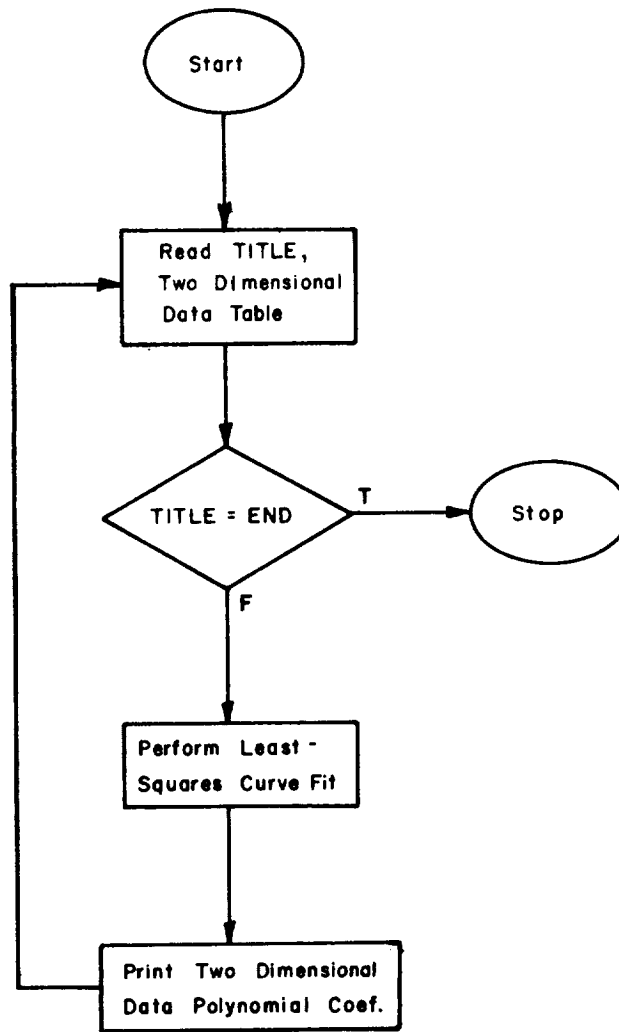
Figure 42. Flow chart of major logic in FITIT.
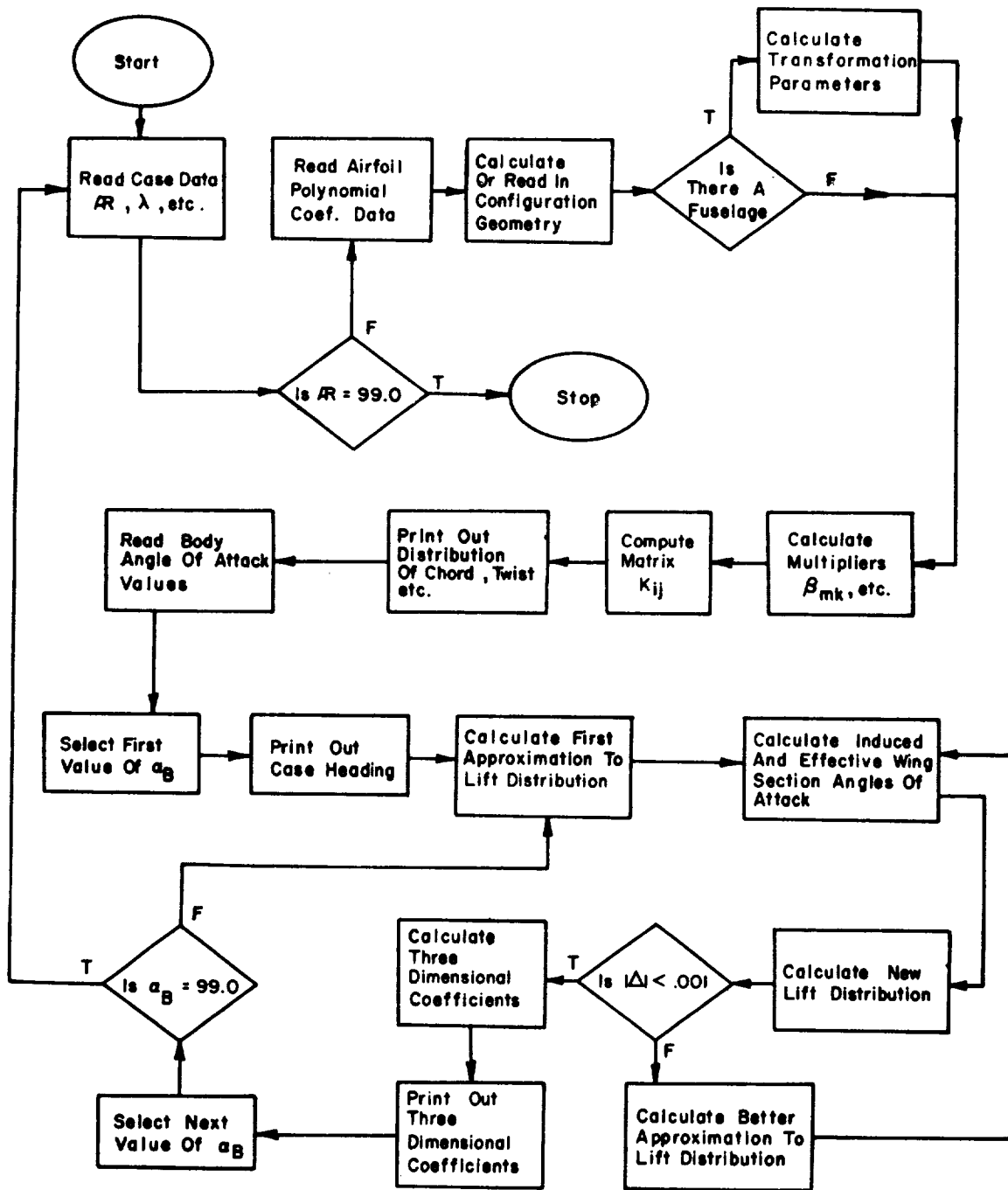
Figure 43. Flow chart of major logic in FUNC.

MODIFICATIONS REQUIRED TO ADAPT THE CDC "STALL" PROGRAM

TO IBM EQUIPMENT

The following changes were necessary in the original program STALL as published in NASA CR-1646 in order to insure effective execution on the NCSU IBM 370-165:

(1) Eliminate the overlay structure by changing programs STALL, ONE, TWO, and THREE into subroutines STALL, ONE, TWO, and THREE.

(2) Run with NOSUBCHK option on Fortran IV G-LEVEL compiler.

(3) Set the variable INNOW equal to zero after logical unit numbers are set in routine MAIN.

(4) Set ATMP (vector) to zero at statement 40 in BRIDGE.

```
40   KGO = 1
     ATMP(1) = XX(1)
     DO 41 J = 2,NP
41   ATMP(J) = 0.0  .
```

(5) Insert the following statement immediately preceeding the statements setting logical unit numbers in MAIN.

```
ACOS(X) = ARCOS(X)
```

(6) Prevent illegal entry into a DO loop in LOOK by inserting at statement 60,

```
60   LOCR = 3
61   IF(REYN-A(LVL,1,LOCR)70,70,610
```

and statements 600 to 620 should read

```
600   GO TO 630
610   LOCR = LOCR+1
      IF(LOCR.LE.MAXC)GO TO 61
611   IE = 1
      GO TO 630
620   REYN = 999.0  .
```

(7) Enclose the variable YY in the argument list of BRIDG in slashes to insure call by name.

GENERAL PROGRAM MODIFICATIONS

It was mentioned in the introduction to this chapter that the original program STALL consisted of approximately 1700 executable statements. This section serves as a guide to the systematic dismantling and conversion of the program to its modified form. The program as supplied to NCSU operated in the overlay mode with a main overlay and three additional overlays of the same sublevel. Since total operation within the core of the IBM 370-165 is less expensive than the time to call in overlays, the program was taken out of the overlay mode and compiled and executed as a unit. Some IBM-CDC incompatibilities noted in the previous section were then removed in order to execute the program on the NCSU 370-165. These changes resulted in a significant reduction in execution time at the expense of increased core requirements. As mentioned previously this trade is quite cost-effective.

A second major modification occurred when portions of the program unneeded for the present investigation were removed. These included all portions associated with stall calculations as well as those portions having to do with flaps. These routines are THREE, MAIN3, and MAIN5. See Table 2. These changes reduced the number of executable statements by 550 to approximately 1150, a substantial reduction.

ORIGINAL PROGRAM (STALL)	MODIFIED NCSU PROGRAM (FUNC)
STALL	
MINV	MINV
DAGET	
AAA	AAA
ZZZ	
SSS	SSS
TERP	TERP
SETSW	SETSW
DATSW	DATSW
AERDA	
BRIDG	BRIDG
ARC	
LOOK	
ONE	
MAIN	MAIN
MAIN1	MAIN1
TWO	
MAIN2	MAIN2
MAIN4	MAIN4
THREE	
MAIN3	
MAIN5	
	FUNX
	FUN

Table 2. Comparison of routines contained in the
original and modified programs.

A further major reduction was realized by making the Reynolds number of interest inherent in the two-dimensional data. An in-depth look at what is involved with this assumption may be found in the next section of this chapter. Use of this assumption, however, resulted in a decrease in executable statements by 250 to approximately 900.

Still another reduction was made by changing the location of data storage. In addition to saving a tremendous amount of input-output time by expurgating the nine peripheral storage devices associated with the table look-up procedure and storing the polynomial coefficients within core, the move reduced the number of executable statements by 100 to a remaining 800.

Reductions to the existing less-than-700 executable statements can be attributed to the clean-up of unused variables and to common block reorganization.

CHANGES CONCERNING REYNOLDS NUMBER

In order to locate the two-dimensional value of a given aerodynamic coefficient, say C_ℓ, in the original program, three separate interpolations were made. Linear interpolation was performed first for the required value of Reynolds number, then for thickness ratio, and finally for camber. Because of the amount of data being searched, an investigation into a possible deletion of one of these variables was initiated. Deletion of any one of the variables would cut the number of interpolations by forty percent. Since the present investigation had excluded the stall region because of the inability of the two-dimensional program to predict boundary layer separation effects adequately, a reduction in dependence of the results on Reynolds number was established. Changes in Reynolds number at low angles of attack produce very little variation in lift coefficient and only slight variation in drag coefficient (Ref. 79). These variations were so slight that it was decided to either calculate the two-dimensional characteristics for both tip and root airfoil families at the Reynolds number of the mean aerodynamic chord or, at the most, enter the characteristics at only the root and tip values.

Error encountered by the use of the mean aerodynamic Reynolds number is low because of the averaging effect of the camber interpolation and eventual integration. Error encountered by the use of root and tip families at their respective Reynolds number should be even less since this effort gives the coefficient data an implicit relationship in Reynolds number.

The calculation of the spanwise Reynolds number distribution was left in as a check for the designer. It is suggested that designs using high taper ratios resulting in wide Reynolds number variation employ the root-tip implicit method. Removal of the Reynolds number interpolation allowed the deletion of two routines, ARC and LOOK, from the original program (see Table 2).

226

CHANGES CONCERNING AIRFOIL DATA STORAGE

With the transfer of the Reynolds number to an implicit dependence, half of the minimum data sets were no longer necessary; furthermore, the remaining portions of the data sets could be stored in core, resulting in a substantial reduction in operating time since the comparatively long time required to transfer data from tape or disc files could be eliminated. Adopting this philosophy removed nine external data files, decreased the execution time further, and eliminated approximately 100 executable statements among these being the two routines DAGET and AERDA of the original program.

A further reduction in execute time was realized by storing the two-dimensional data not as tables of points but as polynomial function coefficients and thus eliminating tabular interpolation.*

A schematic diagram of the storage and evaluation procedure is given in Figure 44. Note that for a geometric configuration with root and tip families different, evaluation of any given aerodynamic coefficient has been reduced to four functional evaluations, two thickness interpolations, and one final camber interpolation.

---

* Note from Figure C-2 that the angle of attack versus coefficient of lift polynomial was added in order to predict the angles of zero lift without solving for the zeros of a fourth degree polynomial function.
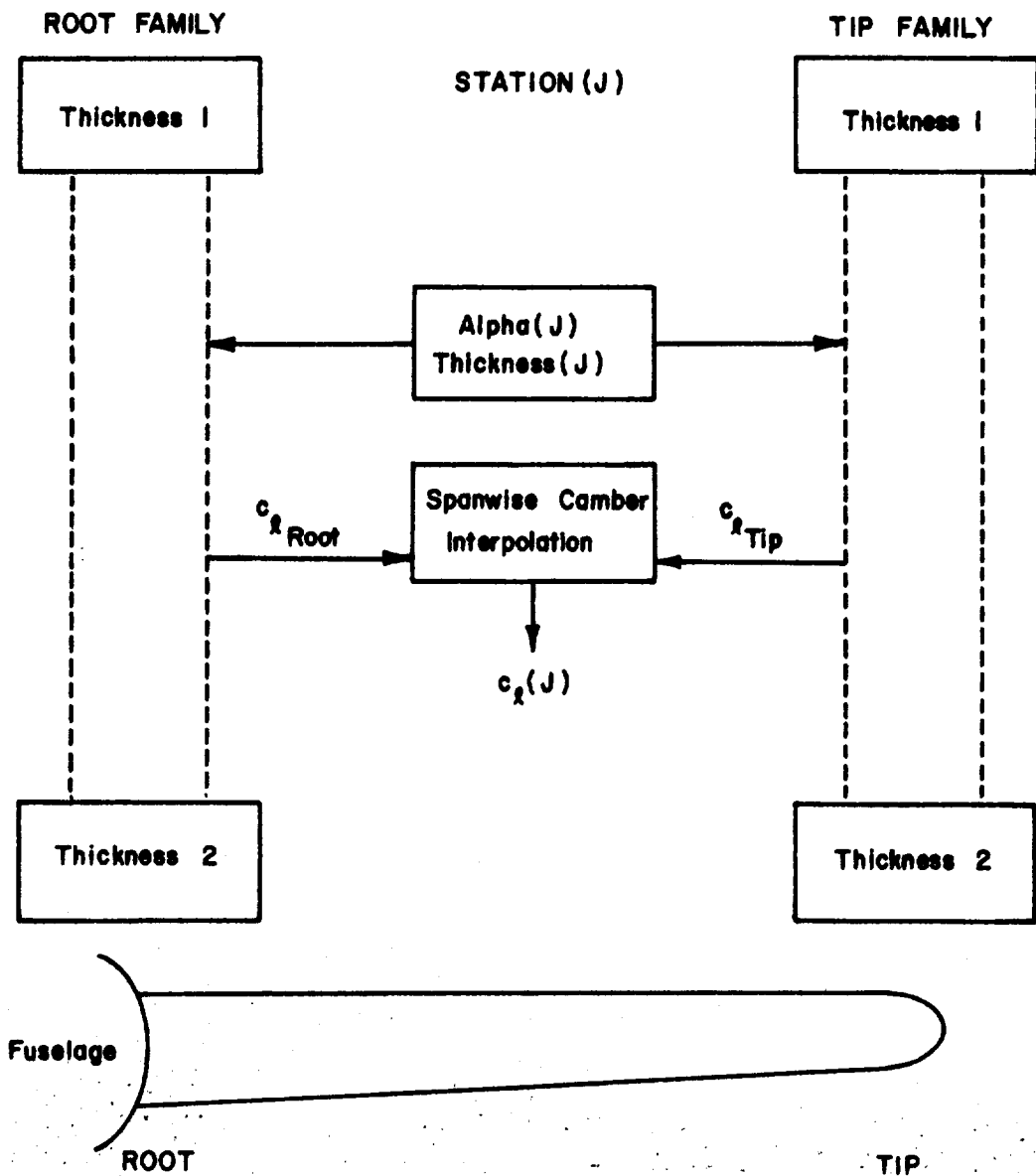
Figure 44. Schematic representation of the modified
data look up procedure.

DISCUSSION OF PROGRAM RESULTS

Results of computations of the three-dimensional lift and drag of a wing using two-dimensional section data as input are shown in Figure 37 for both the reduced NCSU program and for STALL. Agreement with STALL is very good for the lift and quite reasonable for the drag, especially when one considers that drag "buckets" found experimentally on 6-series airfoils and used in STALL cannot be fit satisfactorily with just a fourth order polynomial. This fourth order representation was judged adequate, however, because the airfoil aerodynamic characteristics program inherently predicts relatively smooth drag curves. Since such results were always to be the input to the three-dimensional program, the authors saw no need to increase the program complexity.

A comparison of the results of predictions of the entire computation procedure with experiment is shown in Figures 38, 39, and 40. The lift data for the 6-series airfoils, shown in Figure 38, shows excellent agreement up to a C_L value of 1.2. The drag data is quite acceptable to a C_L value of 0.8. The wing data for the four digit airfoils indicates that, particularly for lift coefficients above 0.4, the viscous effects on these thick airfoils are not properly accounted for. The same behavior is noted for other thick airfoils of the same family. (See Figures 13, 14, and 15.) The prediction is, however, much better on the wing constructed of 230-series airfoils (Figure 40). Here, the lift results agree well with experiment for $C_L < 1.0$ and the drag predictions are in reasonable agreement for $C_L < 0.6$. It seems reasonable to conclude therefore that for $C_L < 1.0$, the program will provide wing lift data as reliable as the two-dimensional data supplied as input. Wing drag predictions made by the program also seem to be as reliable as the section characteristics used as input, at least for $C_L < 0.8$. Thus, for low-to-moderate lift coefficients and moderate-to-high-aspect-ratio, unswept wings this procedure is far simpler computationally and of equal accuracy as compared with the more complex vortex lattice or lifting surface methods.

PROGRAMS FOR THE CALCULATION OF BODY

AERODYNAMIC COEFFICIENTS

INTRODUCTION

The problem of accurately estimating the lift and drag coefficients of arbitrary three-dimensional bodies in a rigorous fashion will certainly be recognized as an extremely difficult task. The word arbitrary, for example, implies that the routine or procedure for estimating these coefficients must be general enough to handle a variety of bodies including bluff ones to be useful to those to whom this work is directed; yet the procedure cannot require an excessively large number of computations. In order to yield the most reasonable solution within the constraints of small computer run times and small computer storage requirements the present authors chose to use the following procedure:

(1) A program to calculate the inviscid flow about arbitrary three-dimensional bodies was obtained from the Naval Ship Research and Development Center at Bethesda, Maryland. One important reason for selecting this program was the fact that it was equipped with the capability of computing on-body streamlines. Another was the fact that it was already limited to bodies alone. Two other programs which the authors acquired in the process of developing this procedure would have required removing the wing characteristics calculation portion of the programs.

(2) The program was reduced in size as much as possible and specialized to calculate the inviscid flow over bodies at zero angles of attack and sideslip. (As received, it permitted one to calculate the inviscid flow field with onset flow components along all three axes.)

(3) The skin friction drag was estimated by applying a two-dimensional boundary layer technique to the on-body streamlines and integrating the resultant wall shear over the body surface.

(4) The viscous form or pressure drag was estimated by considering the body wake to be representable within the framework of an inviscid flow solution in much the same manner as the complete airfoil solution was achieved.

The reasons for choosing this procedure are quite straightforward:

(1) It relies heavily on existing programs or procedures whose applicability and computational problems have been well charted. The time and effort required for program development is thus reduced to a minimum.

(2) The procedure is step-like so that if necessary it can be done in pieces on a small machine with limited storage capacity. In contrast, it would be very difficult to segment an attempt to solve the general three-dimensional Navier-Stokes equations for a complex boundary shape.

232

Theoretical justifications for some of the steps in the procedure are given at length in earlier sections of the present work.

Of course, in order to obtain shorter running times and smaller storage requirements, it has been necessary, as is always the case, to limit flexibility and employ certain assumptions which are not uniformly valid. The restriction to $\alpha = 0$ cases has already been mentioned. In addition, the boundary layer routine used is a two-dimensional one so that stagnation point flows are not well described nor are cases where cross flows are present. Further, the boundary layer procedure does not permit one to consider flow separation, $i.\ e.$, it assumes separation does not exist. Although body wakes are included in the analysis it has been necessary to assume, in the absence of an understanding at the proper criteria, that they all develop according to the same simplified rules. As a result of these limitations, some discretion should be exercised in applying the results of the computation and in selecting cases for computation so that the governing assumptions are not greatly violated.

The discussion below outlines the computational procedure employed. The original program is described in general so as to provide the reader or user with a reference for understanding the modifications made to it. The modifications are then discussed in detail. Also discussed in detail are the locally developed elements of the program, in particular the boundary layer and wake-body computation procedures. We begin with a presentation of a very effective means to identify errors in the input data. The section concludes with a discussion of the results obtained using the program to compute the drag of three bodies: a sphere, a 3:1 prolate spheroid, and a Cessna 182 aircraft fuselage.

233

SPECIFICATION OF INPUT DATA WITH VERIFICATION BY PLOTTING

When calculating the potential flow over a three-dimensional body, one is confronted with the problem of how to specify the shape of the body surface. Obviously, one would like to have an analytical expression for the body surface; however, for general three-dimensional bodies practical analytical representations are usually impossible to obtain. The general practice is therefore to approximate the body surface by a large number of quadrilateral-shaped panels defined by a finite number of points in space; each point is presumably exactly on the body surface. If a computer is to be used to solve for the potential flow over these bodies, someone must input the three coordinates of each point; this is a laborious and error-prone task. In addition, each point must be indexed in such a way that the four corner points of each panel are defined in a clockwise fashion. Checking the input for errors is also a tedious process since detecting errors by simply scanning a list of the input data points is extremely difficult. To minimize such errors one would like a simplified, orderly procedure for inputting the data, and an effective procedure of finding errors before lengthy potential flow calculations are made. Probably the simpliest procedure for inputting the data is to specify the shape of the body cross section at various stations along the longitudinal body axis. Among the most effective procedures for detecting input errors is to graph the points as viewed from various directions (See Figure 45).

While the above is true for any arbitrary body, in this report we are concerned only with aircraft fuselages which have a plane of symmetry along the longitudinal axis. In 1970 NASA, aware of the problems of specifying and checking numerical data, developed a computer program to generate the necessary instructions for automatic plotting of an airplane model in numerical form (Ref. 113). The plotting capability of this program along with its simplified data input procedure makes it ideal for producing a final, verified numerical data set describing an aircraft fuselage. The program also has the capability of displaying the complete aircraft configuration including wings, pods, fins, and canards. Using it one may draw three-view and oblique orthographic projections, as well as perspective projections of an airplane. The program even has the capability of plotting stereo frames of the aircraft suitable for viewing in a stereoscope. Because of its versitility the authors chose to use this NASA program to verify aircraft input data. A copy of the program, written for the CDC 6000 computer, was obtained from NASA Langley Research Center and then modified so that it would run on the IBM 370-165 computer at N. C. State University. User instructions, plotting software modification procedures, a program listing, and several sample output plots for the modified plot program are given in Appendix D.

234
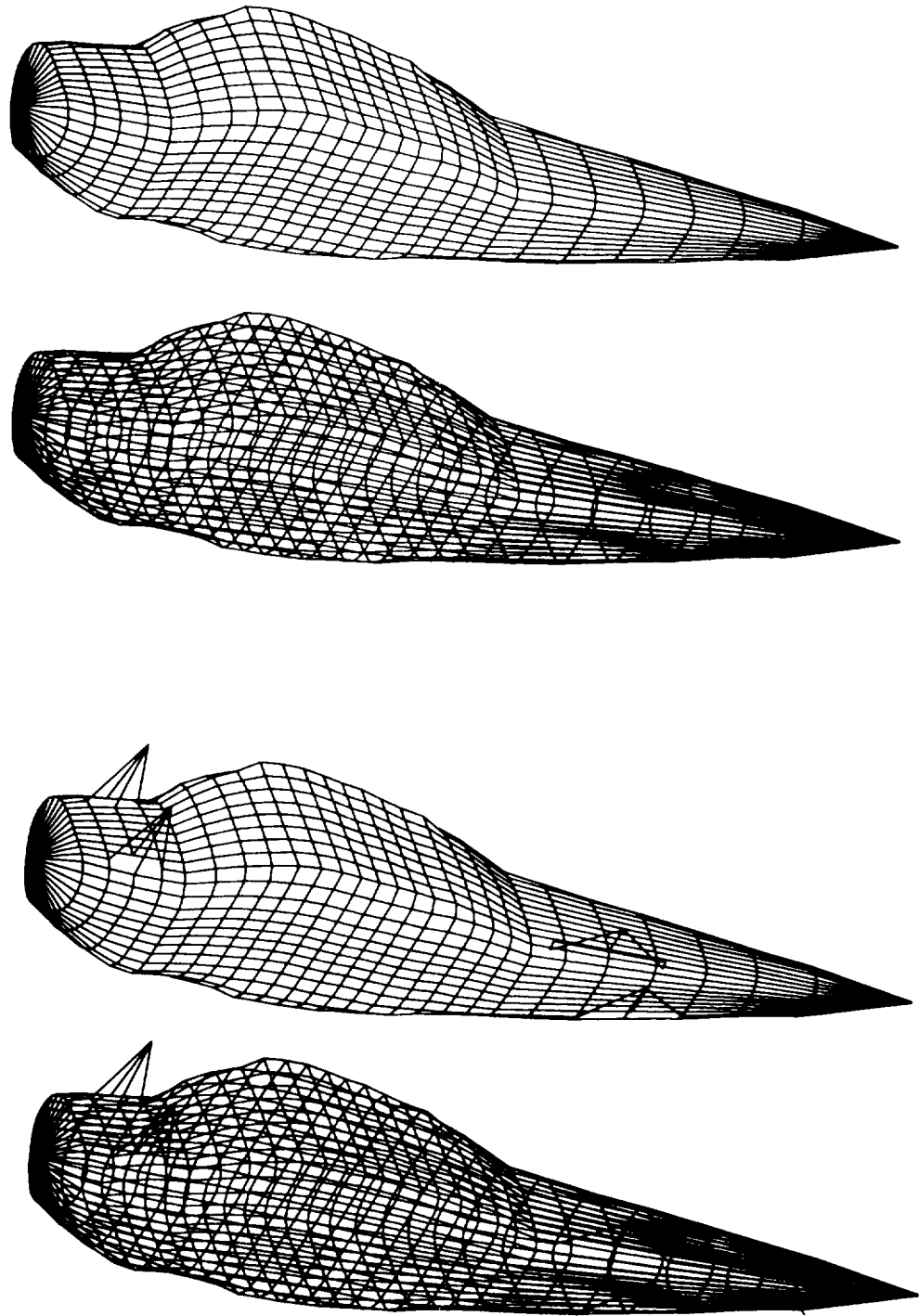
Figure 45. Example of a correct and an incorrect data
set for the Cessna 182 fuselage.

It should be noted that most of the modifications made to the original program were those necessary to enable the program to run on the NCSU IBM computer. These modifications included:

(1) changing the DECODE form of program input to ordinary READ input

(2) removing the OVERLAY procedure used to reduce program size during execution at NASA

(3) adding subroutines which could call IBM software plotting instructions which were equivalent to the specified CDC software instructions[†]

(4) assigning variable names to the input and output file unit numbers as well as to data storage file unit numbers (the user must therefore only specify the appropriate file unit number required at his computing facility).

For more information concerning the basic program the reader is advised to consult Reference 113 which provides a detailed description of much of the program as well as flow charts of each important program section.

While the input to the PLOT program is as concise as possible, the input to the NCSU BODY program (as well as the XYZ potential flow program) is both lengthy and time consuming since three coordinates and two indexes are specified on each point input card. Accordingly, a program which converts a data set for the PLOT program into a properly indexed data set for the NCSU BODY program was developed; thus, once the plot data set is verified, a correct data set for the NCSU BODY program may be generated. As discussed in General Program Theory, one criteria for obtaining a reasonable potential flow solution is that adjacent body panels must generally have areas which differ by less than 50 percent. Therefore, the CONVERT program was also designed to compute the area of each body panel and display these areas in an orderly fashion. Also, the program displays the ratio of the area of each panel to the area of the panel below it and the panel to its right. These ratios simplify the procedure of checking panel areas to see if they meet the criteria described above. It should be noted that while the CONVERT program produces a data set for the fuselage, it will accept the input of a data set for a complete aircraft configuration and ignore the unnecessary information. The program input is obviously the same as the input for the PLOT program and its description is therefore not repeated in Appendix E; however, the program listing and a

---

[†]Since plotting software is different at almost every computing facility, the user _must_ either provide the original CalComp plotting software for which the program was designed or provide three _equivalent_ dummy subroutines as was necessary at the N. C. State faillity. See the Plotting Software Modifications section of Appendix D for detailed instructions.

236

sample output depicting the capabilities described above are given in the appendix.

By inspecting Figure D-1 and Figure F-1, the reader will discover that the body orientations, with respect to the input axis system, are different for the PLOT program and the NCSU BODY program. In order to overcome this difficulty the CONVERT program also contains the instructions required to invert the body with respect to the X-axis and approximately center the body about the origin. These instructions are necessary since the flow is in the direction of the negative X-axis for the NCSU BODY program.

In order to calculate the boundary layer over the body the NCSU BODY program must have the properties of the flow field as well as the body geometry. The CONVERT program was designed specifically for any plot data set as described in the original PLOT program, and these data sets describe only the body shape. It is therefore necessary to insert a flow field parameter card into the data set created by the CONVERT program before the data set is input into the NCSU BODY program. The card (described in Appendix F), which is inserted behind the first card (identification card), specifies the free stream velocity, density, and kinematic viscosity of the flow field as well as the reference area upon which the coefficients are based and an output control parameter. Failure to include this card in the NCSU BODY program input will result in an invalid program execution.

237

GENERAL PROGRAM THEORY FOR INVISCID BODY PROGRAM

The XYZ potential flow program, obtained from the Naval Ship Research and Development Center (Ref. 94) is a computer program for the computation of irrotational, incompressible potential flow about three-dimensional bodies of arbitrary shape. The solution method is essentially that developed by Douglas Aircraft Company in References 23 and 83. For a detailed description of the program theory the reader is advised to consult these references; however, an excellent brief description of the method is given in Reference 94, major portions of which are excerpted below.

The body surface is approximated by a set of plane quadrilaterals, and the solution is constructed in terms of a source density on the surface of the body. Based on the assumption that the source density is constant on each quadrilateral, a system of algebraic equations is used to approximate the integral equation for the source density over the body. The source density in each quadrilateral is chosen so that the normal component of the velocity is zero at one point in the quadrilateral. The matrix equation is solved by a simultaneous displacement iteration scheme with a two-eigenvalue extrapolation procedure to speed up convergence.

The XYZ program received at N. C. State is divided into five basic sections. Section 1 reads the input cards, computes the descriptive parameters for each quadrilateral, and checks for errors. Section 2 computes the matrix elements in the equations for the source density and the velocity for points on the body. Section 3 solves the matrix equation for the source density. Section 4 computes and edits the velocity and pressure coefficient on each quadrilateral. Section 5 computes the coordinates of the streamlines on the body surface.

The program actually solves a problem involving a stationary, three-dimensional body in a moving ideal fluid. The fluid is assumed to have a uniform velocity at infinity ($\vec{V}_\infty$) which is parallel with the x-axis of the body. The velocity potential ϕ satisfies the following equations:

$$\nabla^2 \phi = 0 \qquad \text{in the fluid} \qquad (1)$$

$$\frac{\partial \phi}{\partial n} = 0 \qquad \text{on the surface of the body} \qquad (2)$$

$$\phi = -x \cdot V_{\infty_x} - y \cdot V_{\infty_y} - z \cdot V_{\infty_z} \qquad \text{at infinity} \qquad (3)$$

238

where n indicates the direction normal to the surface. A solution to these equations is constructed in the form of a source density (S) on the surface of the body,

$$\phi(p) = \iint_{\text{Body Surface}} S(q) \frac{1}{r(p,q)} dA_q - x \cdot V_{\infty_x} - y \cdot V_{\infty_y} - z \cdot V_{\infty_z} \qquad (4)$$

where $r(p,q)$ is the distance between the point p at which we are interested in finding the potential and some other point q on the body, and A_q denotes the area of the quadrilateral containing the point q. Note that Equations (1) and (3) are satisfied by ϕ as defined by Equation (4). The boundary condition on the body, Equation (2), can be applied to obtain an equation for the source density (S).

$$0 = \frac{\partial \phi}{\partial n_p} = -2 S(p) + \iint_{\text{Body Surface}} S(q) \frac{\partial}{\partial n_p} \frac{1}{r(p,q)} dA_q$$

$$(5)$$

$$-n_{px} \cdot V_{\infty_x} - n_{py} \cdot V_{\infty_y} - n_{pz} \cdot V_{\infty_z}$$

Since the surface of the body is approximated by a set of plane quadrilaterals which are generated from input points (See Figure 46), in the limit the above equation requires that the body surface be represented by an infinite number of panels at each of which the flow normal to the surface is zero. It is important to obtain satisfactory results with a finite number of panels and to properly size and position these panels on the body surface. In Reference 23 Hess and Smith state that the proper distribution of elements over the body surface is largely a matter of intuition and experience. Panels should be concentrated in regions where the flow properties, particularly the source density, are expected to vary rapidly. The method gives correct results for convex corners, but concave corners cause difficulty that may or may not be serious. Accordingly, they recommend that panels should not be concentrated near unrounded concave corners; but if the corner is extreme enough to require rounding, a very great concentration of panels is necessary in that region. The panel sizes also play an important role in determining the validity of the solution. Hess and Smith note that if several small panels are in the vicinity of a large one, the accuracy is that associated with the large panel. Thus, the size of panels should change gradually when going from a region of highly concentrated small panels to a region of sparsely concentrated panels. They recommend that the characteristic dimensions of a panel should usually be no more than 50 percent greater than those of adjacent elements.   239
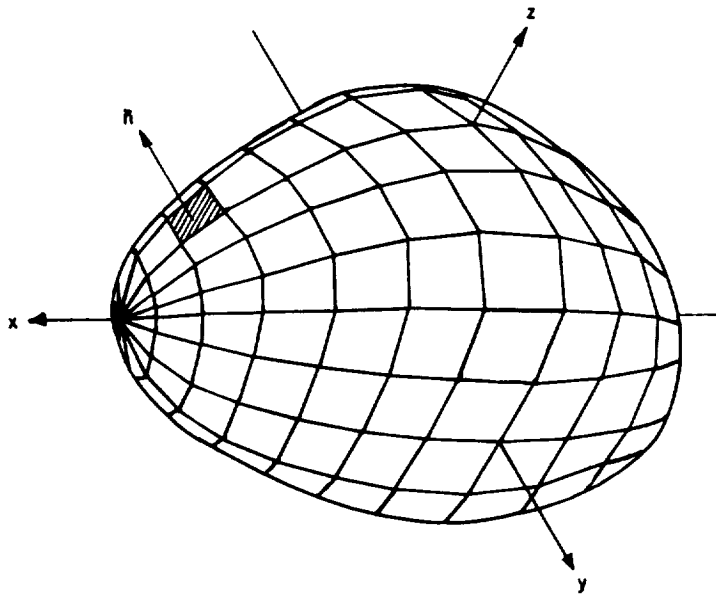
Figure 46. The approximate representation
of the body surface.

The source density is assumed to be constant in each of the body panels and is computed by satisfying Equation (5) at one point in each of the quadrilaterals. The point chosen is the panel centroid point. Thus, the integral Equation (5) is approximated by a matrix equation

$$S_i = \sum_j C_{ij} S_j + V_i \qquad (6)$$

where

$$C_{ij} = \frac{1}{2\pi} \iint_{\text{Quad. } j} \frac{\partial}{\partial n_i} \left(\frac{1}{r_{ij}}\right) dA$$

$$C_{ii} = 0, \text{ and}$$

$$V_i = -\frac{1}{2\pi} (n_x \cdot V_{\infty_x} + n_y \cdot V_{\infty_y} + n_z \cdot V_{\infty_z}).$$

240

Equation *(8)* is solved for S_j by the iterative procedure mentioned above (see Reference 83 for more detail). The velocity components at each centroid point are then computed from the following equations:

$$VX_i = \sum_j V1_{ij}S_j + V_{\infty_x} \tag{7}$$

$$VY_i = \sum_j V2_{ij}S_j + V_{\infty_y} \tag{8}$$

$$VZ_i = \sum_j V3_{ij}S_j + V_{\infty_z} \tag{9}$$

where

$$V1_{ij} = -\iint_{Quad.\ j} \frac{\partial}{\partial x}(\frac{1}{r_{ij}})dA$$

$$V2_{ij} = -\iint_{Quad.\ j} \frac{\partial}{\partial y}(\frac{1}{r_{ij}})dA,\ and$$

$$V3_{ij} = -\iint_{Quad.\ j} \frac{\partial}{\partial z}(\frac{1}{r_{ij}})dA$$

The pressure coefficient is then computed from these velocity components.

An integral over a quadrilateral is evaluated by one of three methods, depending upon the ratio of the distance of the ith point from the quadrilateral to the maximum dimension of the quadrilateral. If the ratio is greater than 4.0, the quadrilateral is approximated by a monopole (as if it were concentrated at one point). If the ratio is greater than 2.0 and less than or equal to 4.0, the quadrilateral is approximated by a quadrupole. If the ratio is less than or equal to 2.0, the integrals are evaluated exactly. The approximate methods are used because they require much less time than the exact method. The evaluation of the integrals is extensively discussed in References 23 and 83.

The on-body streamlines are computed once the velocities are known at each quadrilateral centroid. The streamline computation procedure used in the XYZ program is discussed in some detail in a report to be published by Charles W. Dawson and Janet S. Dean of the Naval Ship Research and Development Center in late 1975. This procedure may be outlined in the following manner:

(1) The coordinates of a starting point within a particular quadrilateral are specified.

(2) The two points at which the streamline, passing through the starting point, intersects the sides of the starting quadrilateral are found.

(3) The intersection point which is in the upstream flow direction is retained.

(4) A search is made of the adjacent quadrilaterals to determine which quadrilateral the streamline is entering in the upstream direction.

(5) Using this quadrilateral as a starting quadrilateral the point at which the streamline leaves this new quadrilateral in the upstream direction is determined.

(6) The above procedure, steps (4) and (5), is continued until the streamline reaches the nose of the body.

(7) The upstream portion of the streamline so traced is now defined by the coordinates of its intersection points on the sides of the quadrilaterals through which it passes.

(8) It should be noted that as each point on the streamline is found, the velocity at that point and the distance from that point to the previous streamline point are calculated.

(9) After returning to the original starting quadrilateral the same procedure is used to trace the streamline in the downstream direction to the body tail.

(10) The arc lengths computed from point to point in (8) are then all referenced to the nose of the body so that the distance of any point on a streamline from the nose of the body is known.

242

GENERAL PROGRAM MODIFICATIONS

Many of the changes made in the XYZ potential flow program as obtained from the Naval Ship Research and Development Center may be classified as general program modifications applicable to the entire program rather than a particular subroutine; these general modifications are discussed below. Throughout the remainder of the text the program as received at N. C. State will be referred to as the XYZ program while the modified viscous flow program will be referred to as the NCSU BODY program.

The XYZ program required only minor modifications in order to execute on the IBM 370-165 computer at N. C. State. Four of the five separate sections of the program were changed to subroutines all of which were called by the first section which was designated as the mainline. It was also necessary to modify the input and output file numbers as well as some of the data storage files.

In the interest of saving scratch file space, several of the files in the XYZ program were eliminated or reduced in size. In some cases the information stored on these files was put into a COMMON statement and thus made available to all five sections of the program. Table 3 gives a list of both the original file numbers used in the XYZ program and, if the information on these files was not commoned or deleted, the new file numbers used in the NCSU BODY program. Variable names were assigned to the input and output file unit numbers (JREAD-input, JWRITE-output) and to each scratch file unit number used in the NCSU BODY program. The values of the unit numbers are assigned by specification statements at the beginning of the mainline of the NCSU BODY program (JREAD=1, JWRITE=3, KFILE1=7, KFILE2=8, KFILE3=9, KFILE4=10, and KFILE5=11). For installations having different input/output unit numbers and file numbers these specifications may be easily changed. The addition of a wake body to the original body required more body panels and therefore more panel geometry storage space on file KFILE1. To prevent recalculating original panel geometry, the appropriate information for the new quadrilaterals was calculated and then added to the original information by using a second file (KFILE2) for panel geometry information. The original panel geometry which was unchanged was copied from file KFILE1 to KFILE2, and the new information was then added to file KFILE2. This two-file procedure is used to prevent file READ-WRITE incompatibilities at other computing facilities.

In an effort to reduce the size of the XYZ program by specializing it to the problem of interest, several modifications were made. Since the program was to be used for bodies at zero angles of attack and sideslip, those portions of the XYZ program which calculate the contributions to the flow and pressure over the body resulting from the Y-flow and Z-flow components were removed when the NCSU BODY program was created. Thus, the free stream velocity in the NCSU BODY program was specified as -1.0 in the X-direction, 0.0 in the Y-direction and 0.0 in the Z-direction. While the potential flow calculations are correctly made using just this unit velocity, the viscous part of the program

243

requires the specification of the magnitude of the free stream velocity or the Reynolds Number in order to make the boundary layer calculations. The magnitude of the free stream velocity must therefore be specified (see last paragraph in this section).

XYZ Program	NCSU BODY Program	
File Number	File Name	File Number
1	KFILE3	9
2	KFILE4	10
3	Information Commoned	
4	KFILE KFILE1 KFILE2	7 8
5 (Input)	JREAD	1 (input)
6 (output)	JWRITE	3 (output)
7	File Deleted	
8	File Deleted	
9	File Deleted	
11	KFILE5	11
12	File Deleted	
16	File Deleted	

Table 3. Data file comparison for the XYZ and NCSU BODY Programs.

The maximum array size for the quadrilateral input arrays and geometry storage arrays was set at 650. This maximum input array size coupled with the addition of the wake body, which uses part of the geometry storage arrays, means that the user should specify the original body using less that 600 panels. The coefficients of local quadratic representation of the body surface, which were stored on a scratch file as quadrilateral geometry information in the XYZ program,

244

were deleted from file storage in the NCSU BODY program, thereby reducing the file storage space and the size of the file input array B. The WS array contained in the XYZ program was also deleted from the NCSU BODY program, and the required control variables originally held in this array were commoned or placed in subroutine argument lists.

The addition of the wake body to the original body and the assumption of a plane of symmetry necessitated two important restrictions on specifying data for the NCSU BODY program. First, the line of reference with respect to which the Y-coordinates of the body are specified must be the Y=0 line. Second, the line of reference with respect to which the X-coordinates of the body are specified must be a line parallel to a line from the nose of the body to the tail of the body (see Figure 47). This last restriction was incorporated in order to determine the direction in which the wake body should be added onto the original body.
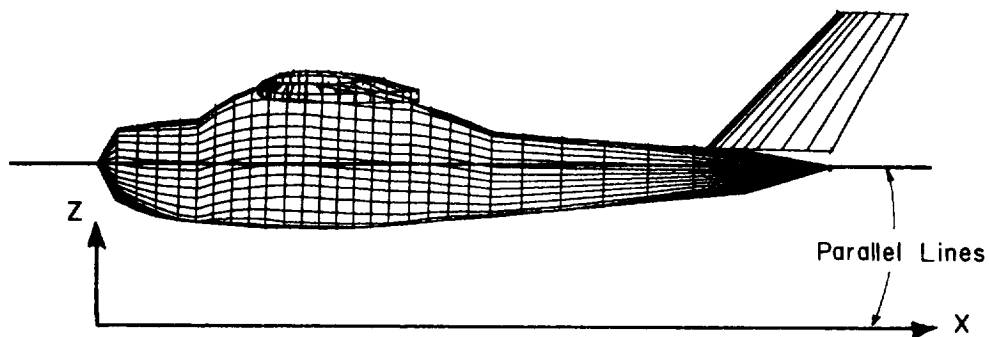
Figure 47. Orientation of body with respect to reference line.

In addition to those mentioned above, there were also other input and output modifications made to the XYZ program. Many of the input integers in the XYZ program were just specified as constant values in the NCSU BODY program: NSE=1, MIX=150, ISM=1, EPS=0.0001, and ISP=0. The other input integers MIY, MIZ, IUCT, IPS, AND IPF were deleted. While the above input variables were deleted, it was also necessary to add the variables VINF (free stream velocity), VO (kinematic viscosity), ROE (density), REFA (reference area), and IWRITE to the NCSU BODY program input. The first three were added to supply the boundary layer routines with the necessary information

245

to compute the Reynolds Number of the body. REFA was added to provide a reference area for normalizing the lift and drag coefficients. IWRITE was added as an output control parameter which has the value 0, 1, or 2. IWRITE= 0 gives maximum output while IWRITE=2 gives minimum output. For an exact description of the input required for the NCSU BODY program the reader is referred to the User Instructions in Appendix F.

246

STREAMLINE MODIFICATION AND BOUNDARY LAYER CALCULATION

The only practical method for predicting the viscous flow field about an arbitrary three-dimensional body involves the same procedure as that used for the viscous flow about two-dimensional airfoils. The basic steps for this procedure are: (1) obtain an inviscid flow solution for the basic arbitrary body, (2) obtain a boundary layer solution based on the inviscid flow solution, (3) construct a modified body by adding a wake body to the original body, and (4) obtain an inviscid flow solution for the body plus wake body to obtain the final pressures and force coefficients on the physical body. These steps represent an outline of the procedure used in the NCSU BODY program.

The basic method for obtaining the inviscid flow solution has already been discussed in the section General Program Theory For Inviscid Body Program. Present methods available for obtaining a three-dimensional boundary layer solution over arbitrary bodies are very time consuming, require a large amount of computer storage, and are therefore beyond the scope of this report. As an alternative, the authors chose to use two-dimensional boundary layer calculations along streamlines to approximate the actual three-dimensional case. Justification for this procedure has already been discussed in detail beginning on page 113 of the theory section. This method is ideally suited for use with XYZ since the program already provides the capability for computing on-body streamlines. Further, the streamline procedure calculates the absolute velocity and surface length from the nose for each streamline point. Considering the approximate nature of two-dimensional boundary layer solutions along streamlines as applied to this problem, it is appropriate to employ relatively simple laminar and turbulent boundary layer calculation procedures. The boundary layer displacement thickness δ^* and wall shear τ_w are calculated at each of the panel centroid points. Assuming τ_w in each panel is constant over that panel area, and given (1) the axial (X-direction) component of velocity from the potential flow solution, (2) the absolute velocity for each panel, and (3) the area of each panel, the skin friction drag coefficient is found by integrating the axial component of τ_w over the body surface. The values of δ^* at the panel centroids are used to construct a wake body which is added to the original body. As in the case of the airfoil, the purpose of the wake body is to model the relief of the stagnation condition at the body tail accompanying the presence of the boundary layer (see page 128). The construction of the wake body is discussed in the next section of this report. Once the wake body is generated, a new inviscid solution for the body plus wake body is found; this solution represents the viscous flow solution over the original body. There is no iterative procedure as was the case with the airfoil because of the large amount of computer time required for each solution.

Thus far this section has presented a brief summary of how the viscous flow solution over an arbitrary body can be obtained by modifying the inviscid flow solution. Attention will now be directed to (1) the actual modifications made in the streamline section of the XYZ program and (2) the addition of the two-dimensional boundary layer calculation procedures.

Since the inviscid solution calculates the velocities at the panel centroid points, the appropriate boundary layer parameters are also estimated at these centroids. This is accomplished by the following procedure:

(1) For a given panel a streamline is traced, using the centroid point as the starting point, from the body nose to three points downstream of the panel centroid. The reader should note that this represents two modifications of the original streamline procedure provided in the XYZ program. First, in the XYZ program, the coordinates for a streamline starting point were read in by the user, while in the NCSU BODY program the streamline starting points are automatically specified as the panel centroid points; thus, the user no longer has the option of inputting the starting coordinates for streamlines. Second, the XYZ program traced a streamline from the nose of the body to the body tail; however, since the boundary layer information is needed only at the panel centroid, computation time is simply wasted by continuing to trace a streamline far downstream of a panel centroid. In the NCSU BODY program at most only three points are traced downstream of the panel centroid point.

(2) A cubic spline curve of streamline velocity versus arc length is fitted to the streamline points. This curve is used to generate a more finely spaced set of velocity versus arc length points as well as the derivative of the velocity with respect to arc length. This information is required by the boundary layer computation procedure. The cubic spline curve was used because it may be differentiated to give smooth first derivatives from tabulated data.

(3) A boundary layer is computed along each streamline with transition fixed at the point on the streamline where the arc length is 5 percent of the total length of the body. If laminar separation arises before this point is reached, the program assumes that turbulent reattachment occurs at the point of laminar separation. The laminar boundary layer is computed using the Holstein-Bohlen formulation of the Kármán-Pohlhausen momentum integral method (Ref. 65). If the velocity for the first point on the streamline is zero then the laminar routine assumes stagnation point starting conditions; however, if a nonzero velocity is found then flat plate starting conditions are used. These two types of starting conditions are necessary since the potential flow program does not always achieve a stagnation point (zero velocity) at the nose of the body. The turbulent boundary layer method, derived by Goradia in Reference 31, is a shortened version of the one used in the airfoil program (Appendix A). Goradia's method is designed to remain stable under the influence of extreme gradients, both favorable and adverse, and provide reasonable momentum and displacement thicknesses downstream of the turbulent separation point. Consequently, turbulent separation is never predicted with this method.

(4) The streamline values of δ^* and τ_w at the point corresponding to the streamline starting point (panel centroid point) are retained. These quantities are later used for the construction of the wake body and the calculation of the skin friction drag coefficient.

The above procedure, steps (1) through (4) are repeated for each panel centroid point except for the triangular panels at the nose and tail of the body. Initiating streamlines from the centroids of triangular panels is omitted because it was found that the streamline tracing procedure experiences great difficulty in the region where apexes of several triangular panels come together (i.e. in the region of the nose or tail). Thus, it is necessary to approximate the values of δ^* and τ_w at the centroids of the panels using the following procedure:

(1) For the triangular panels at the nose of the body, the values of δ^* and τ_w are taken to be one-third of their respective values in the quadrilateral immediately aft of each triangle.

(2) For triangular panels at the tail of the body the values of δ^* and τ_w are taken to be equal to their respective values in the quadrilateral immediately preceeding each triangle.

The streamline procedure was modified in one final way which has not been referred to as of yet. In the XYZ program a search is made of all quadri- laterals to determine the next quadrilateral into which a streamline is traced. In actuality, a streamline traced in either the upstream or downstream direction must enter one of five quadrilaterals adjacent to the quadrilateral it is leaving. Consequently, only these five quadrilaterals need to be checked to see which quadrilateral the streamline will enter. To reduce program execution time this modified search procedure is incorporated in the NCSU BODY program. Figure 48 illustrates the order in which each of the five quadri- laterals are searched when tracing in either the upstream or downstream direction.

It should also be mentioned that in the original XYZ program the stream- line variables were dimensioned large enough to provide for tracing 650 stream- line points. This is well in excess of the number needed for the NCSU BODY program. The array sizes of the appropriate streamline variables were there- fore reduced in the NCSU BODY program in order to reduce overall program size.

249
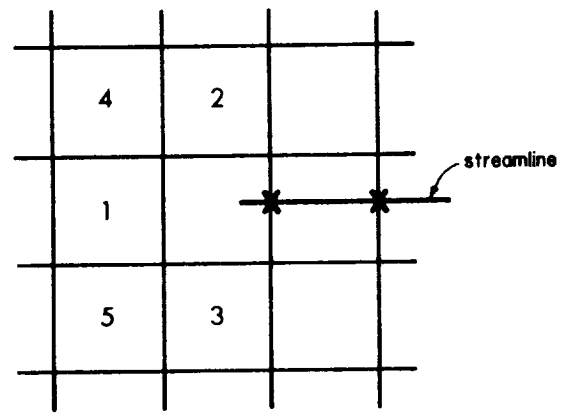
tracing in the up-stream direction
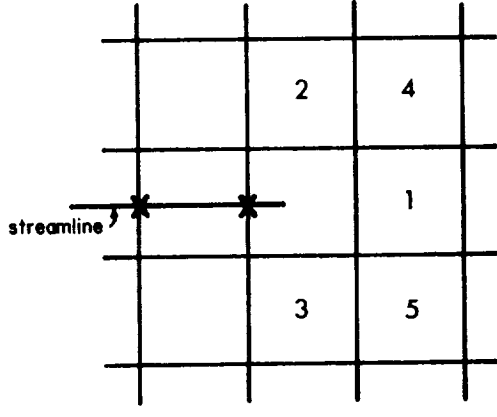
tracing in the down-stream direction

Figure 48. New panel identification procedure
for tracing streamlines.

ADDITION OF WAKE BODY

In order to predict the drag of an arbitrary three-dimensional body both the skin friction drag and pressure drag must be calculated. As seen in the previous section, the skin friction drag coefficient is calculated by integrating the wall shear over the body surface. Unfortunately, estimating the pressure drag using an inviscid flow solution technique is not quite as simple. Actually, the pressure or form drag is a relief of the rear stagnation condition caused by the presence of the boundary layer. Thus, in order to estimate the pressure drag this viscous phenomenon must be correctly modeled in the inviscid flow solution. The authors chose to model this effect with the addition of a wake body. A detailed discussion of the effect of the wake on the pressure drag, the modeling of the wake with a wake body, and the procedure chosen to construct the wake body is given beginning on page 128 in the theory section and is therefore not repeated here. Accordingly, in this section, attention will be directed toward the programming aspect of adding the wake body to the inviscid solution.

(1) The authors chose to define the wake body as the last two sets of panels (modified to some extent) on the original body plus two additional sets of panels downstream of the original body (See Figure 49).
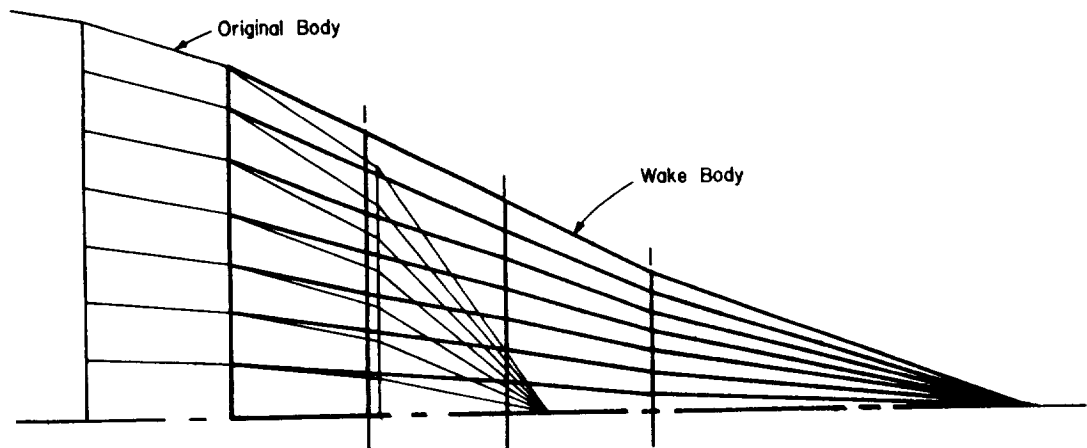
Figure 49. Definition of wake body.

(2) For simplicity's sake, the wake body is chosen as a body of revolution with its axis on the line joining the nose and tail of the original body as shown in Figure 47. The reader should note that the original body must therefore be input using a reference line parallel to the line between the nose and tail of the original body.

(3) The radius of the wake body is chosen to decrease exponentially from the beginning of the wake body to the end. The particular exponential shape is calculated using the average radius and initial slope at the beginning of the wake body as well as the total length of the wake body.
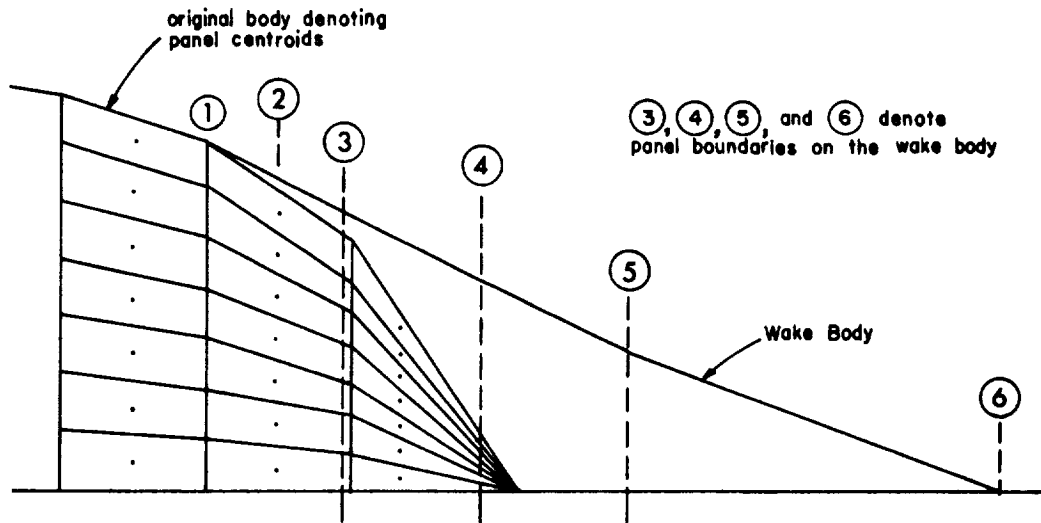
Figure 50.   Panel boundaries on the original body and the wake body.

(4) Noting Figure 50, the average radius of the original body is computed for the X-coordinate of the panel centroids at station 2 and is denoted by AVXCG. The average value is computed for the boundary layer displacement thickness of the panels at station 2 and is denoted by AVDELS.

(5) The actual radius is computed for each body input point at station 1 and is denoted by R1. The radius corresponding to each body input point is computed for each panel at station 2 and is denoted by R2. Twice the average boundary layer displacement thickness is then added to each of the R2 values computed.

(6) A value for the slope (denoted as SLOPE) at each panel around the circumference of the body is then computed from the equation below:

$$SLOPE = (R2 - R1) / (XHOLD1 - AVXCG)$$

where XHOLD1 is the X-coordinate of station 1.

252

(7)   The average slope (denoted by AVSLOP) is computed by summing each slope previously calculated and dividing this sum by the number of points at station 1.   This slope is taken to be the initial slope of the exponential.

(8)   The location of the end of the wake body, XINF, is then computed from the following equation:

$$XINF = XHOLD1 - AREAT / (3.14159*RAV)$$

where AREAT = 4.0*AREAAV*2.0*MMAXQD and MMAXQD is the total number of points around the half body.

(9)   Since all the parameters needed to determine the exponential curve have been found, the X-stations at 3, 4, and 5 are chosen to give panel areas on the surface of the wake body which are approximately equal to AREAAV.   The radii corresponding to these X-stations are then calculated using the exponential curve.

(10)   It should be noted that if a non-negative average slope is calculated for the initial slope of the exponential, the average slope is calculated from the following equation:

$$AVSLOP = - Absolute Value (RAV / (XINF - XHOLD1)).$$

(11)   Using the radii values at stations 3 through 6, the X, Y, and Z-coordinate values are generated for surface points on the wake body.

(12)   Since the geometric information for the panels on the original body ahead of station 1 need not be changed, this information is copied from KFILE1 to KFILE2 (see page 243) in General Program Modifications). The geometric information for the wake body panels is then calculated and this information is added to KFILE2 in such a manner that body plus wake body appears as one large pseudo-body.

(13)   The pressure coefficients for the panels on the pseudo-body are calculated by finding a new inviscid solution over this body.   These pressure coefficients are then applied to the corresponding panels on the original physical body along the normals of the original panels (see pages 127 - 132).   The appropriate components of these pressure coefficients are then integrated over the body surface to yield a pressure drag coefficient and a pressure lift coefficient.

As an illustration of the shape and location of the wake body with respect to the original body Figures 51 and 52 are included.   Figure 51 depicts a 3-1 ellipsoid before and after a wake body is added, while Figure 52 shows the Cessna 182 which has an arbitrary cross-section.
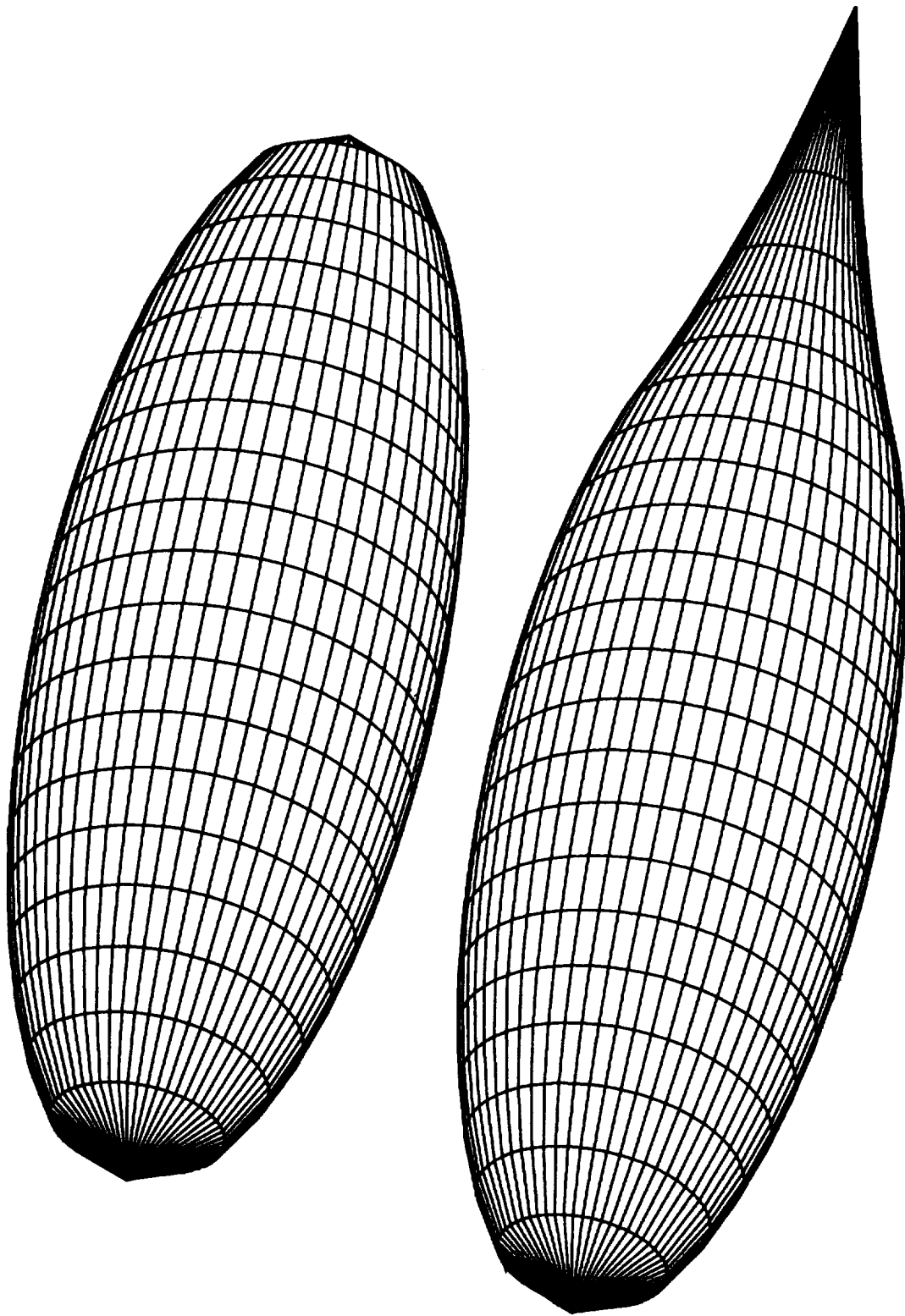
Figure 51. 3-1 ellipsold with and without a wake body.

254
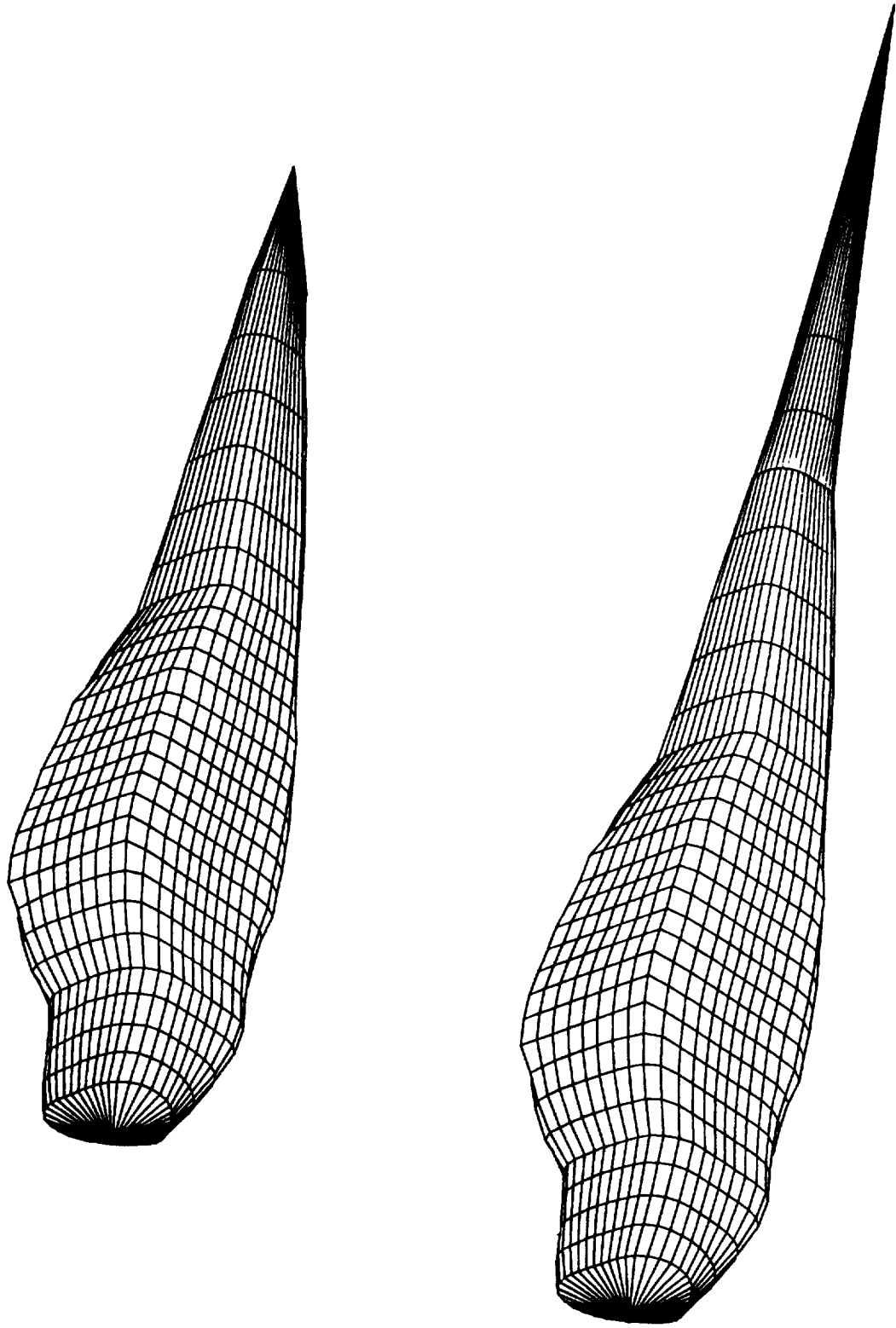
Figure 52. Cessna 182 Fuselage with and without a wake body.

255

DISCUSSION OF PROGRAM RESULTS

To the time of the present writing it has not been possible to investigate the applicability of the NCSU BODY program to as many test cases as the authors would have liked. The greater geometric variability and intricacy characteristic of fuselages (in contrast to relatively simple geometry of airfoil families at least), the paucity of reliable experimental data, the extended development time required for the program, and finally, the cost of running the program (more than 10 times as much as the airfoil program) all served to limit the number of runs made using the final version of the program. Three bodies, however, were studied.

The first was a sphere. For a Reynolds Number of 2 million (based on sphere diameter) a drag coefficient of 0.041 was computed. This is about 1/5 the commonly accepted value for the sphere at such Reynolds Numbers. The reason for the low drag value, however, is easy to explain. It may be recalled that the computer program adds the wake body only behind the last two sets of surface panels in the X-direction. For a sphere the wake obviously emanates from a much larger area than the last two sets of panels if one uses a reasonably large number of quadrilaterals to represent the body. During the course of program development it was readily demonstrated that beginning the wake body at the point where separation is observed experimentally does in fact yield the correct value of drag. However, most streamlined bodies produce proportionately smaller separated flow regions; since the program is intended for use primarily with streamlined bodies, it was felt that the wake body formation procedure should attempt to model closely only the wakes of such bodies. One might also point out that if one better understood the relationships among the direction of and location of the flow separation from the body and the body geometry and flow Reynolds Number, then one could include analytical versions of these relationships in the system of equations used to calculate the flow; the appropriate wake body would then come out as part of the solution. In the absence of such knowledge, we can do little more than choose a model representative of one class of bodies.

The second body against which the program predictions were tested was a 3:1 prolate spheroid. In this case a Reynolds Number of 6 million was used in the computations. With 560 panels the calculated drag coefficient was 0.094 With only 100 panels the drag coefficient was 0.109 indicating the effect which a poorer representation of the body has on the computed drag value. No direct experimental data was found for this case but a 2:1 ellipsoid was found (Ref. 117) to give C_D = .07. There is also some evidence that the 3:1 ellipsoid should have a slightly higher drag because the surface area (skin friction) increases faster than the wake size, and hence, the form drag, diminishes. At any rate, the drag coefficient computed by the program for prolate spheroids with fineness ratios of 3 to 10 appears to be approximately correct.

The final test case was the Cessna 182 fuselage. It was found quite difficult to represent such a body with 560 or fewer panels of nearly equal area. As a result, while the rule of keeping adjacent panel areas within a ratio of 1:1.5 was honored, there is substantial variation in panel areas between the nose region (where many panels are needed to represent the geometry satisfactorily) and the aft cabin region (where large panels are adequate). This variation, it is felt, could be responsible for some error in the computation. The computed drag coefficient value, based on wing area and a length Reynolds Number of 30 million, was .01245. The body lift coefficient at the same condition was 0.00444. The computation, it may be noted, also assumes a turbulent boundary layer for most of the fuselage. It is therefore applicable primarily to the higher speed portions of the flight envelope.

Of course no test data were available against which to compare these figures, but a computation using the $C_{D\pi}$ method (Ref. 2) gives a drag coefficient .00876, about 30% less than the value given by the NCSU BODY program. The $C_{D\pi}$ method is probably the method most often used for preliminary design in the light aircraft industry. It does not differentiate between laminar and turbulent boundary layers but nevertheless it seems to yield drag values which match measured performance reasonably well. Thus it is to be expected that the drag value given by the NCSU BODY program is approximately correct. A more detailed test of the ability of the NCSU BODY program and the other programs discussed in the present work to predict the aerodynamic characteristics of actual aircraft is planned for early 1975. A light twin with an advanced wing design is being carefully instrumented for a series of performance and stability tests. It will be among the objectives of these tests to develop lift and drag flight test data against which to compare the prediction of the programs given in the present work. Until the results of this and other comparisons are available, the authors suggest that potential users of the NCSU BODY program employ its predictions cautiously until its range of validity is better defined.

REFERENCES

1. Smetana, F. O.; Summey, D. C.; and Johnson, W. D.: "Point and Path Performance of Light Aircraft - A Review and Analysis". NASA CR-2272, June 1973, 131 pages.

2. Smetana, F. O.; Summey, D. C.; and Johnson, W. D.: "Riding and Handling Qualities of Light Aircraft - A Review and Analysis". NASA CR-1975, March 1972, 409 pages.

3. Smetana, F. O.; Summey, D. C.; and Johnson, W. D.: "Flight Testing for the Evaluation of Light Aircraft Stability Derivatives - A Review and Analysis". NASA CR-2016, May 1972, 110 pages.

4. von Helmholtz, H.: "Über discontinuirliche Flüssingheitsbewegungen". Monatsberichte der Königlichen Academic der Wissenschaften zu Berlin, 1868, pp. 215-228.

5. Kirchhoff, G.: "Zur Theorie freier Flüssigkeitsstrahlen". Journal für die reine und angewandte Mathematik, 1869, pp. 289-298.

6. Lamb, Horace: Hydrodynamics. Cambridge University Press, First Edition 1879, Sixth Edition 1932.

7. Kutta, M. W.: "Auftriebkräfte in strömenden Flüssigkeiten". Illustrierte aeronautische Mitteilungen, Vol. 6, 1902, pp. 133-135.

   Kutta, M. W.: "Über eine mit den Grundlagen des Flugproblems in Beziechung Stehende Zwei dimensionale Strömung". Sitzungsberichte der Bayerischen Akademie der Wissenschaften, 1910, pp. 1-58.

8. von Karman, Theodore: Aerodynamics. Cornell University Press, 1954.

9. Joukowski, N.: "Über die Konturen der Tragflächen der Drachenflieger". ZFM, 1910, pp. 281-284.

10. von Karman, T.; and Trefftz, E.: "Potentialströmung um gegebene Trangflächenquerschnitte". ZFM, 1918.

11. von Mises, R.: "Zur Theorie des Tragflächenauftriebes". Zeitschrift für Flugtechnik und Motorluftschiffahart (ZFM) Vol. II, p. 68, 1920.

12. Müller, W.: "Zur Konstruktion von Tragflächenprofilen". ZAMM, 1924.

13. Theodorsen, T.: "Theory of Wing Sections of Arbitrary Shape". NACA Report No. 411, 1932.

14. Munk, Max M.: "General Theory of Wing Sections". NACA Report 142, 1922.

15. Birnhaum: "Die tragende Wirbelfläche als Hilfsmittel zur Behandlung des ebenen Problems der Tranglügel Theorie". ZAMM, 1923.

16. Glauert, H.: "Theory of Thin Aerofoils". A.R.C. R & M 910 1224.

    Glauert, H.: The Elements of Aerofoil and Airscrew Theory. Cambridge University Press, 1966.

17. Karamcheti, K.: Principles of Ideal-Fluid Aerodynamics. John Wiley and Sons, New York. 1966.

18. Abbott, Ira H.; and von Doenhoff, Albert E.: Theory of Wing Sections. Dover Publications, Inc., New York, 1959, 693 pages.

19. Abbott, Ira H.; von Doenhoff, Albert E.; and Stivers, Louis S.: "Summary of Airfoil Data". NACA TR No. 824, 1945.

20. Weber, J.: "The Calculation of the Pressure Distribution over the Surface of Two-Dimensional and Swept Wings with Symmetrical Airfoil Sections". A.R.C. R & M 2918, July 1953.

21. Weber, J.: "The Calculation of the Pressure Distribution on the Surface of Thick Cambered Wings and the Design of Wings with Given Pressure Distribution". A.R.C. R & M 3026, June 1955.

22. Bridewater, J.; and Whitlery, M. D.: "An Algol Program for Calculating the Inviscid Subsonic Pressure Distribution on Airfoils by the Weber-Küchemann Method". NPL Aero Note 1062, July 1967, N68-15738.

23. Hess, J. L.; and Smith, A. M. O.: "Calculation of Potential Flow about Arbitrary Bodies". Progress in Aeronautical Sciences, Vol. 8, pp. 1-138, Pergamon Press, 1967.

24. Martensen, E.: "Berechnung der Druckverteilung an Gitterprofilen in ebener Potentialstromung mit einer Fredholmscher Integralgleichung". Arch. Rational Mech. Anal, Vol 3, No. 3, 1959, pp. 235-270.

25. Jacob, K.; and Riegels, F. W.: "Berechnung der Druckverteilung endich dicker Profile ohne und mit Kapper und Vorflugeln". Zeitschrift für Flugwissenschaften, Vol. 11, No. 9, Sept. 1963, pp. 357-367.

26. Oellers, H. J.: "Die Inkompressible Potentialstromung in der Ebener Gitterstufe". Jahrbuch 1962 der Wissenschaftlichen Gessellschaft für Luft und Roumfahrt, e.v., pp. 349-353.

27. Chen, A. Wen-Shin: "The Determination of the Geometries of Multiple-Element Airfoil Optimized for Maximum Lift Coefficient". Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1971.

260

28. Lighthill, M. J.: "A New Method of Two-Dimensional Aerodynamic Design". A.R.C. R & M 2112, April 1945.

29. Sato, J.: "An Exact Two-Dimensional Incompressible Potential Flow Theory of Airfoil Design with Specified Velocity Distributions". Transactions Japan Society Aero. Space Sciences, Vol. 9, No. 14, pp. 11-18, 1966.

30. Powell, B. J.: "The Calculation of the Pressure Distribution on a Thick Cambered Aerofoil at Subsonic Speeds, Including the Effects of the Boundary Layer". A.R.C. CP. 1005, June 1967, N69-13421.

31. Stevens, W. A.; Goradia, S. H.; and Braden, J. A.: "Mathematical Model for Two-Dimensional Multi-Component Airfoils in Viscous Flow". NASA CR-1843, July 1971, 181 pages.

32. Bennett, J. A.; and Goradia, S. H.: "Methods for Analysis of Two-Dimensional Airfoils - With Subsonic and Transonic Applications". Lockheed Georgia Company Report CR 8591, July 1966.

33. Van Dyke, Milton D.: "Second-Order Subsonic Airfoil Theory Including Edge Effects". NACA Report 1274, 1956.

34. Lanchester, F. W.: Aerodynamics. London Constable & Co. Ltd., 1907; Aerodonetics, London, 1908.

35. Prandtl, Ludwig: "Tragflügeltheorie". Göttinger Nachrichten, 1918, pp. 451-477.

36. McVeigh, M. A.; and Kisielowski, E.: "A Design Summary of Stall Characteristics of Straight Wing Aircraft". NASA CR-1646, June 1971, 209 pages.

37. Margason, R. J.; and Lamar, J. E.: "Vortex-Lattice Fortran Program for Estimating Subsonic Aerodynamic Characteristics of Complex Planforms". NASA TND-6142, February 1971, 139 pages.

38. Jordan, Peter F.: "Remarks on Applied Lifting Surface Theory". Research Institute for Advanced Studies, Baltimore, Md., August 1967. Available from NTIS as N68-32042.

39. Coote, Ian: "A Fortran Program for Determining the Subsonic Lift Distribution on a Wing Using Lifting-Surface Theory". Department of Aeronautical Engineering, Sydney University. Available from NTIS as N71-33220.

40. Yager, P. M.; Holland, C. H.; and Strand, T.: "Modified Weissinger Lifting-Surface Method for Calculating Aerodynamic Parameters of Arbitrary Wing-Canard Configurations". Air Vehicle Corporation Report No. 354, August 1967, AD660423.

41. Garner, H. C.: "Numerical Appraisal of Multhopp's Low-Frequency Subsonic Lifting Surface Theory". A.R.C. R & M No. 3634, 1970.

42. James, R. M.: "On the Remarkable Accuracy of the Vortex Lattice Discretization in Thin Wing Theory". McDonnel-Douglas Aircraft, Inc., Report No. DAC-67211, February 1969, N69-37939.

43. Garner, H. C.; Hewitt, B. L.; and Labrujere, T. E.: "Comparison of Three Methods for the Evaluation of Subsonic Lifting-Surface Theory". A.R.C. 30 324, June 1968, N69-19965.

44. Coote, Ian: "Note on the Spanwise Integration of the Kernel Function in Subsonic Lifting-Surface Theory". Sydney University Department of Aeronautical Engineering, Aero Tech Note 6093, November 1969, N71-33219.

45. Margason, Richard, J.; and Lamar, John E.: "Vortex-Lattice Fortran Program for Estimating Subsonic Aerodynamic Characteristics of Complex Planforms". NASA TND-6142, February 1971.

46. Hess, J. L.; and Faulkner, Suzanne M.: "Determination of Low Speed Interference Effects by Superposition". Available from NTIS as N71-19377.

47. Anon: "Analytical Methods in Aircraft Aerodynamics". NASA SP-228, 1970. A collection of 31 papers by various authors.

48. Carmichael, Ralph L.: "Recent Experience in Using Finite Element Methods for the Solution of Problems in Aerodynamic Interference". NASA TMX-66884, 1971.

49. Timman, R.: "The Potential Flow About a Yawed Ellipsoid at Zero Incidence". National Aerospace Laboratory NLR, The Netherlands, Report F74. N68-34853 from NTIS.

50. Loeve, W.: "Computer Programs in Use at NLR for the Calculation of Stationary Subsonic Flow around Wing-Body Combinations". Report AT-69-12. Available from NTIS as N70-18119.

51. Multhopp, H.: "Aerodynamics of the Fuselage". NACA TM 1036, December 1942.

52. Bingham, Gene J.; and Chen, Allen Wen-shin: "Low Speed Aerodynamic Characteristics of an Airfoil Optimized for Maximum Lift Coefficient". NASA TND-7071, December 1972, 53 pages.

53. Roskam, Jan: Methods for Estimating Drag Polars of Subsonic Aircraft. Published by the author, Lawrence, Kansas, 1971.

54. Cebeci, Tuncer; Mosinskis, G. J.; and Smith, A. M. O.: "Boundary Layer Separation on Two-Dimensional and Axisymmetric Bodies in Incompressible Flows". McDonnell-Douglas Corp. Report No. MDC-J0973-01, November 1970, N71-25868.

262

55. Lundry, J. L.: "A Numerical Solution for the Minimum Induced Drag, and the Corresponding Loading, of Non-Planar Wings". McDonnell-Douglas Corp. Report DAC-66900, NASA CR-1218.

56. Liepmann, H. W.; and Puckett, A. E.: *Aerodynamics of a Compressible Fluid*. John Wiley and Sons, New York, 1947, 262 pages.

57. Liepmann, H. W.; and Roshko, A.: *Elements of Gasdynamics*. John Wiley and Sons, New York, 1957, 439 pages.

58. Mercer, J. E.; Weber, J. A.; and Lesferd, E. P.: "Aerodynamic Influence Coefficient Method Using Singularity Splines". AIAA Paper No. 73-123, 7 pages.

59. Lamar, John E.: "A Modified Multhopp Approach for Predicting Lifting Pressures and Camber Shape for Composite Planforms in Subsonic Flow". NASA TND-4427, July 1968.

60. Multhopp, H.: "Methods for Calculating the Lift Distribution on Wings (Subsonic Lifting Surface Theory)". A.R.C. R & M No. 2884, January 1950.

61. Garner, H. C.; and Fox, D. A.: "Algol 60 Programme for Multhopp's Low-Frequency Subsonic Lifting-Surface Theory". A.R.C. R & M No. 3517, 1968.

62. Wolowicz, Chester H.; and Yancey, Roxanah B.: "Longitudinal Aerodynamic Characteristics of Light, Twin-Engine Propeller-Driven Airplanes". NASA TND-6800, June 1972, 361 pages.

63. McNally, William D.: "Fortran Program for Calculating Compressible Laminar and Turbulent Boundary Layers in Arbitrary Pressure Gradients". NASA TND-5681, May 1970, 107 pages.

64. Küchemann, D.; and Weber, J.: "The Subsonic Flow Past Swept Wings at Zero Lift without and with Body". A.R.C. R & M No. 2908, March 1953.

65. Schlichting, H.: *Boundary Layer Theory*. New York, McGraw-Hill Book Co. Inc., Fourth Edition, 1960.

66. Durand, W. F., editor: *Aerodynamic Theory*. Dover Publications, Inc., Vol. I Division C, 1963, reprint of 1934 edition.

67. Nicolai, L. M. and Sanchez, F.: "Correlation of Wing-Body Combination Lift Data". *Journal of Aircraft*, Vol. 10, No. 2, 1973, pp. 126-128.

68. Angelucci, S. B.: "Multivortex Model for Bodies of Arbitrary Cross-Sectional Shapes". AIAA Paper No. 73-104.

69. Callaghan, J. G. and Beatty, T. D.: "A Theoretical Method for the Analysis and Design of Multielement Airfoils". _Journal of Aircraft_, Vol. 9, No. 12, December 1972, pp. 844-848.

70. Spence, D. A.: "Wake Curvature and the Kutta Condition". _Journal of Fluid Mechanics_, Vol. 44, 1970, part 4, pp. 625-636.

71. Riley, N.; and Stewartson K.: "Trailing Edge Flows". _Journal of Fluid Mechanics_, Vol. 39, 1969, part 1, pp. 193-207.

72. Preston, J. H.; and Sweeting, N. E.: "The Experimental Determination of the Boundary Layer and Wake Characteristics of a Simple Joukowski Aerofoil, with Particular Reference to the Trailing Edge Region". A.R.C. R & M No. 1998, March 1943.

73. Preston, J. H.; Sweeting, N. E.; and Cox, Miss D. K.: "The Experimental Determination of the Boundary Layer and Wake Characteristics of a Piercy 12/40 Aerofoil, with Particular Reference to the Trailing Edge Region". A.R.C. R & M No. 2013, February 1945.

74. Howarth, L.: "The Theoretical Determination of the Lift Coefficient of a Thin Elliptic Cylinder". _Proc. Roy. Soc. A_, 149, pp. 558-586, 1935.

75. Preston, J. H.: "The Calculation of Lift Taking Account of the Boundary Layer". A.R.C. R & M No. 2725, November 1949.

76. Preston, J. H.: "Note on the Circulation in Circuits which Cut the Streamlines in the Wake of an Aerofoil at Right-Angles". A.R.C. R & M No. 2957, March 1954.

77. Spence, D. A.; and Beasley, J. A.: "The Calculation of Lift Slopes Allowing for Boundary Layer, with Applications to the RAE 101 and 104 Aerofoils". A.R.C. R & M No. 3137, February 1958.

78. Bollech, Thomas V.: "Experimental and Calculated Characteristics of Several High-Aspect-Ratio Tapered Wings Incorporating NACA 44-Series, 230-Series, and Low-Drag 64-Series Airfoil Sections". NACA TN 1677, September 1948, 37 pages.

79. Dommasch, Daniel O.; Sherby, Sydney S.; and Connolly, Thomas F.: _Airplane Aerodynamics_. Pitman Publishing Corporation, 1967.

80. Bhateley, Ishwar C.; and McWhirter, Jack W.: "Development of Theoretical Method for Two-Dimensional Multi-Element Airfoil Analysis and Design". AFFDL-TR-72-96, Part I, August 1972, 366 pages.

264

81. Hess, J. L.: "Numerical Solution of the Integral Equation for the Neumann Problem with Applications to Aircraft and Ships". SIAM Symposium on Numerical Solution of Integral Equations with Physical Applications, Madison, Wisconsin, October 1971, 108 pages. Also Douglas Aircraft Co. Engineering Paper No. 5987.

82. Hess, J. L.: "Higher Order Numerical Solution of the Integral Equation for the Two-Dimensional Neumann Problem". Computer Methods In Applied Mechanics and Engineering, Vol. 2, 1973, pp. 1-15.

83. Hess, John L.; and Smith, A. M. O.: "Calculation of Non-Lifting Potential Flow about Arbitrary Three-Dimensional Bodies". Douglas Aircraft Co. Report No. E. S. 40622, March 15, 1962, 177 pages.

84. Labrujere, T. E.; Loeve, W.; and Slooff, J. W.: "An Approximate Method for the Calculation of the Pressure Distribution on Wing Body Combinations at Subcritical Speeds". Publication of the National Aeronautical Laboratory (NLR), Amsterdam, Netherlands.

85. Loeve, W.; and Slooff, J. W.: "On the Use of Panel Methods for Predicting Subsonic Flow about Aerofoils and Aircraft Configurations". NLR MP 71018 U, June 10, 1971.

86. Keith, J. S.; Ferguson, D. R.; Merkle, C. L.; Heck, P. H.; and Lahti, D. J.: "Analytical Method for Predicting the Pressure Distribution about a Nacelle at Transonic Speeds". NASA CR-2217, July 1973.

87. Hess, J. L.: "Calculation of Potential Flow about Arbitrary Three-Dimensional Lifting Bodies". Douglas Aircraft Company Report MDC-J5679-01, October 1972.

88. Giesing, J. P.; Kálmán, T. P.; and Rodden, W. P.: "Subsonic Steady and Oscillatory Aerodynamics for Multiple Interfering Wings and Bodies", Journal of Aircraft, Vol. 9, No. 10, October 1972, pp. 693-702.

89. Woodward, F. A.: "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations In Subsonic and Supersonic Flow. Part I - Theory and Application. Part II - Computer Program Description". NASA CR-2228, May 1973, 126 pages and 315 pages.

90. Widnall, Sheila: "Subsonic Aerodynamics". Astronautics and Aeronautics, April 1973, pp. 14, 21, and 28.

91. Otto, H.: "Calculation of Nonlinear Lift and Pitching Moment Coefficients for Slender Wind-Body Combinations". Journal of Aircraft, Vol. 11, No. 8, August 1974, pp. 489-491.

92. Roskam, Jan; and Lan, C.: "A Parametric Study of Planform and Aeroelastic Effects on Aerodynamic Center, α- and q-Stability Derivatives". NASA CR-2117, April 1973.

93. Bleekrode, A. L.: "A Survey of Current Collocation Methods in Inviscid Subsonic Lifting Surface Theory. Part II - Calculational Aspects of Solving the Large Systems of Linear Algebraic Equations on a Digital Computer". Publication of NLR Amsterdam, Netherlands.

94. Dawson, Charles W.; and Dean, Janet S.: "The XYZ Potential Flow Program". Naval Ship Research and Development Center, Bethesda, Maryland, Report 3892, June 1972, 65 pages.

95. Hopkins, E. J.: "A Semi-Empirical Method for Calculating the Pitching Moment of Bodies of Revolution at Low Mach Numbers". NACA RM A51C14. 1951.

96. Goodson, K. W.: "Effect of Nose Length, Fuselage Length, and Nose Fineness Ratio on the Longitudinal Aerodynamic Characteristics of Two Complete Models at High Subsonic Speeds". NASA Memo 10-10-58L. 1958.

97. Pitts, W.; Nielsen, J.; and Kaattari, G.: "Lift and Center of Pressure of Wing-Body-Tail Combinations at Subsonic, Transonic, and Super-sonic Speeds". NACA TR 1307. 1957.

98. Donovan, A. F., and Lawrence, H. R., editors, Aerodynamic Components of Aircraft at High Speeds, Section C: "Interaction Problems" by C. Ferrari. Princeton University Press, 1957. pp. 281-552.

99. Cebeci, Tuncer; Kaups, Kalle; Mosinskis, G. J.; and Rehn, J. A.: "Some Problems of the Calculation of Three-Dimensional Boundary-Layer Flows on General Confirurations". NASA CR-2285. July 1973. 56 pp.

100. Henrici, Peter: Elements of Numerical Analysis, John Wiley, New York, 1964. 336 pp.

101. Allen, H. J., and Perkins, E. W.: "A Study of Viscosity on Flow over Slender Inclined Bodies of Revolution". NACA TR 1048. 1951.

102. Smetana, F. O.: "Investigation of Free Stream and Stagnation Pressure Measurement from Transonic and Supersonic Aircraft. Final Report." WADC TR-55-238. April 1958.

103. Smetana, F.O., and Knepper, D.P.: "Toward Simpler Prediction of Tran-sonic Airfoil Lift, Drag, and Moment".Journal of Aircraft, Vol. 10, No. 2, Feb. 1973. pp. 124-126.

104. Truitt, R.W.: "Shockless Transonic Airfoils," AIAA Paper 70-187, New York. 1970.

266

105.   Sinnott, C. S.; and Osborne, J.: "Review and Extension of Transonic
        Airfoil Theory". A.R.C. R & M 3156. 1958.

106.   Thompson, N.; and Wilby, P. G.: "Transonic Aerodynamics". AGARD Con-
        ference Proceedings No. 35 (AD-685270). September 1968.

107.   Riger, D. D.; and Rose, N. J.: Differential Equations with Applications,
        McGraw Hill Book Co., New York, 1968, pp. 545.

108.   Milgram, Jerome H.: "Section Data for Thin, Highly Cambered Airfoils in
        Incompressible Flow". NASA CR-1767, July 1971, pp. 72.

109.   Cebeci, T.; Mosinskis, G. J.; and Smith, A. M. O.: "Calculation of Vis-
        cous Drag of Two-Dimensional and Axisymmetric Bodies in Incompressible
        Flows". AIAA Paper 72-1, pp. 11.

110.   Cebeci, T.; Mosinskis, G. J.; and Smith, A. M. O.: "Calculation of
        Separation Points in Incompressible Turbulent Flows". Journal of
        Aircraft, Vol. 9, No. 9, pp. 618-624, September 1972.

111.   Granville, P. S.: "The Calculation of Viscous Drag of Bodies of Revo-
        lution". The David Taylor Model Basin Report 849. July 1953.

112.   Von Mises, Richard: Theory of Flight, McGraw-Hill Book Co., New York.
        1945.

113.   Craidon, Charlotte B.: "Description of a Digital Computer Program for
        Airplane Configuration Plots", NASA TM X-2374, September 1970, pp. 84.

114.   Klunker, E. B.; and Newman, P. A.: "Computation of Transonic Flow about
        Lifting Wing-Cylinder Combinations". Journal of Aircraft, Vol. 11,
        No. 4, April 1974, pp. 254-256.

115.   Ahlberg, J. H.; Nilson, Edwin N.; and Walsh, Joseph L.: The Theory of
        Splines and Their Applications. Academic Press, New York , 1967.

116.   Marshall, F. J.; and Deffenbaugh, F. D.: "Separated Flow Over Bodies
        of Revolution Using an Unsteady Discrete-Vorticity Cross Wake: Part
        I - Theory and Application; Part II - Computer Program Description".
        NASA CR-2414 & 2415, June, 1974. Part I - pp. 89. Part II - pp. 152.

117.   Rouse, Hunter: Elementary Mechanics of Fluids. John Wiley and Sons,
        New York, 1947.

APPENDICES

APPENDIX A- Two-Dimensional Wing Aerodynamic Characteristics Program

User Instructions

The program is written in FORTRAN IV and is designed to run in single precision on an IBM 370-165 computer with an average execution time of 20 to 25 seconds for each angle of attack. The program calculates the two-dimensional viscous flow solution of an arbitrary airfoil and evaluates its aerodynamic force and moment coefficients. The program requires the specification of the following input data:

(1) The 80 characters of the array TITLE which are used as a header for identifying output. Since the program allows more than one airfoil to be analyzed in a given run, TITLE is used as a control variable to end execution. Termination of execution is achieved by following the last set of airfoil data to be analyzed by a title card having only the word END in the first three spaces (see last card of the sample data set in Figure A-2).

(2) The number NXU of specified upper surface airfoil coordinates, the number NXL of specified lower surface airfoil coordinates, the control parameter IWRITE, the control parameter IALPHA, and the control parameter IPUNCH. The largest allowed value of either NXU or NXL is 65 and no more than 100 total airfoil points may be specified (NXU + NXL ≤ 100). The control parameter IWRITE (either 0, 1, 2, or 3) determines the amount of output desired for each set of airfoil data. IWRITE = 0 yields the maximum amount of output while IWRITE = 3 yields the minimum (examples of all the IWRITE options are given in Figure A-3 through Figure A-6). The control parameter IALPHA specifies the line of reference for the angle of attack. The line of reference is the x-axis of the reference system of the input data points. For IALPHA = 1 the line of reference is the longest chord line of the airfoil. Thus, if the user knows the location of the chord line of the airfoil and specifies the airfoil data points so that the chord line is parallel to the x-axis of the reference system of the input data points, he should use IALPHA = 0. If the user is unsure of the position of the airfoil chord line with respect to the x-axis of the input data system, he may choose IALPHA = 1, in which case, the program calculates the longest chord line and references the angle of attack to this line. The user will find that for most cases he will want to use the IALPHA = 0 option. For further explanation of this option see page 112. The control parameter IPUNCH gives the user the option of obtaining punched data (lift, drag, and quarter-chord moment coefficients for each angle of attack) which may be used in the program designed to estimate three dimensional aerodynamic characteristics of wings (see Appendix C). IPUNCH = 1 gives punched output while the default IPUNCH = 0 gives none.

(3) The NXU values of the abscissa XU for the upper surface input points. The XU array should be monotonic increasing from airfoil leading to trailing edge with 8 points per data card.

(4) The NXU values of the upper surface ordinate ZU which correspond to the XU values. The ZU array is specified with 8 data points per card.

(5) The NXL values of the abscissa XL for the lower surface input points. The XL array should be monotonic increasing from airfoil leading to trailing edge with 8 points per data card.

(6) The NXL values of the lower surface ordinate ZL which correspond to the XL values. The ZL array is specified with 8 data points per card.

(7) The number NA of angles of attack to be read for which a solution is desired. NA must be less than or equal to 10.

(8) The NA values of angle of attack, ALPHA, given in degrees, 8 values per data card.

(9) The number NM of free stream Mach numbers to be read for which a solution is given at each of the NA angles of attack. NM must be less than or equal to 5.

(10) The NM values of free stream Mach number, FSMACH, 5 values per data card.

(11) The reference chord CREF in feet and the scale factor SF. The reference chord is used to non-dimensionalize all of the output. The scale factor is a multiplicative constant used to convert the values of XU, ZU, XL, ZL, XTRAN, and ZTRAN to feet. It would be advantageous for the user to input the airfoil coordinates as percentages of the reference chord so that CREF and SF will have the same numerical values.

(12) The stagnation temperature TO in degrees Rankine, the Reynolds number RN in millions, the Prandtl number PR, and the heat transfer factor KF. Reynolds number should be calculated based on CREF, and since it is read in millions, RN = 3.0 corresponds to a Reynolds number of 3,000,000. (It should be noted that this Reynolds number specification differs from that used in the NASA program which used millions per foot.) It is recommended that a value of 1.0 be used for the heat transfer factor. For cases where heat transfer effects may be important, the user should consult Reference 32 to determine an appropriate value for KF.

(13) The control variable LTRAN which determines whether boundary layer transition is free (LTRAN = 0) or fixed (LTRAN = 1), the x location for transition XTRAN, and the z location for

transition ZTRAN. If free transition (LTRAN = 0) is used both XTRAN and ZTRAN should be specified as 0.0, and the program will predict the location of the transition points. It should be noted that if the fixed transition option is used the program will still use its own predicted point of transition if it occurs before the specified transition point. Since transition must be specified on both surfaces, _two_ transition cards must be read (upper surface card first).

Statements (1) through (13) represent a complete set of data for a particular airfoil. The format specification for this data is given in Figure A-1. A sample data set of the 23012 airfoil with IWRITE = 3 option is shown in Figure A-2. The output of this particular data set is shown in Figure A-3, and in addition, examples of the other IWRITE options are given in Figures A-4, A-5, and A-6.

272

TITLE
20 A4

NXU	NXL	IWRITE	IALPHA	IPUNCH	
I5	I5	I5	I5	I5	/////////

XU(1)	XL(2)		etc.				
F10.0	F10.0						

ZU(1)	ZU(2)		etc.				
F10.0	F10.0						

XL(1)	XL(2)		etc.				
F10.0	F10.0						

ZL(1)	ZL(2)		etc.				
F10.0	F10.0						

NA	
I5	/////////

ALPHA(1)	ALPHA(2)		etc.				
F10.0	F10.0						

NM	
I5	/////////

FSMACH(1)	FSMACH(2)		etc.				
F10.0	F10.0						

CREF	SF	
F10.0	F10.0	/////////

TO	RN	PR	KF	
F10.0	F10.0	F10.0	F10.0	/////////

LTRAN(1)	XTRAN(1)	ZTRAN(1)	
I10	F10.0	F10.0	/////////

LTRAN(2)	XTRAN(2)	ZTRAN(2)	
I10	F10.0	F10.0	/////////

Figure A-1. Format specification of input data for the 2-D characteristics program.

273

TITLE

END

LTRAN(2) XTRAN(2) ZTRAN(2)
0   0.0   0.0

LTRAN(1) XTRAN(1) ZTRAN(1)
0   0.0   0.0

TO	RN	PR	KF
518.69	3.0	0.77	1.0

CREF   SF
1.0   1.0

FSMACH(1)
0.2

NM
1

ALPHA(1)   ALPHA(NA)
0.0   2.0   4.0   6.0   8.0   10.0   12.0   14.0

NA
8

ZL(NZL)
-0.007   -0.0013

-0.0428   -0.0446   -0.0448   -0.0417   -0.0367   -0.03   -0.0218   -0.0123

ZU(1)   ZU(2)   etc.
0.0   -0.0123 -0.0171 -0.0228 -0.0261 -0.0292 -0.035   -0.0397

XL(NXL)
0.95   1.0

0.25   0.3   0.4   0.5   0.6   0.7   0.8   0.9

XL(1)   XL(2)   etc.
0.0   0.0125   0.025   0.05   0.075   0.1   0.15   0.2

ZU(NZU)
0.0092   0.0013

0.076   0.0756   0.0714   0.0614   0.0547   0.0436   0.0308   0.0168

ZU(1)   ZU(2)   etc.
0.0   0.0367   0.0361   0.0491   0.059   0.0643   0.0719   0.075

XU(NXU)
0.95   1.0

0.25   0.3   0.4   0.5   0.6   0.7   0.8   0.9

XU(1)   XU(2)   etc.
0.0   0.0125   0.025   0.05   0.075   0.1   0.15   0.2

NXU   NXL   IWRITE   IALPHA   IPUNCH
18   18   3   0   0

TITLE
NACA 23012 AIRFOIL/ALPHA=0,2,4,6,8,10,12,14/RN=3.0/MACH NO.=0.2/IWRITE=3

```
      COMMON XCT(100),ZCT(100),THETA(65),S(65),/OV(65),CP(65),GV(65),CFI      M1  1
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBD(65),TITLE(20),XT                M1  2
     1RAN(2),TRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS                M1  3
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),JWRITE,MAXK,IALPHA                        M1  4
      DIMENSION ALPHA(10),FSMACH(5),XTRSV(2),ZTRSV(2),LTRSV(2)                M1  5
      REAL KF                                                                 M1  6
      COMMON /TEMP/XTEMP(100),ZTEMP(100)                                      M1  7
      COMMON /DRAG/THTUP,HUP,THTLO,HLO,CDSY1                                   M1  8
      COMMON /INOUT/JREAD,JWRITE                                              M1  9
      JREAD=1                                                                 M1 10
      JPUNCH=2                                                                M1 11
      JWRITE=3                                                                M1 12
    5 IERR=0                                                                  M1 13
C***  READ THE SPECIFIED AIRFOIL GEOMETRY AND OTHER PERTINENT DATA           M1 14
      CALL READI(NA,ALPHA,NM,FSMACH,IPUNCH)                                   M1 15
C***  OBTAIN A SET IF 65 DISTRIBUTED POINTS FROM INPUT AIRFOIL POINTS        M1 16
      CALL GEOMI(IERR,XNOSE,ZNOSE)                                           M1 17
      IF (IERR.NE.0) GO TO 5                                                  M1 18
      IFLAG=0                                                                 M1 19
      DO 10 JJ=1,65                                                           M1 20
      DLTAS(JJ)=0.0                                                           M1 21
   10 DLTASP(JJ)=0.0                                                          M1 22
      DO 15 JJ=1,2                                                            M1 23
      LTRSV(JJ)=LTRAN(JJ)                                                     M1 24
      XTRSV(JJ)=XTRAN(JJ)                                                     M1 25
   15 ZTRSV(JJ)=ZTRAN(JJ)                                                     M1 26
      CDSY1=0.0                                                               M1 27
C***  SOLVE FOR THE GVBT'S, AIRFOIL BASIC THICKNESS GAMMA DISTRIBUTION       M1 28
      CALL VGVBT(IERR,IFLAG)                                                  M1 29
      IF (IERR.NE.0) GO TO 5                                                  M1 30
      DO 20 I=1,65                                                            M1 31
   20 GVBT(I)=GVBD(I)                                                         M1 32
C***  LOOP ON ALPHA'S (MAX.=10) INSIDE LOOP ON MACH NUMBERS (MAX.=5)         M1 33
      DO 90 I=1,NM                                                            M1 34
      FSMCH=FSMACH(I)                                                         M1 35
      DO 55 J=1,NA                                                            M1 36
      NALPHA=J                                                                M1 37
      ALFA=ALPHA(J)/57.2957795                                               M1 38
      DO 25 L=1,65                                                            M1 39
      CF(L)=0.                                                                M1 40
      DLTASI(L)=0.                                                            M1 41
      DLTASP(L)=0.                                                            M1 42
   25 GVBDP(L)=0.                                                             M1 43
      GVBD(L)=0.                                                              M1 44
      CDSY1=0.C                                                               M1 45
      MAXK=5                                                                  M1 46
C***  LOOP FOR ITERATION BETWEEN POTENTIAL AND VISCOUS FLOW SOLUTIONS        M1 47
      DO 35 K=1,MAXK                                                          M1 48
      DO 30 JJ=1,2                                                            M1 49
      LTRAN(JJ)=LTRSV(JJ)                                                     M1 50
      XTRAN(JJ)=XTRSV(JJ)                                                     M1 51
   30 ZTRAN(JJ)=ZTRSV(JJ)                                                     M1 52
C***  OBTAIN POTENTIAL SOLUTION (MAIN2) AND VISCOUS SOLUTION (MAIN3)         M1 53
      CALL MAIN2(IERR,K)                                                      M1 54
      IF (IERR.NE.0) GO TO 95                                                 M1 55
      CALL MAIN3(K)                                                           M1 56
   35 CONTINUE                                                                M1 57
      IF (IWRITE.GT.2) GO IC 55                                               M1 58
      WRITE (JWRITE,40)                                                       M1 59
                                                                             M1 60

   40 FORMAT (///2X,50HLONGEST CHORDLINE SYSTEM    BASIC REFERENCE SYSTE      M1 61
     M,//,7X,3HX/C,9X,3HZ/C,12X,3HX/C,9X,3HZ/C,7X,2HCP,8X,2HCF,,              M1 62
     15X,8HDELTAS/C,4X,4HV/VD,/)                                             M1 63
      CALL TRANS(XTEMP/ZTEMP,XNOSE,ZNOSE,XCT,ZCT,-BETA,C.,0.,0.,65)          M1 64
      DO 45 JJ=1,65                                                          M1 65
      XOC=XCT(JJ)/CREF                                                        M1 66
      ZOC=ZCT(JJ)/CREF                                                        M1 67
      XREF=XTEMP(JJ)/CREF                                                     M1 68
      ZREF=ZTEMP(JJ)/CREF                                                     M1 69
      SOC=S(JJ)/CREF                                                          M1 70
      DOC=DLTAS(JJ)/CREF                                                      M1 71
      CFOC=ABS(CF(JJ))                                                        M1 72
      VVOC=SQRT(ABS(1.-CP(JJ)))                                              M1 73
   45 WRITE (JWRITE,50) XOC,ZOC,XREF,ZREF,SOC,CP(JJ),CFOC,DOC,VVOC           M1 74
   50 FORMAT (1X,F11.5,F12.5,F15.5,2F12.5,4F10.5)                            M1 75
   55 CONTINUE                                                                M1 76
      WRITE (JWRITE,60) TITLE,FSMCH                                           M1 77
   60 FORMAT (///,5X,20A4,//,30X,8HFSMACH =,F9.5,//,4X,77(1H*),//,4X,1H*,4X   M1 78
     1,5HALPHA,12X,2HCL,13X,2HCD,10X,8HCM(NOSE),4X,13HCM(1/4-CHORD),2X,1      M1 79
     1H*,//,4X,1H*,75X,1H*)                                                  M1 80
      WRITE (JWRITE,65) (ALPHA(JK),CLL(JK),CD1(JK),CP1(JK),CM14(JK),JK=1      M1 81
     1,NA)                                                                    M1 82
   65 FORMAT (4X,1H*,1X,F10.6,4F15.6,4X,1H*)                                 M1 83
      WRITE (JWRITE,70)                                                       M1 84
   70 FORMAT (4X,77(1H*))                                                     M1 85
C***  PUNCH TABLE OF ANGLE OF ATTACK, LIFT CCEFF., DRAG COEFF., ANC          M1 86
C***  1/4-CHORD MOMENT COEFF. FOR 2-D TO 3-D PROGRAM                         M1 87
      IF (IPUNCH.EQ.0) GO TO 90                                              M1 88
      WRITE (JPUNCH,75) TITLE                                                 M1 89
   75 FORMAT (20A4)                                                          M1 90
      WRITE (JPUNCH,80) NA                                                    M1 91
   80 FORMAT (I10)                                                           M1 92
      WRITE (JPUNCH,85) (ALPHA(II),CLI(II),CDI(II),CPI4(II),II=1,NA)         M1 93
   85 FORMAT (4G16.5)                                                        M1 94
   90 CONTINUE                                                                M1 95
      GO TO 5                                                                 M1 96
   95 KKK=K-1                                                                 M1 97
      WRITE (JWRITE,100) KKK                                                  M1 98
  100 FORMAT (//,50HOERROR RETURNED FROM SUBROUTINE MAIN2 ON ITERATION,12     M1 99
     1),                                                                     M1100
      GO TO 5                                                                 M1101
      END                                                                     M1102

      SUBROUTINE READI(NN,ALPHA,NM,FSMACH,IPUNCH)                            RED  1
C***  ROUTINE TO READ ALL THE INPUT                                          RED  2
      REAL KF                                                                 RED  3
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),GV(65),CFI      RED  4
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBD(65),TITLE(20),XT                RED  5
     1RAN(2),TRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS                RED  6
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),JWRITE,MAXK,IALPHA                        RED  7
      COMMON /INPUT/NXLO(2),NXUP(2),NA                                        RED  8
      COMMON /INOUT/JREAD,JWRITE                                              RED  9
      DIMENSION ALPHA(10),FSMACH(5)                                          RED 10
      DATA XENO/4HENO /                                                       RED 11
      READ (JREAD,80) TITLE                                                   RED 12
      IF (TITLE(1).EQ.XEND) CALL EXIT                                        RED 13
      WRITE (JWRITE,5) TITLE                                                  RED 14
    5 FORMAT (1H1,50X,26H*****    CASE INPUT    *****,//25X,20A4)            RED 15
```

```
C
C     SUBROUTINE GEOM(IERR,XNOSE,ZNOSE)
C     ROUTINE TO CONVERT GEOMETRY ARRAYS TO COMPUTATIONAL FORM
C
      REAL KF
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),GV(65),CFI
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBDP(65),TITLE(20),XT
     1RAM(2),ZTRAM(2),TTRAM(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS
     1TAG,TSTAG,SSTAG,BETA,LTRAM(2),IWRITE,MAXK,IALPHA
      COMMON /INPUT/XRLO(2),NXUP(2),NA
      COMMON /TEMP/XTEMP(100),ZTEMP(100)
      COMMON /MOMENT/DXNOSE,DZNOSE
      COMMON /INOUT/JREAD,JWRITE
      REAL*8 A
      DIMENSION XU(65),ZU(65),ZL(65),XUD(65),ZUD(65),XLD(65),ZLD(65),ZL0I
     1(65),YSQUP(65),YSQLO(65),SU(65),XSUMU(65),DUM(11),DUM2(11),NXLOT(2),
     1NXUP(2),AI(4)
      EQUIVALENCE (XU(1),THETA(1)),(ZU(1),S(1)),(XL(1),VOV(1)),(ZL(1),CP
     1(1)),(YSQUP(1),GV(1)),(YSQLO(1),CF(1)),(SU(1),DLTAS(1)),(XSUMU(1)
     1,GVBT(1)),(XUD(1),XLD(1)),(DLTASP(1),ZUD(1)),(GVBD(1),ZLD(1)),GVBD
     1(1))
      DATA PI/3.1415927/
      IOPT=1
      IF (IWRITE.GT.-1) GO TO 75
      N1=NXUP(1)
      N2=NXLO(2)
      DO 5 I=N1,N2
      XTEMP(I)=XCT(I)
    5 ZTEMP(I)=ZCT(I)
      N1=NXUP(1)
      N2=NXUP(2)
      N3=NXLO(1)
      N4=NXLO(2)
      NA=N2-N1
      NB=N4-N3
      NBEG=MIN0(N1,N3)
      NEND=MAX0(NA,NB)+NBEG
C     PRINT INPUT POINTS (IOPT=1) OR DISTRIBUTED POINTS (IOPT=2)
      DO 70 J=NBEG,NEND
      IF (NBEG-J) 50,15,15
   15 WRITE (JWRITE,20) TITLE
   20 FORMAT (1H1,19X,20A4,/)
      IF (IOPT.EQ.2) GO TO 30
      WRITE (JWRITE,25)
   25 FORMAT (44X,20HINPUT AIRFOIL POINTS,/)
      GO TO 40
   30 WRITE (JWRITE,35)
   35 FORMAT (41X,26HDISTRIBUTED AIRFOIL POINTS,/)
   40 WRITE (JWRITE,45)
   45 FORMAT (28X,13HUPPER SURFACE,27X,13HLOWER SURFACE/24X,2HXU,18X,2HZ
     1U,18X,2HXL,18X,2HZL,/)
   50 IF (N1.GT.N2) GO TO 55
      XTN1=XTEMP(N1)/CREF
      ZTN1=ZTEMP(N1)/CREF
      XTN3=XTEMP(N3)/CREF
      ZTN3=ZTEMP(N3)/CREF
      IF (N3.LE.N2.AND.(N3.LE.N4)) WRITE (JWRITE,60) XTN1,ZTN1,XTN3,ZT
     1N3
   60 FORMAT (10X,4F20.6)
```

```
   10 WRITE (JWRITE,10)
      FORMAT (35H0 NXU NXL IWRITE IALPHA IPUNCH)
      READ (JREAD,85) NXU,NXL,IWRITE,IALPHA,IPUNCH
   15 WRITE (JWRITE,15) NXU,NXL,IWRITE,IALPHA,IPUNCH
      FORMAT (1X,3I5,2X,I5,3X,I5/)
      NU1=1
      NL1=NU2+1
      NU2=NXU
      NL2=NXL+NU2
      READ (JREAD,90) (XCT(L),L=NU1,NU2)
   20 WRITE (JWRITE,20) (XCT(L),L=NU1,NU2)
      READ (JREAD,90) (ZCT(L),L=NU1,NU2)
   25 WRITE (JWRITE,25) (ZCT(L),L=NU1,NU2)
      READ (JREAD,90) (XCT(L),L=NL1,NL2)
   30 WRITE (JWRITE,30) (XCT(L),L=NL1,NL2)
      READ (JREAD,90) (ZCT(L),L=NL1,NL2)
   35 WRITE (JWRITE,35) (ZCT(L),L=NL1,NL2)
   20 FORMAT (20X,5HXU = ,8E13.6/(25X,8E13.6))
   25 FORMAT (20X,5HZU = ,8E13.6/(25X,8E13.6))
   30 FORMAT (20X,5HXL = ,8E13.6/(25X,8E13.6))
   35 FORMAT (20X,5HZL = ,8E13.6/(25X,8E13.6))
      NXUP(1)=NU1
      NXLO(1)=NL1
      NXUP(2)=NU2
      NXLO(2)=NL2
      READ (JREAD,85) NA
   40 READ (JREAD,90) (ALPHA(I),I=1,NA)
      WRITE (JWRITE,40) NA,(ALPHA(I),I=1,NA)
      FORMAT (7H0 NA = ,I2,5X,08HANGLES OF ATTACK W.R.T. REFERENCE LINE
     1(IALPHA=0) OR W.R.T. LONGEST CHORDLINE (IALPHA=1),//,16X,6(E13.6,3
     1X)/,(4X,4(E13.6,3X)))
      READ (JREAD,85) NM
   45 READ (JREAD,90) (FSMACH(I),I=1,NM)
      WRITE (JWRITE,45) NM,(FSMACH(I),I=1,NM)
   45 FORMAT (7H0 NM = ,I3,16X,6HFSMACH,12X,1H=,5E13.6)
      READ (JREAD,90) CREF,SF
      DO 50 I=1,NL2
      XCT(I)=XCT(I)*SF
   50 ZCT(I)=ZCT(I)*SF
      READ (JREAD,90) TO,RN,PR,KF
   55 WRITE (JWRITE,55) CREF,SF,TO,RN,PR,KF
      FORMAT (8H0CREF = ,E13.6,8H  SF = ,E13.6,8H  TO = ,E13.6,8H  RN
     1 = ,E13.6,8H  PR = ,E13.6,8H  KF = ,E13.6)
      WRITE (JWRITE,60)
   60 FORMAT (//,18X,5HLTRAM,5X,5HXTRAM,11X,5HZTRAM/)
      DO 65 J=1,2
      READ (JREAD,95) LTRAM(J),XTRAM(J),ZTRAM(J)
   65 WRITE (JWRITE,70) (LTRAM(J),XTRAM(J),ZTRAM(J),J=1,2)
   70 FORMAT (1X,13HUPPER SURFACE,5X,I2,5X,E13.6,3X,E13.6/1X,13HLOWER SU
     1RFACE,5X,I2,5X,E13.6,3X,E13.6)
      DO 75 J=1,2
      XTRAM(J)=XTRAM(J)*SF
   75 ZTRAM(J)=ZTRAM(J)*SF
      NNA=NA
      RETURN
   80 FORMAT (20A4)
   85 FORMAT (5I5)
   90 FORMAT (8F10.0)
   95 FORMAT (I10,3F10.0)
      END
```

276

```
      IF (N1.GT.N2) WRITE (JWRITE,65) XTN3,ZTN3
   65 FORMAT (50X,"",2G.6)
      IF (N3.GT.N4) WRITE (JWRITE,60) XTN1,ZTN1
      N1=N1+1
   70 N3=N3+1
      IF (IOPT.GT.1) RETURN
   75 IOPT=2
      NXLOT(1)=1
C***  ARRANGE THE X AND Z ARRAYS TO GO FROM THE LOWER SURFACE TRAILING
C***  EDGE TO THE UPPER SURFACE TRAILING EDGE
      N1=NXLOT(1)
      N2=NXLOT(2)
      N=N2
      K=0
      DO 80 J=N1,N2
      K=K+1
      XTEMP(K)=XCT(N)
      ZTEMP(K)=ZCT(N)
   80 N=N-1
      NXLOT(2)=K
      N1=NXUPT(1)
C***  IF UPPER SURFACE DUPLICATES I.E. POINT DELETE THE DUPLICATE
      IF (XTEMP(K)-XCT(N1))  90,85,90
   85 NXUPT(1)=K+1
   90 NXUPT(1)=K+1
      NNOSE=K
      DO 95 J=N1,N2
      K=K+1
      XTEMP(K)=XCT(J)
      ZTEMP(K)=ZCT(J)
   95 ZTEMP(K)=ZCT(J)
      NXUPT(2)=K
      N1=NXLOT(1)
      N2=NXLOT(2)
      N3=NXUPT(2)
      DO 100 J=N2,N3
      NMAX=J-1
C***  CALCULATE LONGEST CHORDLINE WHICH IS OF LENGTH C
      DDS2=SQRT((ZTEMP(J)-ZTEMP(N1))**2+(XTEMP(J)-XTEMP(N1))**2)
      IF (DDS2.LT.DDS1) GO TO 105
  100 DDS1=DDS2
  105 NXLOT(2)=NMAX
      NXUPT(1)=NMAX+1
      N1=NXLOT(1)
      N2=NXLOT(2)
      NPTS=NXUPT(2)-N1+1
      N3=NXUPT(2)
      XN=XTEMP(N2)
      ZN=ZTEMP(N2)
      NNOSE=NN
      ZNOSE=ZN
      DZNOSE=(XTEMP(NMAX)-XTEMP(NNOSE))/CREF
      DZNOSE=(ZTEMP(NMAX)-ZTEMP(NNOSE))/CREF
      BETA=ATAN((ZTEMP(N1)+ZTEMP(N3))-2.*ZTEMP(N2))/((XTEMP(N3)-
     1 2.*XTEMP(N2))))
C***  TRANSLATE AND ROTATE INPUT POINTS TO LONGEST CHORDLINE SYSTEM SO
C***  THAT LEADING EDGE IS AT (0,0) AND TRAILING EDGE IS AT (C,0)
      CALL TRANS(XTEMP,ZTEMP,0.,0.,XTEMP,ZTEMP,BETA,XN,ZN,NPTS)
      C=(XTEMP(N1)+XTEMP(N3))-2.*XTEMP(N2))/2.

      NN1=NXLOT(1)
      NN2=NXUPT(2)
      NPTS=NN2-NN1+1
      NL=NXLOT(2)-NXLOT(1)+1
      NU=NXUPT(2)-NXUPT(1)+2
      IF (NL.LE.65.AND.NU.LE.65) GO TO 115
      WRITE (JWRITE,110) NU,NL
  110 FORMAT (/,90H0IN SUBROUTINE GEOM AFTER ROTATION OF DATA POINTS TO
     1LONGEST CHORDLINE SYSTEM, THERE WERE ,I2,25H UPPER SURFACE POINTS
     1AND,I3,6H LOWER/,125HOSURFACE POINTS. IF EITHER EXCEEDS 65, THEN W
     1ORKING ARRAY SIZE HAS BEEN EXCEEDED. SOLUTION -- DELETE (NU-65) OR
     1(NL-65) INPUT/,49HOPOINTS FROM THE SURFACE WHICH EXCEEDS 65 POINT
     15.)
      IERR=1
      RETURN
  115 JT=NXLOT(2)+1
      DO 120 NK=1,NL
      JT=JT-1
      XL(NK)=XTEMP(JT)
  120 ZL(NK)=ZTEMP(JT)
      JT=NXLOT(2)
      XU(1)=XTEMP(JT)
      ZU(1)=ZTEMP(JT)
      DO 125 NK=2,NU
      XU(NK)=XTEMP(JT)
  125 ZU(NK)=ZTEMP(JT)
C***  BEGIN PROCEDURE TO CALCULATE 65 SOLUTION POINTS. POINTS ARE
C***  DISTRIBUTED SO THAT (1) MORE POINTS ARE PLACED IN REGIONS OF LARGE
C***  SURFACE CURVATURE AND (2) UPPER AND LOWER SURFACE X'S ARE THE SAME
      XINTEG=0.25
      IUP=NU
      YSQU(1)=ZU(1)**2
      DO 145 I=2,NU
      IF (ZU(I)**2*ZU(I-1).LE.0.0.OR.XU(I)/C.GT.0.8) GO TO 130
      GO TO 135
  130 IF (IUP.EQ.NU.AND.I.NE.2)IUP=I-1
  135 IF (I.GT.IUP) GO TO 140
      YSQU(I)=ZU(I)**2
  140 IF (XU(I-1).GE.XU(I)) GO TO 170
  145 CONTINUE
      ILO=NL
      DO 165 I=2,NL
      IF (ZL(I)**2*ZL(I-1).LE.0.0.OR.XL(I)/C.GT.0.8) GO TO 150
      GO TO 155
  150 IF (ILO.EQ.NL.AND.I.NE.2)ILO=I-1
  155 IF (I.GT.ILO) GO TO 160
      YSQL(I)=ZL(I)**2
  160 IF (XL(I-1).GE.XL(I)) GO TO 170
  165 CONTINUE
      GO TO 180
  170 WRITE (JWRITE,175)
  175 FORMAT (//,5X,55HIN SUBROUTINE GEOM WHERE X/C IS NOT STRICTLY INCR
     1EASING)
      IERR=1
      RETURN
  180 IF (XU(IUP).LE.XL(ILO)) GO TO 185
      IUP=IUP-1
      GO TO 180
```

```
105   SU(1)=0.0
      DO 190 I=2,MU
190   SU(I)=SU(I-1)+SQRT(((XU(I)-XU(I-1))**2+(ZU(I)-ZU(I-1))**2)
      IUP1=IUP-1
      XSUMU(1)=0.0
      DO 200 I=1,IUPM1
      CALL POINT(RU,YSQUP,XU(I),YDUM,1,IUP,A)
      ADUM=(A(3)*RU(I)+0.5*A(2))*2
      XKUP=DABS(A(3)*YSQUP(I)-ADUM)/(SQRT(YSQUP(I)+ADUM))**3)
      CALL POINT(XL,YSQLO,XU(I),YSQDUM,1,ILO,A)
      ADUM=(A(3)*RU(I)+0.5*A(2))**2
      XKLO=DABS(A(3)*YSQDUM-ADUM)/(SQRT(YSQDUM+ADUM))**3)
      IF (XKLO.GT.XKUP)XKUP=XKLO
      IF (I.EQ.1) GO TO 195
      XSUMU(I)=XSUMU(I-1)+((XKUP+XKUP)*(SU(I)-SU(I-1))*XINTEG)*(SU(I)-SU(I-1))/2.0
195   XKUPP=XKUP
200   CONTINUE
      DO 205 I=IUP,MU
      CALL POINT(RU,ZU,XU(I),YDUM,1,MU,A)
      XKUP=DABS(2.0*A(3))/(DSQRT(1.0+(2.0*A(3)*XU(I)+A(2))**2))**3
      CALL POINT(XL,ZL,XU(I),YDUM,1,ML,A)
      XKLO=DABS(2.0*A(3))/(DSQRT(1.0+(2.0*A(3)*XU(I)+A(2))**2))**3
      IF (XKLO.GT.XKUP)XKUP=XKLO
205   XSUMU(I)=XSUMU(I-1)+((XKUP+XKUP)*(SU(I)-SU(I-1))*XINTEG)*(SU(I)-SU(I-1))/2.0
      XKUPP=XKUP
      DEL=XSUMU(MU)/32
      XSUMA=DEL
      XUD(1)=0.0
      ZUD(1)=0.0
      XLD(1)=0.0
      ZLD(1)=0.0
      DO 220 I=2,33
      M=2
      GO TO 215
210   M=1
215   CALL FTLUP(XSUMA,SS,M,MU,XSUMU,SU)
      CALL FTLUP(SS,XUD(I),M,MU,SU,XU)
      CALL FTLUP(XUD(I),ZUD(I),M,MU,XU,ZU)
      CALL FTLUP(XUD(I),ZLD(I),M,ML,XL,ZL)
      IF (I.EQ.1.OR.M.EQ.1) GO TO 220
      IF (XUD(I).LE.XUD(I-1)) GO TO 210
      JT=1
220   XSUMA=XSUMA+DEL
      JT=34
      DO 225 NK=1,33
      JT=JT-1
      XCT(NK)=XLD(JT)
      ZCT(NK)=ZLD(JT)
      XCTNK=2.0*XCTNK/XLD(33)-1.0
      IF (XCTNK.GT.-1.)XCTNK=-1.0
      IF (XCTNK.LT.-1.)XCTNK=-1.0
225   THETA(NK)=-1.*PI-ARCOS(XCTNK)
      JT=1
      DO 230 NK=34,65
      JT=JT+1
      XCT(NK)=XUD(JT)
      ZCT(NK)=ZUD(JT)
      XCTNK=2.0*XCTNK/XUD(33)-1.0
      IF (XCTNK.GT.-1.)XCTNK=-1.0
      IF (XCTNK.LT.-1.)XCTNK=-1.0
230   THETA(NK)=-2.*PI-ARCOS(XCTNK)
```

```
      S(1)=0.0
      DO 235 J=2,45
      DELA=SQRT((ZCT(J)-ZCT(J-1))**2+(XCT(J)-XCT(J-1))**2)
235   S(J)=S(J-1)+DELA
      DO 250 KL=1,2
      IF (LTRAN(KL).EQ.0) GO TO 250
      DUM1(1)=XTRAN(KL)
      DUM2(1)=ZTRAN(KL)
      CALL TRANS(DUM1,DUM2,0.,0.,DUM1,DUM2,BETA,XM,ZN,1)
      XTRAN(KL)=DUM1(1)
      ZTRAN(KL)=DUM2(1)
      NN1=NLOT(1)
      NN2=NLOT(2)
      ARG=2.0*(DUM1(1)-XTEMP(NN1))/(XTEMP(NN2)-XTEMP(NN2))-1.0
      IF (ABS(ARG).GT.1.)ARG=SIGN(1.,ARG)
      TNTR2=ARCOS(ARG)
      NN1=NKUPT(2)
      ARG=2.0*(DUM1(1)-XTEMP(NN2))/(XTEMP(NN2)-XTEMP(NN2))-1.0
      IF (ABS(ARG).GT.1.)ARG=SIGN(1.,ARG)
      TNTR1=2.0*PI-ARCOS(ARG)
      CALL POINT(THETA,ZCT,TNTR1,ZTR1,2,65,A)
      CALL POINT(THETA,ZCT,TNTR2,ZTR2,2,65,A)
      IF (ABS(DUM2(1)-ZTR1).GT.ABS(DUM2(1)-ZTR1))     240,245,245
240   TTRAN(KL)=TNTR2
      GO TO 250
245   TTRAN(KL)=TNTR1
250   CONTINUE
      IF (IWRITE.GT.1) RETURN
      N1=1
      N2=65
      DO 255 I=N1,N2
      XTEMP(I)=XCT(I)
255   ZTEMP(I)=ZCT(I)
      GO TO 10
      END
```

```
      SUBROUTINE MAIN2(IERR,K)
C000  SOLUTION FOR AIRFOIL GAMMAS IS CONSIDERED TO BE THE SUM OF A
C000  CAMBER SOLUTION AND A THICKNESS SOLUTION. THE CAMBER SOLUTION IS
C000  OBTAINED FROM THE BASIC AIRFOIL AT AN ANGLE OF ATTACK WITH ITS
C000  MEAN CAMBER LINE MODIFIED BY THE DIFFERENCE IN THE UPPER AND LOWER
C000  SURFACE BOUNDARY LAYER DISPLACEMENT THICKNESSES. THE THICKNESS
C000  SOLUTION IS OBTAINED FROM A SYMMETRIC AIRFOIL AT ZERO ANGLE OF
C000  ATTACK WHICH HAS THE SAME THICKNESS DISTRIBUTION AS THE BASIC
C000  AIRFOIL PLUS UPPER AND LOWER BOUNDARY LAYER DISPLACEMENT THICK-
C000  NESSES. SINCE THE CAMBER SOLUTION NECESSARILY INCLUDES A THICKNESS
C000  EFFECT ALSO, IT IS MODIFIED BY SUBTRACTING THE SOLUTION OF A
C000  SYMMETRIC AIRFOIL AT ZERO ANGLE OF ATTACK WITH THE SAME THICKNESS
C000  DISTRIBUTION AS THE BASIC AIRFOIL.
      REAL KF
      COMMON XCT(100),ZCT(100),THETA(65),S(65),YDV(65),CP(65),CFI
     1(65),DLTA5(65),GVDT(65),DLTASP(65),GVBD(65),GVBDP(65),TITLE(20),XT
     1RAN(2),ZTRAN(2),YTRAN(2),ALFA,PSCM,P0,T0,RM,PR,KF,CREF,C,XSTAG,ZS
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),IWRITE,NALK,IALPHA
```

```
        IPFLAG=0
        CALL CAMBER(IERR)
        IF (IERR.NE.0) RETURN
        XM=FSMCH
        IF (K.EQ.1) GO TO 10
        IFLAG=1
        CALL VOVBT(IERR,IFLAG)
        IF (IERR.NE.0) RETURN
        FI=1.
        IF (K.EQ.2)FI=2./3.
        DO 5 I=1,65
C***  GV'S (GAMMAS FROM CAMBER SOLUTION), GVBD'S (GAMMAS FROM BASIC
C***  THICKNESS PLUS BOUNDARY LAYER DISPLACEMENT), & GVBT'S (GAMMAS FROM
C***  BASIC THICKNESS SOLUTION) ARE AVERAGED WITH PREVIOUS GAMMAS FROM
C***  BASIC THICKNESS PLUS BOUNDARY LAYER (GVBDP'S) TO AID CONVERGENCE.
        GV(I)=GV(I)+(2.*GVBD(I)+GVBDP(I))/3.-GVBT(I))*FI
    5   GVBDP(I)=GVBD(I)
        DO 15 I=1,65
   15   DLTASP(I)=DLTAS(I)
C***  CONVERT INCOMPRESSIBLE GAMMAS TO COMPRESSIBLE GAMMAS
        CALL COMPR(IPFLAG,XM,K)
C***  LOCATE POSITION OF NOSE STAGNATION POINT
        CALL STAG
        IF (XM.GT.0.01) RETURN
        IPFLAG=1
        XM=0.01
        CALL COMPR(IPFLAG,XM,K)
        RETURN
        END

        SUBROUTINE VOVBT(IERR,IFLAG)
C***  ROUTINE TO CALCULATE GAMMAS FOR A SYMMETRIC AIRFOIL AT ZERO ALPHA
        REAL KF
        COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),CV(65),CFI
       1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBDP(65),TITLE(20),XT
        ...
        COMMON /TEMP/XTEMP(100),ZTEMP(100),ZTEPP(100)
        COMMON /COEF/AA(65,65)
        COMMON /DRAG/TMTUP,MUP,TMTLO,MLO,COSY1
        COMMON /INOUT/JREAD,JWRITE
        ZINFIN=0.25*COSY1*CREF
        IFLA=IFLAG
        TT=ZCT(65)-ZCT(1)
        IF (TT.GT.0.00011)IFLAG=1
        DX=0.25
        NDX=0
        IF (IFLAG) 10,10,5
    5   DX=(XCT(65)-XCT(64))/C
        NDX=0.25/DX-0.1
        IF (NDX.LT.5)NDX=5
        NDX=4*NDX
        IF (NDX.GT.30)NDX=30
        DX=C/NDX
        NX=33*NDX
        DO 20 I=1,NX
        IF (I.GT.33) GO TO 15

C***  GENERATE SYMMETRIC AIRFOIL Z'S FROM BASIC AIRFOIL Z'S AND BOUNDARY
C***  LAYER DISPLACEMENT THICKNESS
        J=I*32
        JJ=34-I
        XTEMP(I)=XCT(J)
        ZTEMP(I)=0.5*(ZCT(J)-ZCT(JJ)+(DLTAS(J)+DLTAS(JJ))/3.0+(DLTASP(J)+
       1DLTASP(JJ))/6.0
        GO TO 20
C***  EXTRAPOLATION PROCEDURE TO CLOSE THE SYMMETRIC AIRFOIL WHEN IT HAS
C***  A FINITE TRAILING EDGE THICKNESS
   15   XTEMP(I)=C*(I-33)*DX
        XX=XTEMP(I)/C-1.0
        ZTEMP(I)=(ZTEMP(33)-ZINFIN)*EXP(-6.9*XX)+ZINFIN*(1.0-XX)
   20   CONTINUE
        ZTEMP(NX)=0.0
        ZTEMP(1)=ZC
        GVBD(1)=ZC
C***  SOLUTION IS FOUND ONLY FOR HALF THE AIRFOIL SINCE IT IS SYMMETRIC
        NXM=NX-1
        DO 25 I=1,NXM
        XC=(XTEMP(I)+XTEMP(I+1))/2.0
        ZC=(ZTEMP(I)+ZTEMP(I+1))/2.0
        GVBD(I)=ZC
        DO 25 J=1,NXM
        CALL CHEM(XC,ZC,XTEMP(J),ZTEMP(J),XTEMP(J+1),ZTEMP(J+1),I,J,CUP)
        CALL CHEM(XC,ZC,XTEMP(J+1),-ZTEMP(J+1),XTEMP(J),-ZTEMP(J),1,0,CLO)
        AA(I,J)=CUP-CLO
   25   CALL SIMSOL(GVBD,NXM,IERR)
        IF (IERR.EQ.1) GO TO 5C
        IF (IFLAG.NE.0) GO TO 30
        GVBD(65)=GVBD(32)
        DS1=SQRT((XTEMP(33)-XTEMP(32))**2+(ZTEMP(33)-ZTEMP(32))**2)
        GO TO 35
   30   DS1=SQRT((XTEMP(34)-XTEMP(33))**2+(ZTEMP(34)-ZTEMP(33))**2)
        DS2=SQRT((XTEMP(33)-XTEMP(32))**2+(ZTEMP(33)-ZTEMP(32))**2)
        GVBD(65)=(GVBD(34)*DS1+GVBD(33)*DS2)/(DS1+DS2)
        DS1=DS2
   35   DO 40 I=1,31
        J=33-I
        JJ=65-I
        DS2=SQRT((XTEMP(J)-XTEMP(J-1))**2+(ZTEMP(J)-ZTEMP(J-1))**2)
        GVBD(JJ)=(GVBD(J)*DS1+GVBD(J-1)*DS2)/(DS1+DS2)
   40   DS1=DS2
        GVBD(33)=0.0
        DO 45 I=1,32
   45   GVBD(I)=-GVBD(J)
        J=66-I
        IFLAG=IFLA
        RETURN
   50   WRITE (JWRITE,55)
   55   FORMAT (/,6H0IN SUBROUTINE VOVBT A SINGULAR COEFFICIENT MATRIX WA
       1S GENERATED)
        RETURN
        END

C***  SUBROUTINE CAMBER(IERR)
C***  ROUTINE TO OBTAIN GAMMAS FOR THE AIRFOIL WITH MODIFIED CAMBER LINE
        REAL KF
        COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),CV(65),CFI,CAM
```

279

```
      SUBROUTINE CAMER(XC,ZC,X1,Z1,X2,Z2,I,J,COEF)
C***  ROUTINE TO CALCULATE THE COEFFICIENTS OF THE INFLUENCE MATRIX
      REAL*8 DX,DZ,DS,T1,T2,T3,T4,T5,T6,T7,DLOG,DATAN2,DSQRT,DBLE
      DX=DBLE(X2-X1)
      DZ=DBLE(Z2-Z1)
      DS=DSQRT(DX*DX+DZ*DZ)
      DX=DX/DS
      DZ=DZ/DS
      IF (I.EQ.J) GO TO 5
      T1=DBLE(X1-XC)
      T2=DBLE(Z1-ZC)
      T3=DBLE(X2-XC)
      T4=DBLE(Z2-ZC)
      T5=T1*DX+T2*DZ
      T6=T2*DX-T1*DZ
      COEF=SNGL((0.5D0*(T5*DLOG(T5*T5+T6*T6)-T4*DLOG(T4*T4+T6*T6))-T5+T4
     1+T6*(DATAN2(T6,T4)-DATAN2(T6,T5)))/6.283185400)
      RETURN
    5 COEF=SNGL(DS*(DLOG(0.5D0*DS)-1.0D0)/6.283185400)
      RETURN
      END
```

```
      SUBROUTINE COMPRI(PFLAG,XM,IN)
C***  ROUTINE TO CONVERT INCOMPRESSIBLE VELOCITY DISTRIBUTION TO
C***  COMPRESSIBLE VELOCITY DISTRIBUTION
      REAL KF
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),GV(65),CFI
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBDP(65),GVBDP(65),TITLE(20),XT
     1RAN(2),ZTRAN(2),ITRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),IWRITE,MAXK,IALPHA
      COMMON /IMOUT/JREAD,JWRITE
      DIMENSION ML(65)
      REAL*8 A(4)
      REAL ML,MLS
      DATA GAMMA/1.4/
      JN=IN-1
      MLS=5.
      DO 40 I=1,65
      XL=XM**2/(1.+SQRT(1.+XM**2))**2
      VOV(I)=ABS(GV(I))*(1.+(1.-XL)/(1.-XL*GV(I)**2))
      IF (VOV(I))  5,10,10
    5 VOV(I)=0.
   10 CONTINUE
      IF (XM)  25,25,15
   15 VOV=MLS/XM*SQRT((1.+(GAMMA-1./2.*XM*XM)/(1.+(GAMMA-1./2.*ML*ML
     151)
      IF (VOV(I)-VOVS)  25,25,20
   20 VOV(I)=VOVS
      MLS=ML.S+.1
   25 T=1.+(GAMMA-1.)/2.*XM**2*(1.-VOV(I)**2)
      ML(I)=XM*VOV(I))/SQRT(T)
```

```
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBDP(65),GVBDP(65),TITLE(20),XT
     1RAN(2),ZTRAN(2),ITRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),IWRITE,MAXK,IALPHA
      COMMON /TEMP/XTEMP(100),ZTEMP(100)
      COMMON /COEF/AA(65,65)
      COMMON /IMOUT/JREAD,JWRITE
      DIMENSION DUMT(33),DUMX(33),A(4)
      REAL*8 A
C***  SMOOTH UPPER AND LOWER BOUNDARY LAYER DISPLACEMENT THICKNESSES
      DO 5 IJ=1,33
      DUMT(IJ)=THETA(IJ)
    5 DUMX(IJ)=DLTAS(IJ)
      DO 10 K=1,3
   10 CALL SMOOTH(DUMT,DUMX,33)
      DO 15 IJ=1,33
   15 DLTAS(IJ)=DUMX(IJ)
      DO 20 IJ=1,33
      IA=66-IJ
      DUMT(IJ)=THETA(65)-THETA(IA)
   20 DUMX(IJ)=DLTAS(IA)
      DO 25 K=1,3
   25 CALL SMOOTH(DUMT,DUMX,33)
      DO 30 IJ=1,33
      IA=66-IJ
   30 DLTAS(IA)=DUMX(IJ)
      DO 35 J=1,33
      JJ=66-J
      D2C=2./3.*(DLTAS(JJ)-DLTAS(J))+(DLTASP(JJ)-DLTASP(J))/3.
C***  GENERATE Z'S FOR CAMBER SOLUTION BY ADDING THE DIFFERENCE OF UPPER
C***  AND LOWER BOUNDARY LAYER DISPLACEMENT THICKNESSES TO ORIGINAL Z'S
      ZTEMP(J)=ZCT(J)+D2C
   35 ZTEMP(JJ)=ZCT(JJ)+D2C
      SINA=SIN(ALFA-BETA)
      COSA=COS(ALFA-BETA)
      IF (IALPHA.EQ.0) GO TO 40
      SINA=SIN(ALFA)
      COSA=COS(ALFA)
   40 SRATIO=S(J)/(S(65)-S(64))
      DO 60 I=1,64
      XC=(XCT(I)+XCT(I+1))/2.0
      ZC=(ZTEMP(I)+ZTEMP(I+1))/2.0
      DO 50 J=1,64
      CALL CAMER(XC,ZC,XCT(J),ZTEMP(J),XCT(J+1),ZTEMP(J+1),I,J,COEFF)
   45 AA(I,J)=AA(I,J)-COEFF*SRATIO
      AA(I,J)=1.0
   50 GO TO 55
   55 GV(I)=-COSA*ZC+SINA*XC
   60 CONTINUE
      CALL SIMSOL(GV,64,IERR)
      IF (IERR.NE.0) GO TO 70
      GV(44)=GV(1)*SRATIO
      GVTE=(ABS(GV(64))+ABS(GV(1)))*SRATIO)/(1.0+SRATIO)
      GV(65)=SIGN(GVTE,GV(64))
      DO 65 I=2,64
      J=66-I
   65 GV(I)=(GV(J)*(S(J+1)-S(I))+GV(I)*(S(I)-S(J)))/(S(J+1)-S(J))
      GV(1)=-GV(65)
```

```
      RETURN
   70 WRITE (JWRITE,75)
   75 FORMAT (/,65H0IN SUBROUTINE CAMBER A SINGULAR COEFFICIENT MATRIX W
     1AS GENERATED)
      RETURN
      END
```

```
   30 CALL POINT(THETA,GV,THETA(J),DUM,J-1,J+1,A)
      SQ=DSQRT(DABS(A(2)*A(2)-4.*A(1)*A(3)))
      T1=(-A(2)-SQ)/(2.*A(3))
      T2=(-A(2)+SQ)/(2.*A(3))
      IF (ABS(THETA(J)-T1)-ABS(THETA(J)-T2)) 35,35,40
   35 TSTAG=T1
      GO TO 45
   40 TSTAG=T2
   45 XSTAG=C/2.*(1.+COS(TSTAG))
      CALL POINT(THETA,ZCT,TSTAG,ZSTAG,J-1,J+1,A)
      CALL POINT(THETA,S,TSTAG,SSTAG,J-1,J+1,A)
      RETURN
      END
```

```
      IF (XM) 35,30,35
   30 CP(I)=1.-VOV(I)**2
      GO TO 40
   35 CP(I)=-2./((GAMMA*XM**2)*(T**(GAMMA/(GAMMA-1.))-1.)
   40 CONTINUE
      IF (IPPFLAG.GT.0) GO TO 75
      IF (JM.GT.0) GO TO 75
      ALF=ALFA*57.29577795
      IF (((IM-1,LT,MAXK-1)) GO TO 75
      N1=1
      N2=33
      IF (IWRITE.GT.0) GO TO 75
      DO 55 K=1,2
      IF (K.EQ.1) WRITE (IWRITE,60) TITLE,XM,ALF,JN
      IF (K.EQ.1) GO TO 45
      WRITE (JWRITE,65) TITLE,XM,ALF,JN
   45 DO 50 J=N1,N2
      XOC=XCT(J)/CREF
      ZOC=ZCT(J)/CREF
      VT=ABS(GV(J))
   50 WRITE (JWRITE,70) XOC,ZOC,VT,VOV(J),ML(J),CP(J)
      N1=34
      N2=65
   60 FORMAT (1H1,50X,22HINVISCID FLOW SOLUTION,//,25X,20A4,//,32X,14HMA
     1CH NUMBER = ,F7.5,13X,18HANGLE OF ATTACK = ,F9.5,5H DEG.,//,110X,1
     13HLOWER SURFACE,//,9X,107X,16HITERATION NUMBER,13//,11X,1HX,19X,1HZ,1
     15X,10HV/VO)INC.,9X,12H(V/VO)COMPR.,13X,2HML,18X,2HCP,/)
   65 FORMAT (1H1,50X,22HINVISCID FLOW SOLUTION,//,25X,20A4,//,32X,14HMA
     1CH NUMBER = ,F7.5,13X,18HANGLE OF ATTACK = ,F9.5,5H DEG.,//,107X,1
     13HUPPER SURFACE,//,9X,107X,16HITERATION NUMBER,13//,11X,1HX,19X,1HZ,1
     15X,10HV/VO)INC.,9X,12H(V/VO)COMPR.,13X,2HML,18X,2HCP,/)
   70 FORMAT (4X,6(F12.5,8X))
   75 DO 80 I=1,65
   80 VOV(I)=ML(I)
      RETURN
      END
```

```
C***  SUBROUTINE STAG
C***  ROUTINE TO LOCATE POSITION OF NOSE STAGNATION POINT
      REAL KF
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),GV(65),CFI
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBD(65),GVBDP(65),TITLE(20),XT
     1RAN(2),ZTRAN(2),TTRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),IWRITE,MAXK,IALPHA
      DIMENSION A(4)
      REAL*8 A
      J=17
      IF (GV(J)) 10,5,15
    5 TSTAG=THETA(J)
      GO TO 45
   10 J=J-1
      IF (J.GT.49) GO TO 5
      IF (GV(J)) 10,5,20
   15 J=J+1
      IF (J.GT.49) GO TO 5
      IF (GV(J)) 20,5,15
   20 IF (ABS(GV(J))-ABS(GV(J-1))) 30,30,25
   25 J=J-1
```

```
      SUBROUTINE MAIN3(IN)
C***  ROUTINE FOR CONTROLLING BOUNDARY LAYER CALCULATIONS AND FOR
C***  PRINTING LOAD SUMMARY SHEET
      REAL KF
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),GV(65),CFI
     1(65),DLTAS(65),GVBT(65),DLTASP(65),GVBD(65),GVBDP(65),TITLE(20),XT
     1RAN(2),ZTRAN(2),TTRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS
     1TAG,TSTAG,SSTAG,BETA,LTRAN(2),IWRITE,MAXK,IALPHA
      REAL*8 A(4)
      COMMON /VISCOS/SUMS(43),AME(43),SLO(43),DDMCS(43),X(43),ZTE
     1MP(43),TINFX,ITEMP(43),ISURF
      COMMON /TURBTR/SCRIT,RECRIT,STRAN,RTRAN,THETAI,MINIT,AMETR,SLOTR,N
     14,
      COMMON /DRAG/THTUP,HUP,THTLO,HLO,COSY1
      COMMON /OUTPUT/CL1(10),CDL(10),CM1(10),NALPHA
      COMMON /MOMENT/DXNOSE,DZNOSE
      COMMON /INOUT/JREAD,JWRITE
      DIMENSION CMF(4)
      CL=0.
      CD=C.
      AO=49.02*SQRT(TO)
      K1=0
C***  LOOP FOR CALCULATION OF BOUNDARY LAYER ON UPPER (J=1) AND
C***  LOWER (J=2) SURFACES
      DO 60 J=1,2
      ISURF=J
      L=0
      GO TO (5,30),J
C     J=1 FOR UPPER SURFACE
    5 DO 25 K=1,65
      KK=K-K1+1
      IF (ABS(S(K)/CREF-SSTAG/CREF).LE.1.0E-06) GO TO 10
      IF (S(K)-SSTAG) 15,10,20
   10 L=1
   15 K1=K1+1
      GO TO 25
   20 SUMS(KK)=S(K)-SSTAG
      X(KK)=XCT(K)
      ITEMP(KK)=K
      ZTEMP(KK)=ZCT(K)
      AME(KK)=VOV(K)
   25 CONTINUE
      NP=KK
      IF (L.EQ.1)K1=K1-1
```

281

```
      GO TO 40
C     J=2  FOR LOWER SURFACE
30    DO 35 K=1,K1
      L=K
      KK=1OK1-K
      SUMS(K+1)=SSTAG-S(KK)
      X(K+1)=-XCT(KK)
      ITEMP(K+1)=KK
      ZTEMP(K+1)=ZCT(KK)
      AME(K+1)=VOV(KK)
      NP=L+1
35    SUMS(1)=-0.
40    AME(1)=-0.
      X(1)=XSTAG
      ZTEMP(1)=ZSTAG
      DO 45 K=1,NP
      CALL POINT(SUMS,AME,SUPS(K),VINT,1,NP,A)
      SLO(K)=A(2)+2.*A(3)*SUMS(K)
45    UE(K)=AD*AME(K)*SQRT(1.+0.2*AME(K)**2)
      DO 50 K=1,NP
      CALL POINT(SUMS,SLO,SUMS(K),VINT,1,NP,A)
50    DDMOS(K)=A(2)+2.*A(3)*SUMS(K)
      SEND=SUMS(NP)
      JN=IN-1
C***  CALCULATE LAMINAR BOUNDARY LAYER AND TRANSITION POINT
      CALL LAMNA2(NP,JS,JN,ITR,SEND)
      NI=6
      SEND=SUMS(NP)
      IF (ITR.NE.2) GO TO 55
C***  CALCULATE TURBULENT BOUNDARY LAYER
      CALL TURB2(NP,NI,J,JN,SEND)
55    NB=1
      IF (J.EQ.1)NB=59
      CALL LSQ(DLTAS(NB),S(NB),1,7,CNF)
      COR=DLTAS(NB)-CNF(1)-CNF(2)*S(NB)
      IF (J.EQ.2)COR=DLTAS(NB+6)-CNF(1)-CNF(2)*S(NB+6)
      DO 60 KK=1,7
      K=NB-1+KK
60    DLTAS(K)=CNF(1)+CNF(2)*S(K)+COR
      VOVC=SQRT(ABS(1.0-CP(I)))
      HTE=(DLTAS(1)-DLTAS(65))/(4*THTUP*THTLC)
      COSY1=2.C*(THTUP*THTLD/CREF*VOVO**(HTE+5.0)/2.0)
      IF (JN.NE.0.AND.JN.NE.MAXK-1) RETURN
C***  INTEGRATE VISCOUS AND PRESSURE FORCES TO OBTAIN NORMAL AND AXIAL
C***  FORCE COEFFICIENTS
      CALL LOAD(NP,CPI,CNV,CAP,CAF,CMP,CMF)
      CTERM=ALFA-BETA
      IF (IALPHA.EQ.1)CTERM=ALFA
      IF (JN.EQ.0)CDP=CMP*SIN(CTERM)+CAP*COS(CTERM)
      IF (JN.GE.2.AND.JN.NE.MAXK-1) RETURN
      IF (IWRITE.EQ.1.AND.JN.EQ.0) RETURN
C***  BEGIN MODIFICATION TO OBTAIN FINAL VALUES OF FORCE COEFFICIENTS
      CD=CD-CDP
      CL=CL-(1.0-.214*SQRT(ABS(CD)))
      CN=CL*COS(ALFA)+CD*SIN(ALFA)
      CA=CD*COS(ALFA)-CL*SIN(ALFA)
      IF (IALPHA.EQ.1) GO TO 65
      CNVM=CNV
      CNV=CNVM*COS(BETA)+CAF*SIN(BETA)
      CAF=-CNVM*SIN(BETA)+CAF*COS(BETA)
65    CMP=CM-CNV
      CAP=CA-CAF
      ALF=ALFA*57.2957795
      IF (IALPHA.EQ.1) GO TO 70
      CM=CM-CNP*DXNOSE+CA*DZNOSE
70    CM41=CM+CN/4.0
      IF (JN.NE.MAXK-1) GO TO 75
      CLI(NALPHA)=CL
      CDI(NALPHA)=CD
      CMI(NALPHA)=CM
      CM41(NALPHA)=CM41
75    IF (IWRITE.GT.2) RETURN
      BETA1=BETA*57.2957795
      CLP=CMP*COS(ALFA)-CAP*SIN(ALFA)
      CDP1=CMP*SIN(ALFA)+CAP*COS(ALFA)
      CLV=CNV*COS(ALFA)-CAF*SIN(ALFA)
      CDF=CNV*SIN(ALFA)+CAF*COS(ALFA)
      XTRANL=XTRAN(2)
      XTRANU=XTRAN(1)
      WRITE (IWRITE,80) JN,TITLE,FSMCH,CREF,RN,IALPHA,ALF,C,BETA
80    FORMAT (1H1,42X,36HLOAD SUMMARY SHEET--ITERATION NUMBER(3,//,23X,2
     10A4,//,12HOMACH NUMBER,7X,1H=,F9.5,16X,59HREFERENCE CHORD (BASIS F
     1OR ALL COEFFICIENT NORMALIZATION) =,F10.5,1X,3HFT.,//,20HOREYNOLDS
     1NO.   =,F9.5,8H MILLION8X,34HANGLE OF ATTACK (VALUE OF IALPHA =
     1,12,1H),21X,1H=,F10.5,1X,4HDEG.,//,48X,47HIIALPHA=O--ANGLE DEFINED
     1W.R.T. REFERENCE LINE)//,1X,19HLONGEST CHORDLINE =,F9.5,5H FEET,14
     1X,50HIIALPHA=1--ANGLE DEFINED W.R.T. LONGEST CHORDLINE)//,53HOANG
     1LE BETWEEN LONGEST CHORDLINE AND REFERENCE LINE =,F9.5,5H DEG.,58H
     1(POSITIVE FOR REFERENCE LINE BELOW THE LONGEST CHORDLINE),/)
      WRITE (IWRITE,85) CMP,CLP,CNV,CLV,CN,CL,CAP,CDP1,CAF,CDF,CA,CD,CMP
     1,CM,CMF,CM41,CM
85    FORMAT (58X,401H*,/,58X,1H*,/,20H NORMAL FORCE COEFFICIE
     1NTS,32X,19H= LIFT COEFFICIENTS20X,1H=,/,17X,12H(CN)PRESS.   =,F9.5,2
     10X,1H=,16X,12H(CL)PRESS.   =,F9.5,2H =,/,17X,12H(CN)SHEAR   =,F9.5,20
     1X,1H=,16X,12H(CL)SHEAR   =,F9.5,2H =,/,18X,2HCN,8X,1H=,F9.5,20X,1H=
     1,17X,2HCL,8X,1H=,F9.5,2H =,/,58X,1H=,//,25H AXIAL FORCE COE
     1FFICIENTS,20X,1H=,/,17X,12H(CA)PRESS.   =,
     1F9.5,20X,1H=,16X,12H(CD)PRESS.   =,F9.5,2H =,/,17X,12H(CA)SHEAR   =,
     1F9.5,20X,1H=,16X,12H(CD)SHEAR   =,F9.5,2H =,/,18X,2HCA,8X,1H=,F9.5,
     120X,1H=,17X,2HCD,8X,1H=,F9.5,2H =,/,58X,1H=,//,20H MOMENT C
     10EFFICIENTS,38X,1H=,20H MOMENT COEFFICIENTS,18X,1H=,/,2X,12HABOUT
     1NOSE(,3X,12H(CM)PRESS.   =,F9.5,20X,1H=,9X,14H(CM)ABOUT NOSE,4X,1H=
     1,F9.5,2H =,/,17X,12H(CM)SHEAR   =,F9.5,20X,1H=,9X,19H(CM)QUARTER CH
     1ORD =,F9.5,2H =,/,18X,2HCM,8X,1H=,F9.5,20X,1H=,//,58X,401H*,/
     1*),/)
90    FORMAT (52HODRAG COEFFICIENT COMPUTED BY SQUIRE-YOUNG FORMULA =,F9
     1.5,/,33HOZEROTH ITERATION PRESSURE DRAG =,F9.5,/,33HOTRANSITION PO
     1INTS: X/C(LOWER) =,F9.5,10X,12HX/C(UPPER) =,F9.5,/)
      RETURN
      END

      SUBROUTINE LAMNA2(NP,JS,JN,ITRAN,SEND)
```

```
C***  ROUTINE TO CALCULATE LAMINAR BOUNDARY LAYER AND TRANSITION POINT      LAM   2
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VDV(65),CP(65),GV(65),CF1  LAM   3
     1(65),DLTAS(65),GVBD(65),DLTASP(65),GVBDP(65),TITLE(20),XT           LAM   4
     1RAN(2),ZTRAN(2),TRAN(2),ALFA,F5MCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS   LAM   5
     1TAG,ISTAG,SSTAG,BETA,LTRAN(2),IWRITE,MARK,IALPHA                    LAM   6
      COMMON /TURBTR/SCRIT,RECRIT,STRAN,RTRAN,THETAI,HINIT,AMETR,SLOTR,N  LAM   7
     14,                                                                  LAM   8
      REAL*8 AA(4)                                                        LAM  10
      IF (AME(K))  20,20,25                                               LAM  11
      COMMON /VISCOS/SUMS(43),AME(43),SLO(43),DOMDS(43),X(43),ZTE         LAM  12
      IMP(43),TINF,X,ITEMP(43),ISURF                                      LAM  13
      COMMON /LAMTR/H(43),AMTK(43),REMI(43),UET1(43),UET2(43),CN(43),UET  LAM  14
     1(43),KBAR(43),RTHTR(43),REMOM(43),SG(43),SW,AO,VO,SEP,AMU,SQ,F,HL1  LAM  15
     1(43),K,KOM                                                          LAM  16
      COMMON /PLBLK/DISP                                                  LAM  17
      COMMON /INOUT/JREAD,JWRITE                                          LAM  18
      REAL KBAR,KF                                                        LAM  19
      DIMENSION IFLAG(6),COCUBE(4),DISP(43),AMTKOC(43),DISPOC(43),SSG(43  LAM  20
     1),XSAVE(3),AL(43),TAM(43),CFOMU(43),AMJORE(43),SKINFR(43),REW(43),  LAM  21
     1TEMSLO(43),CFBAR(43),CROOT(6)                                       LAM  22
      EQUIVALENCE (CNSTAG,CN(1)),(HLISTG,HL1(1)),(HSTAG,H(1))             LAM  23
      IF (LTRAN(JS).NE.1)XTRAN(JS)=-CREF                                  LAM  24
      FACT=1.0                                                            LAM  25
      AME1F=F5MCH                                                         LAM  26
      IF (F5MCH-.01)  5,5,10                                              LAM  27
    5 AME1F=-.01                                                          LAM  28
   10 DO 15 I=1,6                                                         LAM  29
   15 IFLAG(I)=0                                                          LAM  30
      KOM=0                                                               LAM  31
      SW=KF*(1.+.2*PR**(1./3.)*AME1F**2)/(1.+.2*AME1F**2)-1.              LAM  32
      SEP=-.0691058-.0395712*SW+.0671724*SW**2                           LAM  33
      CNSTAG=-.1064+.017250*SW*.375*SW**2                                LAM  34
      TWALL=TO*(1.+SW)                                                    LAM  35
      COCUBE(4)=-.7312+4.32497*SW+12.251*SW**2+21.89199*SW**3+30.92805*SW*  LAM  36
     1*4                                                                  LAM  37
      COCUBE(3)=-1.227559-1.662158*SW-9.1919*SW**2-13.197*SW**3-17.78815*  LAM  38
     1SW**4                                                               LAM  39
      COCUBE(2)=-.088925-.389889*SW+1.11802*SW**2-.990225*SW**3+1.21953    LAM  40
     1*SW**4                                                             LAM  41
      BB=-10.8327-3.2*SW+2.9611*SW**2-1.298*SW**3                         LAM  42
      CB=23.4938+292.987(*SW+418.165*SW**2                               LAM  43
      DB=-1748.459-8973.663*SW-13346.32*SW**2                            LAM  44
      EB=-633.05-3431(.95*SW-4980(.875*SW**2                             LAM  45
      A1=.44                                                              LAM  46
      B1=5.56903+2.51309*SW                                              LAM  47
      C1=3.195945-7.0807*SW                                              LAM  48
      D1=-6.358574-13.64784*SW                                          LAM  49
      CC=((1(.76-1*SEP-(1./12.)*CNSTAG)/(SEP*CNSTAG+9*(CNSTAG-SEP))      LAM  50
      HLISTG=HSTAG/(1.+SW)                                              LAM  51
      HSTAG=-1.1138*CNSTAG+2.38411)*1.185*SW                            LAM  52
      BT=-CC*CNSTAG+1./CNSTAG                                           LAM  53
      AO=49.02*SQRT(TO)                                                 LAM  54
      AMU=((TO+198.6)/(TO*(1.+SW)+198.6))*SQRT(1.+SW)                   LAM  55
      TINF=TO/(1.+.2*AME1F**2)                                          LAM  56
      AINF=49.02*SQRT(TINF)                                             LAM  57
      VRAT=(TINF/TO)**3.5*(TO+198.6)/(TINF+198.6)                       LAM  58
C***  A CORRECTION TO KINEMATIC VISCOSITY HAS BEEN MADE TO IMPROVE DRAG   LAM  59
C***  CORRELATION WITH TEST DATA.  THE VARIABLES INVOLVED IN CORRECTION   LAM  60
C***  ARE VO (CORRECTED KINEMATIC VISCOSITY) = 0.5 VOTRUE (TRUE K.V.)     LAM  61
      RHORAT=(1.+.2*AME1F**2)**2.5                                      LAM  61

      RMPFT=RN/CREF                                                     LAM  62
      VINF=AME1F*AINF/RMPFT/1.E06                                       LAM  63
      VOTRUE=VINF/VRAT/RHORAT                                           LAM  64
      PO=1716.*2.27*TO**2.5*1.E-8/(TO+198.6)/VOTRUE                     LAM  65
      VO=0.5*VOTRUE                                                     LAM  66
      A=1.-C1*CNSTAG**2-2.*D1*CNSTAG**3                                 LAM  67
      B=B1+2.*C1*CNSTAG+3.*D1*CNSTAG**2                                 LAM  68
      DO 30 K=1,NP                                                      LAM  69
      IF (AME(K))  20,20,25                                             LAM  70
   20 AME(K)=0.001                                                      LAM  71
   25 UET(K)=AO*AME(K)                                                  LAM  72
      UET1(K)=AO*SLO(K)*(1.+0.2*AME(K)**2)/AMU*C                        LAM  73
      UET2(K)=AO*(1.+.2*AME(K)**2)*SLO(K)**3/AMU**2*DOMDS(K)*C          LAM  74
      K=NP                                                              LAM  75
      UET2(K)=UET2(K)+1.6*AO*(1.+.2*AME(K)**2)*7/AMU*2*AME(K)*SLO(K)**  LAM  76
     12*C**2                                                           LAM  77
      K=2                                                              LAM  78
      DO 135 I=3,NP                                                    LAM  79
      IF (K.NE.2) GO TO 35                                             LAM  80
      CN(K)=(-A*SLO(K))/AME(K)*SUMS(K)/2.                              LAM  81
      DELCUT=0.                                                        LAM  82
      GO TO 40                                                         LAM  83
   35 K=1                                                              LAM  84
      FIG51=(1.+.2*AME(J)**2)**4                                       LAM  85
   40 SQ=AME(K)*AME(K)                                                 LAM  86
      FIG52=(1.+.2*SQ)**4                                              LAM  87
      KBAR(K)=0.                                                       LAM  88
      RTHTR(K)=0.                                                      LAM  89
      IF (K.EQ.2) GO TO 55                                             LAM  90
      IF (ABS(SLO(K))-1.E-6)  45,50,50                                 LAM  91
   45 SGN=SIGN(1.0,SLO(K))                                             LAM  92
      SLO(K)=1.E-6*SGN                                                 LAM  93
   50 CN(K)=SLO(K)*FIG52*CN(J)*AME(J)**8/AME(K)**8*B/(SLO(J)*FIG51)-A*AME1  LAM  94
     1K)*(-B)*SLO(K)*FIG52*(AME(K)/AME(J))**8*(B-1.)/FIG52*AME(J)**8-1./FIG51  LAM  95
     1*(SUMS(K)-SUMS(J)/2.)                                            LAM  96
   55 IF (CN(K)-CNSTAG*DELCUT)  60,75,75                               LAM  97
   60 IF (DELCUT)  70,65,70                                            LAM  98
   65 FACT=FACT*10.E-5                                                 LAM  99
   70 CN(K)=CNSTAG-DELCUT)*FACT                                        LAM 100
   75 IF (CN(K)-10.*SEP)  85,85,80                                     LAM 101
   80 CN(K)=10.*SEP                                                    LAM 102
   85 B=B1+2.*C1*CN(K)+3.*D1*CN(K)**2                                  LAM 103
      A=A1-C1*CN(K)**2-2.*D1*CN(K)**3                                  LAM 104
      F=1.+.17504*SQ+.6*SW-.42171*SW*SQ-.008832*SQ*SQ+.060343*SW*SQ*SQ+.  LAM 105
     11825)*SW+SW-.C73*SW+SW*SQ-.C073*SW*SW*SQ                         LAM 106
      TAW(K)=TO*(1.+.2*SQRT(PR)*SQ)/(1.+.2*SQ)                         LAM 107
      AMTK(K)=SQRT(ABS(VO*AMU*(-CN(K))/(AO*SLO(K)))*(1.+SW)*(1.+.2*SQ)  LAM 108
      ALPHA=1./3.+BT*CN(K)+CC*CN(K)**2                                 LAM 109
      H(K)=(-1.1138*CN(K)+2.38411)*(1.+.1.18*(1.+SW)*(1.+.2*SQ)*(1.-.)*(1.+.2*  LAM 110
     11PR*ALPHA*SQ                                                     LAM 111
      HL1(K)=(H(K)-PR*ALPHA*.2*SQ)/(1.+SW)/(1.+.2*SQ)                  LAM 112
      DISP(K)=AMTK(K)*H(K)                                             LAM 113
      AMTKOC(K)=AMTK(K)/C                                              LAM 114
      DISPOC(K)=DISP(K)/C                                              LAM 115
      COCUBE(1)=-.071501G-.04559*SW-.04871*SW**2-CN(K)                 LAM 116
      CALL PRDOT(COCUBE,CROOT)                                         LAM 117
      JSAVE=C                                                          LAM 118
      DO 90 IK=1,6,2                                                   LAM 119
      IF ((CROOT(IK).LT.0.0).OR.(CROOT(IK).GT.0.7)) GO TO 90           LAM 120
      JSAVE=JSAVE+1                                                    LAM 121
```

283

```
 90 CONTINUE
    XSAVE(JSAVE)=CROOT(IK)                                          LAM 122
    IF (JSAVE) 95,95,100                                            LAM 123
 95 AL(IK)=0.                                                       LAM 124
    GO TO 110                                                       LAM 125
100 XMIN=10.0                                                       LAM 126
    DO 105 IK=1,JSAVE                                               LAM 127
105 XMIN=AMIN1(XSAVE(IK),XMIN)                                      LAM 128
    AL(K)=XMIN                                                      LAM 129
110 SKIMPR(K)=2.-ALIK)*SQRT(ABS((SUMS(K))*(-SLO(K)/CN(K))/AME(K))) LAM 130
    CFOMU(K)=2.00+B0*CN(K)*C0*CN(K)**2+D0*CN(K)**3+E0*CN(K)**4      LAM 131
    AMUORE(K)=(SKIMPR(K)*PR*ALPHA)/CFOMU(K)                         LAM 132
    RERO(K)=A0*AMTK(K)*AME(K)/(F(G52*V0*ARU*(1.+SW)**2)             LAM 133
    RER(K)=A0*AMTK(K)*AME(K)/((1.+2*SQ)**4.5*V0*F)                  LAM 134
    REN(K)=RERO(K)*(SUMS(K))/AMTK(K)                                LAM 135
    TERSLO(K)=SQRT(REN(K))*AMUORE(K)*(TAW(K)-TWALL)/(SUMS(K))       LAM 136
    SG(K)=SKIMPR(K)/SQRT(RE(K))                                     LAM 137
    IF (K-2) 120,115,120                                            LAM 138
115 EVAL=SQ*(1.+.2*SQ)**(-4.5)*SG(K)*SUMS(K)/2.-/C                  LAM 139
    GO TO 125                                                       LAM 140
120 EVAL=EVAL+(SQ*(1.+.2*SQ)**(-6.5)*SG(K)+AME(J)**2*(1.+.2*AME(K)**2) LAM 141
    )*(-4.5)*SG(J))*(SUMS(K)-SUMS(J))/2./C                          LAM 142
125 CFBAR(K)=1./(1.+SW)*1./AME(F)**2*(1.+.2*AME(F)**2)**3.5/SUMS(K)*EVAL LAM 143
    I/C                                                             LAM 144
    SSG(K)=SG(K)/(1.-SW)*(AME(K)/AME(F)**2*(1.+.2*AME(K)**2)        LAM 145
    E(K)**2)**3.5/(1.-.2*AME(K)**2)                                 LAM 146
    IA=ITEMP(K)                                                     LAM 147
    DLTAS(IA)=DISP(K)                                               LAM 148
    CF(IA)=SSG(K)*(1.-CP(IA))                                       LAM 149
C*** CHECK FOR TRANSITION FROM LAMINAR TO TURBULENT BOUNDARY LAYER  LAM 150
    CALL BLTRAN(ITRAN,IFLAG,JS,NP,SEND)                             LAM 151
    IF (ITRAN) 35,135,130                                           LAM 152
130 N=K                                                             LAM 153
    NPT=K                                                           LAM 154
    GO TO 140                                                       LAM 155
135 CONTINUE                                                        LAM 156
140 IF (ITRAN.NE.1) CALL POINT(SUMS,X,STRAM,XTRAN(JS),1,NP,AA)      LAM 157
    IF ((J.GT.0).AND.(J.LT.MAXK-1)) GO TO 255                       LAM 158
    IF (ISURF.EQ.1.AND.JN.EQ.0) GO TO 255                           LAM 159
    IF (ISURF.GT.1) GO TO 140                                       LAM 160
    IF (ISURF.EQ.1) WRITE (JWRITE,145) TITLE,JN                     LAM 161
    IF (ISURF.EQ.1) GO TO 155                                       LAM 162
    WRITE (JWRITE,150) TITLE,JN                                     LAM 163
145 FORMAT (1H1,43X,30HLAMINAR BOUNDARY LAYER SUMMARY//26X,20A4,9X,13H LAM 164
   1UPPER SURFACE/112X,16HITERATION NUMBER,13,//)                   LAM 165
150 FORMAT (1H1,43X,30HLAMINAR BOUNDARY LAYER SUMMARY//26X,20A4,9X,13H LAM 166
   1LOWER SURFACE/112X,16HITERATION NUMBER,13,//)                   LAM 167
155 SCRIT=SCRIT/CREF                                                LAM 168
    STRN=STRAM/CREF                                                 LAM 169
    THETI=THETA1/CREF                                               LAM 170
    ANGL=ALFA*57.2957795                                            LAM 171
    XST=XSTAG/CREF                                                  LAM 172
    ZST=ZSTAG/CREF                                                  LAM 173
    WRITE (JWRITE,160) TO,PO,ANEIF,C,RN,PR,KF,SEP,A0,VGTRUE,ANGL,XST,Z LAM 174
   1ST                                                              LAM 175
160 FORMAT (23H0STAGNATION TEMPERATURE,3X,1H=,F8.2,6X,19HDEGREES RANK1 LAM 176
   1NE,7X,19HSTAGNATION PRESSURE,8X,1H=,F9.3,5X,8HLB0/SC FT/23H FREESTR LAM 177
   1EAM MACH NUMBER,3X,1H=,3X,F8.5,25X,13HAIRFOIL CHORD,14X,1H=,3X,F8.  LAM 178
   15,3H,2HF7/16H REYNOLDS NUMBER,10X,1H=,2X,F9.5,3X,7HMILLION,15X,14H  LAM 179
   1PRANDTL NUMBER,13X,1H=,3X,F8.5/23H HEAT TRANSFER FACTOR K,3X,1H=,3  LAM 180
1ST                                                                 LAM 181
    1X,F8.5,25X,26HSEPARATION CORRELATION NO.,1X,1H=,3X,F9.6/19H SPEED  LAM 182
   10F SOUND,11X,1H=,F9.3,5X,11HFEET/SECOND,11X,19HKINEMATIC VISCOSITY  LAM 183
   1,8X,1H=,4X,F9.7,1X,9H5Q FT/SEC/24H ANGLE OF ATTACK (ALPHA),2X,1H=,  LAM 184
   12X,F9.5,3X,7HDEGREES/18H STAGNATION AT X/C,6X,1H=,3X,F10.7,23X,17H  LAM 185
   1STAGNATION AT Z/C,10X,1H=,3X,F10.7)                             LAM 186
    WRITE (JWRITE,165)                                              LAM 187
165 FORMAT (1H0,8X,3H1/C,10X,3HS/C,11X,1H8,8X,9HDW/DIS/C),7X,1H=,10X,7 LAM 188
   1HTHETA/C,6X,6HDELS/C,7X,8HDELTAS/C,7X,2HCF,10X,2HCP//)          LAM 189
    DO 170 K=2,NPT                                                  LAM 190
    XOC=X(K)/CREF                                                   LAM 191
    SOC=SUMS(K)/CREF                                                LAM 192
    DWDS=CREF*SLO(K)                                                LAM 193
    TOC=AMTK(K)/CREF                                                LAM 194
    DOC=DISP(K)/CREF                                                LAM 195
    NHH=ITEMP(K)                                                    LAM 196
    SSG(K)=CF(NHH)                                                  LAM 197
170 WRITE (JWRITE,175) XOC,SOC,AME(K),DWDS,MI(K),TOC,DOC,SSG(K)K,CP(NHH) LAM 198
175 FORMAT (1H-,9F13.6,13X,F13.6,F12.5)                            LAM 199
    IF (ITRAN.EQ.1) RETURN                                          LAM 200
    WRITE (JWRITE,180) SCRIT,RECRIT,STRN,RTRAM,THET1                LAM 201
180 FORMAT (1H0,9HSCRIT/C =,F13.6,3X,8HRECRIT =,F10.2,5X,9HSTRAM/C =,F LAM 202
   113.8,3X,7HRTRAM =,F10.2,2X,10HTHETA1/C =,F13.8)                 LAM 203
    DO 250 I=1,6                                                    LAM 204
    IF (IFLAG(I)) 250,250,185                                       LAM 205
185 GO TO (190,200,210,220,230,240),I                              LAM 206
190 XTR=XTRAM(JS)/CREF                                             LAM 207
    WRITE (JWRITE,195) XTR,STRN                                     LAM 208
195 FORMAT (47H0INSTABILITY TRANSITION HAS OCCURRED AT   X/C =,F11.6,8 LAM 209
   1X,5H5/C =,F11.6)                                                LAM 210
    GO TO 250                                                       LAM 211
200 XTR=XTRAM(JS)/CREF                                             LAM 212
    WRITE (JWRITE,205) XTR,STRN                                     LAM 213
205 FORMAT (41H0FIXED TRANSITION HAS OCCURRED AT   X/C =,F11.6,8X,5H5/ LAM 214
   1C =,F11.6)                                                      LAM 215
    GO TO 250                                                       LAM 216
210 XTR=XTRAM(JS)/CREF                                             LAM 217
    WRITE (JWRITE,215) XTR,STRN                                     LAM 218
215 FORMAT (44H0LOCURLE LAMINAR SEPARATION HAS OCCURRED AT   X/C =,F11.6 LAM 219
   1,8X,5H5/C =,F11.6)                                              LAM 220
    GO TO 250                                                       LAM 221
220 XTR=XTRAM(JS)/CREF                                             LAM 222
    WRITE (JWRITE,225) XTR,STRN                                     LAM 223
225 FORMAT (44H0CM LAMINAR SEPARATION HAS OCCURRED AT   X/C =,F11.6,8X LAM 224
   1,5H5/C =,F11.6)                                                 LAM 225
    GO TO 250                                                       LAM 226
230 WRITE (JWRITE,235) RTRAM                                       LAM 227
235 FORMAT (19H0SHORT BUBBLE,10X,25HTRANSITION REYNOLDS NO. =,F13.6)  LAM 228
    GO TO 25C                                                       LAM 229
240 WRITE (JWRITE,245) RTRAM                                       LAM 230
245 FORMAT (12H0LONG BUBBLE,10X,25HTRANSITION REYNOLDS NO. =,F13.6)   LAM 231
250 CONTINUE                                                        LAM 232
    RETURN                                                          LAM 233
    IF (LTRAN(JS).GT.0)NINIT=1.45                                   LAM 234
    END                                                             LAM 235

      SUBROUTINE BLTRAN(ITRAN,IFLAG,JS,NP,SEND)                     BLT 1
C***  ROUTINE TO PREDICT LAMINAR TO TURBULENT BOUNDARY LAYER TRANSITION BLT 2
      REAL KF                                                       BLT 3
```

```
      COMMON RCT(100),ZCT(100),THETA(65),SI(65),VDV(65),CP(65),CV(65),CFI
     1(65),DLTAS(65),GVDT(65),GVDI(65),DLTASP(65),GVBDP(65),TITLE(20),KT
     IRAN(2),ZTRAN(2),TTRAN(2),ALFA,FSMCH,PO,TO,RM,PK,KF,CREF,C,XSTAG,ZS
     ITAG,TSTAG,SSTAG,BETA,LTRAN(2),IWRITE,MAXK,IALPHA
      COMMON /VISCOS/SUMS(43),AME(43),SLO(43),UET(43),DDMOS(43),X(43),ZTE
     IMP(43),TIMFX,ITEMP(43),ISURF
      COMMON /LAMTR/AMTK(43),REMI(43),UET1(43),UET2(43),CNI(43),UET(43),UET
     1(43),KBAR(43),RTHTR(43),REMOM(43),SG(43),SW,AD,VO,SEP,AMU,SQ,F,ML,I
     1(43),K,KOM
      COMMON /CMFLTR/DLTA(2),THTA(2)
      REAL*8 A
      DIMENSION XKAY(43),MBL(43),RTCR(43),MUF(43),VM(43),LBL(43),THETAF(
     1(43),A(4),IFLAG(4)
      COMMON /TURBTR/SCRIT,RECRIT,STRAN,RTRAN,MINIT,AMETR,SLOTR,M
   14
      EQUIVALENCE (CMSTAG,CM(11)),(LBLST,LBL(1))
      REAL LBL,MBL,MUF,(LSEP,LBLK,KBAR,LBLST
      PI=3.1415927
      ITRAN=0
      ZTSIGM=ZTEMP(K)
      ZRSIGM=ZTRAN(JS)
      IF (ZTSIGM.EQ.0.0)ZTSIGM=1.0E-08
      IF (ZRSIGM.EQ.0.0)ZRSIGM=1.0E-08
      ZTSIGM=ZTEMP(K)/ABS(ZTSIGM)
      ZRSIGM=ZTRAN(JS)/ABS(ZRSIGM)
      J=1
      CALL POINT(SUMS,X,SEND,XEMO,1,NP,A)
      AMEIF=FSMCH
    5 AMEIF=-.01
   10 IF (LTRAN(JS).GT.0.AND.K.EQ.2) GO TO 15
      GO TO 25
   15 THETAF(1)=TSTAG
      DO 20 LL=2,NP
      LLK=ITEMP(LLJ)
      THETAF(LLJ)=THETAF(LLK)-THETAF(1)
      IF (JS.EQ.2)THETAF(LLJ)=-THETAF(ILLJ)
   20 CONTINUE
      THETAF(1)=0.0
      TMTFIK=ABS(TTRAN(JS)-TSTAG)
   25 LBLST=.0855
      COM1=EXP(43.37450+.0767+218.28+.0767++2-1934.6+.0767++3-23980.+.07
     1670++45.46963)
      VM(K)=VO+1.+.2+AME(K)++2)ME(K)++2)++1.5+F
      LBL(K)=(AMTK(K)++2QUET1(K))/VM1(K)++1.+.2+AME(K)++2)++2
      MBL(K)=(UET(K)+UET2(K))/UET1(K)++2+LBL(K)++2
      LBL(K)=LBL(K)
      IF (LBLK-.13) 30,35,35
   30 LBLK=-.13
   35 F1=0.048609390.705064+9LBLK+1.97856+9LBLK++2-3.309716+9LBLK++3+27.0
     1173+9LBLK++4
      G1=0.150354490.931702349LBLK+3.23025949LBLK++2+1.002594LBLK++3-178.2
     1490LBLK+++1670.5520LBLK++5
      MUF(K)=F1/G1
      XKAY(K)=REMI(K)++2++VO/AD/SQ+SLO(K)
      IF (XKAY(K)-.0767) 45,45,40
   40 XKAY(K)=-0767
      GO TO 55
   45 IF (XKAY(K)+.1567) 50,50,55
   50 XKAY(K)=-.1567
   55 RTCR(K)=EXP(5.46963+43.37450+XKAY(K)+218.28+XKAY(K)++2-1934.6+XKAY
     1(K)++3-23980.+XKAY(K)++4)
      IF (K-2) 75,65,75
   60 ITRAN=-1
      RETURN
   65 IF (RTCR(K)-REMI(K)) 70,70,170
   70 SCRIT=COM1+SUMS(K)/(REMI(K)-RTCR(2)+COM1)
      RECRIT=REMI(K)+SCRIT/SUMS(K)
      GO TO 125
   75 IF (KOM-1) 80,130,130
   80 KCH=1
      IF (RTCR(K)-REMI(K)) 85,85,95
   85 IF (KOM-1) 90,130,130
   90 KOM=1
      GO TO 100
   95 KCH=0
  100 SCRIT=SUMS(J)+(RTCR(J)-REMI(J))+(SUMS(K)-SUMS(J))/(REMI(K)-REMI(J))
     1-RTCR(J)+RTCR(J))
      IF (SCRIT) 105,110,110
  105 SCRIT=0.
  110 IF (SCRIT-SUMS(NP)) 120,120,115
  115 SCRIT=SUMS(NP)+0.001+C
  120 RECRIT=REMI(J)+(SCRIT-SUMS(J))+(REMI(K)-REMI(J))/(SUMS(K)-SUMS(J))
  125 CALL POINT(SUMS,AME,SCRIT,AMECR,1,NP,A)
      SLOCR=A(2)+2.+A(3)+SCRIT
      EKKA=RECRIT++2/C+(1.+.1750+AMECR++2+.6+SW+.42171+SW+AMECR++2+.008
     132+AMECR++4-.060343+SW+AMECR++4+.18253+SW+SW+.073+SW+SW+AMECR++2+
     1.0073+SW+SW+AMECR++4)+VO/AD/AMECR++2+SLOCR+C
      AMEIF=(XKAY(K)+EXKA)+(SUMS(K)-SCRIT)/2.
      ADD=(XKAY(K)+EXKA)+(SUMS(K)-SCRIT)/2.
      GO TO 135
  130 ADD=ADD+(XKAY(K)+XKAY(J))+(SUMS(K)-SUMS(J))/2.
  135 KBAR(K)=ADD/(SUMS(K)-SCRIT)
      IF (KBAR(K)+.05) 140,145,145
  140 KBAR(K)=-.05
  145 BGM=825.45+28183.5+KBAR(K)+72198.+KBAR(K)++2+63173800.+KBAR(K)++3
      RTHTR(K)=BGM+RECRIT
      IF (KCH) 150,150,155
  150 KCH=1
      GO TO 225
  155 IF (K-2) 215,160,215
  160 IF (RTHTR(K)-REMI(K)) 165,165,170
  165 STRAN=SCRIT+(RTHTR(K)-RECRIT)+SUMS(K)/(REMI(K)-RTCR(K)+COM1)
      IFLAG(1)=1
      GO TO 210
  170 IF (SUMS(K)-SEND) 175,180,180
  175 IF (LTRAN(JS).LE.0) GO TO 190
      IF (ZTSIGM.NE.ZRSIGM) GO TO 60
      IF (THETAF(K)-TMTFIK) 60,185,185
  180 IFLAG(2)=1
      STRAN=SUMS(K)+(XTRAN(JS)-XSTAG)/(X(K)-XSTAG)
      GO TO 210
  185 CALL POINT(THETAF,SUMS,TMTFIK,STRAN,1,NP,A)
      GO TO 210
  190 IF (MUF(K)-MBL(K)) 195,200,200
  195 LSEP=MUF(K-1)/(MBL(K)-LBLST)-(MUF(K-1)-MUF(K))/(LBL(K)-LBL
     1ST))
      STRAN=SUMS(K)+(LSEP-LBLST)/(LBL(K)-LBLST)
      SCSEP=STRAN
      GO TO 210
```

285

```
200  IF (CN(2)-SEP)  60,205,205
205  IFLAG(4)=1
     STRAM=SUMSK(K)+(SEP-CN(TAG)/(CN(K)-CN(TAG)
     LSEP=LBL(K-1)+(LBL(K)-LBL(K-1))+STRAM/SUMS(K)
     SCSEP=STRAM
210  RTRAM=STRAM+REN(K)/SUMS(K)
     GO TO 350
215  IF (RTHTR(K)-REN(K))  220,220,225
220  STRAM=SUMS(J)+(RTHTR(J)-REN(J))+SUMS(K)-SUMS(J))/(REN(K)-REN(J))
     1=1-RTHTR(K)+RTNTR(J))
     IFLAG(1)=1
     GO TO 255
225  IF (SUMS(K)-SEMO)  230,235,235
230  IF (LTRAM(JS)-LE.0) GO TO 245
     IF (ZFSIGN.NE.ZRSIGN) GO TO 245
     IF (THETAF(K)-THTF(X) 245,240,240
235  IFLAG(2)=1
240  IFLAG(2)=1
     GO TO 300
245  IF (CN(K)-SEP)  410,250,250
250  IFLAG(4)=1
255  IF (SUMS(K)-SEMO)  265,260,260
260  XTRAM(JS)-XEMO
     GO TO 270
265  IF (LTRAM(JS)-LE.0) GO TO 305
270  ARG=2.+(XTRAM(JS)-XCT(33))/(XCT(1))-XCT(33))-1.0
     IF (ABS(ARG)-1.0)  280,280,275
275  ARG=SIGN(1.0,ARG)
280  THT=ARCOS(ARG)
     THT1=2.+PI-THT
     THTR2=THT
     CALL POINT(THETA+ZCT,THTR1,ZTR1,1,65,A)
     CALL POINT(THETA+ZCT,THTR2,ZTR2,1,65,A)
     D1=ABS(ZTRAM(JS)-ZTR1)
     D2=ABS(ZTRAM(JS)-ZTR1)
     IF (D2-D1)  285,290,290
285  THTR=THTR2
     GO TO 295
290  THTR=THTR1
295  CALL POINT(THETA+S,THTR,STR,1,65,A)
     STRAM=STR-SSTAG
     STRAM=ABS(STRAM)
     GO TO 300
300  CALL POINT(THETAF,SUMS,THTF(X,STRAM,1,MP,A)
305  IF (CN(K)-SEP)  345,345,310
310  IF (IFLAG(1)+IFLAG(2)-1)  320,315,315
315  STRAM=STRAM
320  CALL POINT(CN,SUMS,SEP,STRAM,K-2,K,A)
     IF (SUMS(K)-STRAM)  330,335,325
325  IF (SUMS(K)-STRAM)  335,335,330
330  STRAM=SUMS(K-1)+(SUMS(K)-SUMS(K-1))/(CN(K)-CN(K-1))+(SEP-CN(K-1))
335  STRAM=SUMS(K-1)+(SUMS(K)-SUMS(K-1))+(STRAM-SUMS(K-1))/(SUMS(K)-SUMS(K-
     1))
     SCSEP=STRAM
     IFLAG(4)=1
340  STRAM=STRAM
345  RTRAM=REN(J)+(STRAM-SUMS(J))+(REN((J)-REN((J))/(SUMS(K)-SUMS(J))
```

```
350  CALL POINT(SUMS,AME,STRAM,AMETA,1,MP,A)
     IF (RMF(K)-REN(K))  360,355,355
355  IF (CN(K)-SEP)  375,360,360
360  PG=-0.02+REMDR(K)-1.0
     IF (PG-SLOK(K)C)  365,370,370
365  IFLAG(5)=1
     GO TO 375
370  IFLAG(4)=1
375  IF (AMETA)  390,385,385
380  AMETR=1.E-9
385  SLOTR=A(12)+2.+A(13)+STRAM
     CALL POINT(SUMS,ML,STRAM,MINI,1,K,A)
     IF (RTRAM)  390,390,395
390  RTRAM=1.01
395  DEFORM=-.793091E+-.06590469+ALOG(RTRAM)-.0012717(6)+(ALOG(RTRAM))+2-
     1.2
     MINIT=MINI-DEFORM
     CALL POINT(SUMS,AMTK,STRAM,THETA1,2,K,A)
     CALL POINT(SUMS,SG,STRAM,CFTRA,2,K,A)
     CFTRA=CFTRA+(1.+0.2+AME(K)++2)/(1.+SW)
     IF (XTRAM(JS)-XEMO+0.1E-B+C)  400,400,405
400  ITRAM=2
     GO TO 410
405  ITRAM=1
     KDK=-C-2
     IF (K.EQ.2)KDK=1
     CALL POINT(K,SUMS,XEMO,STRAM,KDK,K,A)
     CALL POINT(SUMS,AMTK,STRAM,THETA1,KDK,K,A)
     TMTA(JS)=THETA1
     DLTA(JS)=0.+THETA1
410  RETURN
     END
```

```
C***  SUBROUTINE TURB2(MP,MI,JS,JM,SEMO)
C     ROUTINE FOR CALCULATION OF TURBULENT BOUNDARY LAYER
      COMMON XCT(100),ZCT(100),THETA(65),S(65),VOV(65),CP(65),GV(65),CFI
     1(65),DLTAS(65),GVT(65),DLTASP(65),GVBD(65),GVBDP(65),TITLE(20),XT
     IRAM(2),ZTRAM(2),TTRAM(2),ALFA,FSPCH,PO,TO,RN,PR,KF,CREF,C,KSTAG,ZS
     ITAG,TSTAG,SSTAG,BETA,LTRAM(2),IWRITE,MAKK,IALPHA
      REAL KF,RMOLD,RBAR
      COMMON /VISCOS/SUMS(43),AME(43),SLO(43),UE(43),CDMOS(43),X(43),ZTE
     IMP(43),TIMFK,I(TEMP(43),ISURF
      COMMON /CMFLTR/DLTA(2),TMTA(2)
      COMMON /TURBTR/SCRIT,RECRIT(STRAM,RTRAM,RTRAM,THETA1,MINIT,AMETR,SLOTR,M
     14
      COMMON /LAMTR/MI(43),AMTK(43),REN(43),UET(43),REN1(43),UET2(43),CN(43),UET
     1(43),KBAR(43),RTHTR(43),REMOR(43),SG(43),SW=AO,VO,SEP,AMU,SQ,F,ML1
     1(43),K,KON
      COMMON /INOUT/JREAD,JWRITE
      DIMENSION HMEAM(43),HVMEAM(43),DEL2(43),DELL(43),CFLOG(43),DSTR(43
     1),TTA(43)
      EQUIVALENCE (JMU2,XMU2)
      EQUIVALENCE (STRAM,XINI),(AMETR,XME),(TTA(1),AMTK(1)),(HMEAM(1),HI
     1))
      COMMON /DRAG/TNTUP,MUP,TNTLO,MLO,COSTI
      IF (ILMITE.EQ.1.AND.JM.EQ.0) GO TO 35
      IF (ILMITE.GT.1) GO TO 35
      IF ((J.GT.0).AND.(JM.LT.MAKK-1)) GO TO 35
```

```
TB2  26   IF (ISURF.EQ.1) WRITE (JWRITE,5) TITLE,JN
TB2  27   IF (ISURF.EQ.1) GO TO 15
TB2  28   WRITE (JWRITE,10) TITLE,JN
TB2  29  5 FORMAT (1H1,24X,55HTURBULENT BOUNDARY LAYER SUMMARY FOR EQUIVALENT
TB2  30   1 AIRFOIL,5X,16HIGORADIA METHOD//25X,20A4,5X,13HUPPER SURFACE/10TX
TB2  31   1,16HITERATION NUMBER,I3,//)
TB2  32 10 FORMAT (1H1,24X,55HTURBULENT BOUNDARY LAYER SUMMARY FOR EQUIVALENT
TB2  33   1 AIRFOIL,5X,16HIGORADIA METHOD//25X,20A4,5X,13HLOWER SURFACE/10TX
TB2  34   1,16HITERATION NUMBER,I3,//)
TB2  35 15 SOC=STRAN/CREF
TB2  36   TH=THETA1/CREF
TB2  37   ANGL=ALFA*57.2957795
TB2  38   WRITE (JWRITE,20) TO,PO,FSMCH,C,RN,PR,RF,SOC,TH,HINIT,ANGL
TB2  39 20 FORMAT (23HOSTAGNATION TEMPERATURE,9X,1H=,F8.2,4X,15HDEGREES RANKI
TB2  40   1NE,7X,19HSTAGNATION PRESSURE,8X,1H=,F9.3,3X,8MLB/SC FT/23H FREESTR
TB2  41   1EAM MACH NUMBER,9X,1H=,3X,F8.5,23X,13HAIRFOIL CHORD,14X,1H=,3X,F8.
TB2  42   15.3H FT/16H REYNOLDS NUMBER,16X,1H=,2X,F9.5,8H MILLION,15X,14HPRAN
TB2  43   1DTL NUMBER,13X,1H=,3X,F8.5/23H HEAT TRANSFER FACTOR K,9X,1H=,3X,F8
TB2  44   1.5,23X,21HTRANSITION POINT, S/C,6X,1H=,3X,F8.5/31H (INITIAL MOMENT
TB2  45   1UM THICKNESS)1/C,1X,1H=,3X,F8.5/23X,26HINIT. INCCMPR. FORM FACTOR,1
TB2  46   1X,1H=,3X,F8.5/24H ANGLE OF ATTACK (ALPHA),8X,1H=,2X,F9.5,1X,7HDEGR
TB2  47   1EES)
TB2  48   WRITE (JWRITE,25)
TB2  49 25 FORMAT (1HO,4X,3HX/C,13X,3HS/C,11X,1HH,8X,8HDM/(S/C),8X,1HH,10X,7H
TB2  50   1THETA/C,6X,6HDELS/C,6X,8HDELTAS/C,8X,2HCF,10X,2HCF/)
TB2  51   XTR=XTRAN(JS)/CREF
TB2  52   STR=STRAN/CREF
TB2  53   DMDX=SLOTR*CREF
TB2  54   DPRN=THETA1*HINIT/CREF
TB2  55   WRITE (JWRITE,30) XTR,STR,AMETR,DMDX,HINIT,TH,DPRN
TB2  56 30 FORMAT (1H ,F10.6,2X,6F13.6,13X,F13.6)
TB2  57 35 C1=0.56
TB2  58   C2=1.667
TB2  59   C3=1.45
TB2  60   C4=0.246
TB2  61   C5=0.678
TB2  62   C6=0.268
TB2  63   N1=N4-1
TB2  64   UE(N1)=UE(N1)+((STRAN-SUMS(N1+1)-SUMS(N1+1))*(UE(N1+1)-UEIN
TB2  65   1))
TB2  66   THOLD=TTA(N1)
TB2  67   XHOLD=SUMS(N1)
TB2  68   UHOLD=UE(N1)
TB2  69   RHOLD=AME(N1)
TB2  70   TTA(N1)=THETA1
TB2  71   IF (XINI-(SUMS(N1+1)-.00001)) 65,65,40
TB2  72   XIN1=SUMS(N1+1)-.CCCG1
TB2  73 40 SUMS(N1)=XIN1
TB2  74 45 UE(N1)=UEIN1
TB2  75   AME(N1)=XME
TB2  76   HMEAN(N1)=HINIT
TB2  77   HVMEAN(N1)=1.26*HMEAN(N1)/(HMEAN(N1)-.379)
TB2  78   XNUX1=(1.+0.2*XME**2)*1.75*V0
TB2  79   E1=3.+2.*C2
TB2  80   E2=1.+C2
TB2  81   E3=3.+3.*C2
TB2  82   TI=0.02*C1/C3*E2
TB2  83   IBEG=N1+1
TB2  84   DO 90 I=IBEG,MP
TB2  85   JSAVE=1
```

```
TB2  86   HMEAN(I)=0.0
TB2  87   HVMEAN(I)=0.0
TB2  88   XNUX2=(1.+0.2*AME(I)**2)*1.75*V0
TB2  89   IF (I.EQ.MP) GO TO 50
TB2  90   DUEDX=(UE(I)-UE(I-1))/(SUMS(I-1)-SUMS(I-1))*(UE(I+1)-UE(I))/(SUMS(I
TB2  91   1+1)-SUMS(I)))*0.5
TB2  92   UETR=(UE(I-1)/UE(I))*E3
TB2  93   UEINT=0.5*(SUMS(I-1)-SUMS(I-1))*(UE(I)*UE(I-1)**E1)
TB2  94   TTA(I-1)=TTA(I-1)**E2*UETR*(XNUX2/XNUX1)*C2*T1*XNUX2**C2/(UE(I)**E
TB2  95   131)*UE(N1)**(./E2)
TB2  96   13)*UE(N1)**(./E2)
TB2  97   TERRA=(0.02*C1/(UE(I)*TTA(I))/XNUX2)*C2*G1.1
TB2  98   T2=TTA(I)/UE(I)*DUEDX
TB2  99   P=HMEAN(I-1)
TB2 100   HV=HVMEAN(I-1)
TB2 101   DO 75 J=1,N1
TB2 102   TERMB=-HVC4*10-0*(-C5*HM1)*(UE(I)*TTA(I))/XMU2)**(-C6)
TB2 103   TERMB=TERMB/2.
TB2 104   TERMC=(HM-1.1)*HV*T2
TB2 105   DHVDX=(TERMA*TERMB*TERMC)/TTA(I-1)
TB2 106   HV=HVMEAN(I-1)*DHVDX*(SUMS(I-1)-SUMS(I-1))
TB2 107   IF (HVITR-1.65) 60,55,55
TB2 108   55 HVITR=1.85
TB2 109   60 IF (HVITR-1.55) 65,65,70
TB2 110   65 HVITR=1.55
TB2 111   70 1-ITR=0.379*HVITR/(HVITR-1.269)
TB2 112   HVMEAN(I)=HVMEAN(I)*HVITR
TB2 113   HMEAN(I)=HMEAN(I)*HITR
TB2 114   HV=HVITR
TB2 115 75 H=ITR
TB2 116   HVMEAN(I)=HVMEAN(I)/NI
TB2 117   HMEAN(I)=TTA(I)/NI
TB2 118   DSTR(I)=TTA(I)*HMEAN(I)
TB2 119   DEL(I)=(TTA(I)*HMEAN(I)*(HMEAN(I)*1.01)/(HMEAN(I)-1.)
TB2 120   DEL2(I)=-.000497*13.89*TTA(I)*191.64*TTA(I)**2
TB2 121   CFLOC(I)=C4*10.0*(-C5*HMEAN(I))*UE(I)*TTA(I)/XMU2))**(-C6)
TB2 122   XNUX1=XNUX2
TB2 123   XOC=X(I)/CREF
TB2 124   SOC=SUMS(I)/CREF
TB2 125   DMDX=SLO(I)/CREF
TB2 126   DEL=DSTR(I)/CREF
TB2 127   DEL=DEL(I)/CREF
TB2 128   DELT=DEL(I)/CREF
TB2 129   IF (IWRITE.EQ.1.AND.JN.EQ.0) GO TO 85
TB2 130   IF (IWRITE.GT.1) GO TO 85
TB2 131   IF ((I.JN.GT.01.AND.(JN.LT.MAXK-1))) GO TO 85
TB2 132   NHH=ITEMP(I)
TB2 133   CFLOC1=CFLOC(I)*(1.-CP(NHH))
TB2 134   WRITE (JWRITE,80) XOC,SOC,AME(I),DMDX,HMEAN(I),TH,CEL,DEL,CFLOC1,
TB2 135   1CP(NHH)
TB2 136 80 FORMAT (1H ,F10.6,2X,9(1X,F12.6))
TB2 137 85 IF (SEND-SUMS(I)) 95,95,90
TB2 138 90 CONTINUE
TB2 139 95 DO 100 I=IBEG,JSAVE
TB2 140   IA=ITEMP(I)
TB2 141   OLTAS(IA)=DSTR(I)
TB2 142 100 CF(I(IA)=CFLOC(I))*(1.-CP(IAI)
TB2 143   OLTA(JS)=DEL(JSAVE)
TB2 144   THTA(JS)=TTA(JSAVE)
TB2 145 C  RETURN N1 ELEMENTS TO THEIR ARRAYS
```

```
      TTA(N1)=THOLD                                                  TB2 146
      SUMS(N1)=ZHOLD                                                 TB2 147
      UE(N1)=RHOLD                                                   TB2 148
      AME(N1)=MHOLD                                                  TB2 149
      IF (ISURF.NE.1) GO TO 105                                      TB2 150
      THTUP=TTA(JSAVE)                                               TB2 151
      HUP=HMEAN(JSAVE)                                               TB2 152
      GO TO 110                                                      TB2 153
  105 THTLD=TTA(JSAVE)                                               TB2 154
      HLD=HMEAN(JSAVE)                                               TB2 155
  110 RETURN                                                         TB2 156
      END                                                            TB2 157

      SUBROUTINE LOAD(NP,CMP,CMF,CAP,CAF,CMP,CMF)                    LOD 1
C***  ROUTINE TO INTEGRATE VISCOUS AND PRESSURE FORCES               LOD 2
      REAL KF                                                        LOD 3
      COMMON XCT(100),ZCT(100),THETA(100),S(65),VOV(65),CP(65),CV(65),CF1  LOD 4
     1(65),DLTAS(65),GVBT(65),GVBD(65),TITLE(20),XT                  LOD 5
     2RAN(2),ZTRAN(2),TRAN(2),ALFA,FSMCH,PO,TO,RN,PR,KF,CREF,C,XSTAG,ZS  LOD 6
     3TAG,ISTAG,SSTAG,BETA,LTRAN(2),IWRITE,MAXK,IALPHA              LOD 7
      CMF=0.0                                                        LOD 8
      CAF=0.0                                                        LOD 9
      CMP=0.0                                                        LOD 10
      CAP=0.0                                                        LOD 11
      CMP=0.0                                                        LOD 12
      DO 5 I=2,65                                                    LOD 13
      ISTAG=I                                                        LOD 14
      IF (GV(I).GE.0.) GO TO 10                                      LOD 15
    5 CONTINUE                                                       LOD 16
   10 DO 15 I=2,ISTAG                                                LOD 17
   15 CF(I-1)=CF(I-1)                                                LOD 18
      DO 20 I=2,65                                                   LOD 19
      X1=XCT(I-1)                                                    LOD 20
      X2=XCT(I)                                                      LOD 21
      Z1=ZCT(I-1)                                                    LOD 22
      Z2=ZCT(I)                                                      LOD 23
      CAP=CAP+(CP(I)+CP(I-1))*(Z2-Z1)/2.*(Z2-Z1)                     LOD 24
      CMP=CMP-(CP(I)+CP(I-1))*(X2-X1)/2.*(X2-X1)                     LOD 25
      CMP=CMP-(CP(I)+CP(I-1))*(X2-X1)*((X2-X1)+(Z2-Z1))*(Z2-Z1))/4.0 LOD 26
      X3=CF(I)+CF(I-1)                                               LOD 27
      CMF=CMF+X3*(Z2-Z1)/2.0                                         LOD 28
      CAF=CAF+X3*(X2-X1)/2.                                          LOD 29
   20 CMF=CMF+X3*((X2-X1)*(Z2-Z1))-((Z2-Z1)*(X2+X1))/4.0             LOD 30
      CAP=CAP/CREF                                                   LOD 31
      CMP=CMP/CREF                                                   LOD 32
      CMP=CMP/CREF**2                                                LOD 33
      CAF=CAF/CREF                                                   LOD 34
      CMF=CMF/CREF                                                   LOD 35
      CMF=CMF/CREF**2                                                LOD 36
      RETURN                                                         LOD 37
      END                                                            LOD 38

      SUBROUTINE LSQ(FX,X,N,NP,C)                                    LSQ 1
C***  LEAST SQUARES CURVE FIT ROUTINE. FX IS THE DEPENDENT VARIABLE, LSQ 2
C***  X IS THE INDEPENDENT VARIABLE. NP IS THE NUMBER OF OBSERVATIONS, LSQ 3
C***  AND N IS THE ORDER OF THE FITTED POLYNOMIAL.                   LSQ 4
C***  FX(I)=C(1)+C(2)*X+C(3)*X**2+...+C(N+1)*X(INP)*FX(INP)         LSQ 5
      DIMENSION A(11,12),C(4),T1,T2                                  LSQ 6
      REAL*8 A,T1,T2                                                 LSQ 7
      N1=N+1                                                         LSQ 8
      N2=N+2                                                         LSQ 9
C***  SET UP MINIMIZING MATRIX                                       LSQ 10
      DO 5 I=1,N1                                                    LSQ 11
      DO 5 J=1,N2                                                    LSQ 12
    5 A(I,J)=0.                                                      LSQ 13
      DO 15 K=1,NP                                                   LSQ 14
      T1=1.                                                          LSQ 15
      T2=T1                                                          LSQ 16
      DO 15 J=1,N1                                                   LSQ 17
      DO 10 I=1,N1                                                   LSQ 18
      A(J,I)=A(J,I)+T2                                               LSQ 19
   10 T2=T2*X(K)                                                     LSQ 20
   15 T1=T1*X(K)                                                     LSQ 21
C***  SOLVE FOR COEFFICIENT VECTOR                                   LSQ 22
      DO 20 K=1,N                                                    LSQ 23
      DO 20 J=K,N                                                    LSQ 24
      T1=A(J+1,K)/A(K,K)                                             LSQ 25
      DO 20 I=K,N2                                                   LSQ 26
   20 A(J+1,I)=A(J+1,I)-A(K,I)*T1                                    LSQ 27
      C(N1)=-A(N1,N2)/A(N1,N1)                                       LSQ 28
      DO 25 I=2,N1                                                   LSQ 29
      K=N2-I                                                         LSQ 30
      L=K-1                                                          LSQ 31
      C(K)=-A(K,N2)/A(K,K)                                           LSQ 32
      DO 25 J=L,N1                                                   LSQ 33
   25 C(K)=C(K)-C(J)*A(K,J)/A(K,K)                                   LSQ 34
      RETURN                                                         LSQ 35
      END                                                            LSQ 36
                                                                     LSQ 37

      SUBROUTINE TRANS(XTRANS,ZTRANS,XMS,ZMS,X,Z,DELTA,XP,ZP,NPT)   TRN 1
C***  ROUTINE WHICH TRANSLATES AND ROTATES (THROUGH ANGLE DELTA) A SET  TRN 2
C***  OF NPT POINTS (X'S AND Z'S) TO A NEW SET OF POINTS (XTRANS'S AND   TRN 3
C***  ZTRANS'S) SUCH THAT THE POINT (XP,ZP) BECOMES THE POINT (XMS,ZMS). TRN 4
      DIMENSION XTRANS(NPT),ZTRANS(NPT),X(NPT),Z(NPT)                TRN 5
      COSD=COS(DELTA)                                                TRN 6
      SIND=SIN(DELTA)                                                TRN 7
      DO 5 I=1,NPT                                                   TRN 8
      XSAVE=X(I)                                                     TRN 9
      XTRANS(I)=XMS+(X(I)-XP)*COSD+(Z(I)-ZP)*SINC                    TRN 10
    5 ZTRANS(I)=ZMS+(Z(I)-ZP)*COSD-(XSAVE-XP)*SIND                   TRN 11
      RETURN                                                         TRN 12
      END                                                            TRN 13

      SUBROUTINE PRCOT(A,X)                                          PRO 1
C***  ROUTINE FOR SOLVING A CUBIC POLYNOMIAL                         PRO 2
      DIMENSION A(4),Z(6),X(6)                                       PRO 3
      DIMENSION U(6),V(6)                                            PRO 4
      REAL M                                                         PRO 5
      PI=3.1415927                                                   PRO 6
      W=(3.*A(4)*A(2)-A(3)**2)/(9.*A(4)**2)                          PRO 7
```

```
G=(2.*A(3)**3-9.*A(4)*A(3)*A(2)+27.*A(4)**2*A(1))/(27.*A(4)**3)      PRO 8
TEST=G**2+4.*H**3                                                     PRO 9
IF (TEST) 5,15,15                                                     PRO 10
5 IM=SQRT(ABS(TEST))/2.                                               PRO 11
RE=-G/2.                                                              PRO 12
R=SQRT(RE**2+IM**2)                                                   PRO 13
THETA=ATAN(IM/RE)                                                     PRO 14
IF (THETA) 10,30,30                                                   PRO 15
10 THETA=THETA+PI                                                     PRO 16
GO TO 30                                                              PRO 17
15 R=-(-G+SQRT(TEST))/2.                                              PRO 18
IF (R) 20,25,25                                                       PRO 19
20 THETA=PI                                                           PRO 20
R=-R                                                                  PRO 21
GO TO 30                                                              PRO 22
25 THETA=0.                                                           PRO 23
30 R13=R**(1./3.)                                                     PRO 24
T=THETA/3.-(K-1)*PI/3.                                                PRO 25
CT=COS(T)                                                             PRO 26
ST=SIN(T)                                                             PRO 27
U(K)=R13*CT                                                           PRO 28
U(K+1)=R13*ST                                                         PRO 29
V1=-H/(U(K)**2+U(K+1)**2)                                             PRO 30
V(K)=V1*U(K)                                                          PRO 31
V(K+1)=-V1*U(K+1)                                                     PRO 32
DO 40 K=1,6                                                           PRO 33
35 Z(K)=U(K)+V(K)                                                     PRO 34
40 REAL(3)/3.-RA(4))                                                  PRO 35
DO 45 I=1,6,2                                                         PRO 36
45 X(I+1)=Z(I+1)                                                      PRO 37
X(I)=Z(I)-R                                                           PRO 38
RETURN                                                                PRO 39
END                                                                   PRO 40
                                                                     PRO 41

C***   SUBROUTINE POINT(X,Y,XINT,YINT,NN1,NN2,A)                      PDT 1
C***   ROUTINE FOR PARABOLIC OR LINEAR INTERPOLATION, GIVEN X AND Y   PDT 2
C***   ARRAYS THE ROUTINE CALCULATES THE VALUE YINT AT THE GIVEN VALUE PDT 3
C***   XINT. IT ALSO RETURNS THE COEFFICIENTS OF THE FIT IN ARRAY A.  PDT 4
       IMPORTANT NOTE: THE X ARRAY MUST BE MONOTONIC INCREASING.      PDT 5
       REAL*8 A2,A3,A                                                 PDT 6
       DIMENSION X(NN2),Y(NN2),A(4)                                   PDT 7
       DIMENSION XP(3),YP(3)                                          PDT 8
       NP=NN2-NN1+1                                                   PDT 9
       IF (NP-2) 55,60,5                                              PDT 10
5      DO 10 I=NN1,NN2                                                PDT 11
       KK=I                                                           PDT 12
       IF (XINT-X(I)) 15,25,10                                        PDT 13
10     CONTINUE                                                       PDT 14
15     XR=X(KK-1)+(X(KK)-X(KK-1))/2.                                  PDT 15
       IF (XINT-XR) 20,25,25                                          PDT 16
20     KK=KK-1                                                        PDT 17
       IF (KK.LT.NN1)KK=NN1                                           PDT 18
25     A(4)=KK                                                        PDT 19
       N2=KK-1                                                        PDT 20
       IF (N2-NN1) 30,30,35                                           PDT 21
30     N2=NN1                                                         PDT 22
       GO TO 45                                                       PDT 23
```

```
35     IF (NN2-(N2+2)) 40,45,45                                       PDT 24
40     N2=NN2-2                                                       PDT 25
45     XPB=XINT-X(N2+1)                                               PDT 26
       DO 50 K=1,3                                                    PDT 27
       L=N2+K-1                                                       PDT 28
       XP(K)=X(L)-X(N2+1)                                             PDT 29
       YP(K)=Y(L)-Y(N2+1)                                             PDT 30
50     A2=(YP(1)*XP(3)/XP(1)-YP(1)*XP(1)/XP(3))/(XP(1)-XP(3))         PDT 31
       A3=(YP(1)/XP(1)-YP(3)/XP(3))/(XP(1)-XP(3))                     PDT 32
       YINT=Y(N2+1)-A2*XPB+A3*XPB*XPB                                 PDT 33
       A(1)=Y(N2+1)-A2*X(N2+1)+A3*X(N2+1)*X(N2+1)                     PDT 34
       A(2)=A2-2.*A3*X(N2+1)                                          PDT 35
       A(3)=A3                                                        PDT 36
       RETURN                                                         PDT 37
55     YINT=Y(NN1)                                                    PDT 38
       A(1)=YINT                                                      PDT 39
       A(2)=0.0                                                       PDT 40
       A(3)=0.0                                                       PDT 41
       RETURN                                                         PDT 42
60     A(2)=(Y(NN2)-Y(NN1))/(X(NN2)-X(NN1))                           PDT 43
       A(1)=Y(NN1)-A(2)*X(NN1)                                        PDT 44
       YINT=A(1)+A(2)*XINT                                            PDT 45
       A(3)=0.0                                                       PDT 46
       RETURN                                                         PDT 47
       END                                                            PDT 48

       SUBROUTINE SMOOTH(X,Y,N)                                       SMT 1
C***   ROUTINE FOR LINEAR LEAST SQUARES SMOOTHING OF THE N POINTS OF  SMT 2
C***   THE X AND Y ARRAYS                                             SMT 3
       DIMENSION X(N),Y(N)                                            SMT 4
       SF=0.5                                                         SMT 5
       IF (N.LE.3) RETURN                                            SMT 6
       A=0.                                                           SMT 7
       B=0.                                                           SMT 8
       NE=N-3                                                         SMT 9
       DO 15 J=1,NE                                                   SMT 10
       AP=A                                                           SMT 11
       BP=B                                                           SMT 12
       SX=0.                                                          SMT 13
       SY=0.                                                          SMT 14
       SXY=0.                                                         SMT 15
       SX2=0.                                                         SMT 16
       J3=J+3                                                         SMT 17
       DO 5 I=J,J3                                                    SMT 18
       SX=SX+X(I)                                                     SMT 19
       SY=SY+Y(I)                                                     SMT 20
       SXY=SXY+X(I)*Y(I)                                              SMT 21
       SX2=SX2+X(I)**2                                                SMT 22
5      SX2=SX2+X(I)**2                                                SMT 23
       A=(SX*SXY-SX2*SY)/(SX*SX-4.*SX2)                               SMT 24
       B=(SX*SY-4.*SXY)/(SX*SX-4.*SX2)                                SMT 25
       IF (J.NE.1) GO TO 10                                           SMT 26
       Y(J+1)=(1.-SF)*Y(J+1)+SF*(A+B*X(J+1))                          SMT 27
       GO TO 15                                                       SMT 28
10     Y(J+1)=(1.-SF)*Y(J+1)+SF*(A+AP+(B+BP)*X(J+1))/2.              SMT 29
15     CONTINUE                                                       SMT 30
       Y(N-1)=(1.-SF)*Y(N-1)+SF*(A+B*X(N-1))                          SMT 31
       RETURN                                                         SMT 32
       END                                                            SMT 32
```

289

```
C*** SUBROUTINE FTLUP(X,Y,M,N,VARI,VARD)
C*** ROUTINE FOR LINEAR OR PARABOLIC INTERPOLATION. IT CALCULATES THE
C*** VALUE Y AT POINT X (INDEPENDENT) FROM ARRAYS VARI (INDEPENDENT)
C*** AND VARD. N IS THE NUMBER OF POINTS IN THE ARRAYS. M=1 GIVES A
C*** PARABOLIC INTERPOLATION WHILE M=2 GIVES A LINEAR INTERPOLATION.
      COMMON /INOUT/JREAD,JWRITE
      DIMENSION VARI(N),VARD(N),VI(3),VY(2)
      DIMENSION II(43)
      DATA II/43*-1/
      MA=IABS(M)
      LI=1
      I=II(LI)
      IF (I.GE.0.OR.N.LT.2) GO TO 35
      IF (VARI(2)-VARI(1)) 15,15,25
    5 R=VARI(J)
      S=VARD(J)
      WRITE (JWRITE,10) R,S,(VARI(J),J=1,N),(VARD(J),J=1,N)
   10 FORMAT (1H1,52HOTABLE BELOW OUT OF ORDER FOR FTLUP AT ELEMENT VARI
     1(,I3,4H) = ,G15.8,6H VARD(,I3,4H) = ,G15.8,//(8G15.8))
      STOP
   15 DO 20 J=2,N
      IF (VARI(J)-VARI(J-1)) 20,5,5
   20 CONTINUE
      GO TO 35
   25 DO 30 J=2,N
      IF (VARI(J)-VARI(J-1)) 5,5,30
   30 CONTINUE
   35 IF (I.LE.0)I=1
      IF (I.GE.N)I=N-1
      IF (N.LE.1) GO TO 40
      IF (MA.NE.0) GO TO 45
   40 Y=VARD(1)
      GO TO 85
   45 IF ((VARI(I)-X)*(VARI(I+1)-X)) 60,60,50
   50 IN=SIGN(1.0,(VARI(I+1)-VARI(I))*(X-VARI(I)))
   55 IF ((I+IN).LE.0) GO TO 60
      IF ((I+IN).GE.N) GO TO 60
      I=I+IN
      GO TO 85
   60 IF ((VARI(I)-X)*(VARI(I+1)-X)) 60,60,55
      Y=(VARD(I)*(VARI(I+1)-X)-VARD(I+1)*(VARI(I)-X))/(VARI(I+1)-VARI(I))
     1)
      GO TO 85
   65 IF (M.EQ.2) GO TO 5
      IF (I.EQ.(N-1)) GO TO 75
      IF (I.EQ.1) GO TO 70
      SK=VARI(I+1)-VARI(I)
      IF ((SK*(X-VARI(I-1))).LT.(SK*(VARI(I+2)-X))) GO TO 75
   70 L=I
      GO TO 80
   75 L=I-1
   80 VI(1)=VARI(L)-X
      VI(2)=VARI(L+1)-X
      VI(3)=VARI(L+2)-X
      VY(1)=(VARD(L)*VI(2)-VARD(L+1)*VI(1))/(VARI(L+1)-VARI(L))
      VY(2)=(VARD(L+1)*VI(3)-VARD(L+2)*VI(2))/(VARI(L+2)-VARI(L+1))
      Y=(VY(1)*VI(3)-VY(2)*VI(1))/(VARI(L+2)-VARI(L))
   85 II(LI)=I
      RETURN
      END
```

```
      SUBROUTINE SIMSOL(B,N,IERR)
C*** ROUTINE SOLVES THE LINEAR SIMULTANEOUS EQUATION AX=B IN N UNKNOWNS
C*** AND RETURNS THE SOLUTIONS IN B
      COMMON /COEF/A(65,65)
      DIMENSION B(65)
      DO 55 J=1,N
      JJ=J+1
C*** CHECK FOR SINGULARITY
      IF (A(J,J)) 35,5,35
    5 IF (J-N) 10,20,20
   10 DO 15 I=JJ,N
      IF (A(I,J)) 25,15,25
   15 CONTINUE
   20 IERR=1
      RETURN
   25 A(J,J) = 0   INTERCHANGE ROWS
      DO 30 L=J,N
      T=A(J,L)
      A(J,L)=A(I,L)
   30 A(I,L)=T
      T=B(J)
      B(J)=B(I)
      B(I)=T
C*** DIVIDE JTH ROW BY A(J,J)
   35 T=A(J,J)
      DO 40 KK=J,N
   40 A(J,KK)=A(J,KK)/T
      B(J)=B(J)/T
C*** GET A(I,J) = 0, I=1,.....,J-1,J+1,....,N
      DO 55 I=1,N
      IF (I-J) 45,55,45
   45 T=A(I,J)
      DO 50 KK=J,N
   50 A(I,KK)=A(I,KK)-T*A(J,KK)
      B(I)=B(I)-T*B(J)
   55 CONTINUE
      RETURN
      END
```

***** CASE INPUT *****

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0

NXU	NXL	IWRITE	IALPHA	IPUNCH
18	18	0	0	C

XU = 0.0   0.125C00E-01 0.250000E-01 0.500000E-01 0.750000E-01 0.100000E 00 0.150000E 00 0.200000E 00
     0.250000E 00 0.300000E 00 0.400000E 00 0.500000E 00 0.600000E 00 0.700000E 00 0.800000E 00 0.900000E 00
     0.950000E 00 0.100000E 01

ZU = 0.0   0.267000E-01 0.361000E-01 0.491000E-01 0.580000E-01 0.642990E-01 0.719000E-01 0.750000E-01
     0.740000E-01 0.755000E-01 0.714000E-01 0.614000E-01 0.547000E-01 0.436000E-01 0.308000E-01 0.168000E-01
     0.420000E-02 0.130000E-02

XL = 0.0   0.125000E-01 0.250000E-01 0.500000E-01 0.750000E-01 0.100000E 00 0.150000E 00 0.200000E 00
     0.250000E 00 0.300000E 00 0.4C0000E 00 0.500000E 00 0.600000E 00 0.700000E 00 0.800000E 00 0.900000E 00
     0.950000E 00 0.100000E 01

ZL = 0.0   -0.123000E-01-0.171000E-01-0.226000E-01-0.261000E-01-0.292000E-01-0.350000E-01-0.397000E-01
    -0.428000E-01-0.444000E-01-0.448000E-01-0.417000E-01-0.367000E-01-0.300000E-01-0.216000E-01-0.123000E-01
    -0.700000E-02-0.130000E-02

NA = 1   ANGLES OF ATTACK W.R.T. REFERENCE LINE (IALPHA=C) OR W.R.T. LONGEST CHORDLINE (IALPHA=1)

          0.0

NR = 1   FSMACH   = 0.200000E 00

CREF = 0.100000E 01   SF = 0.100000E 01   TO = 0.518690E 03   RN = 0.300000E 01   PR = 0.770000E 00   KF = 0.100000E 01

	LTRAN	XTRAN	ZTRAN
UPPER SURFACE	0	0.0	0.0
LOWER SURFACE	0	0.0	0.0

NACA 23012 AIRFOIL/ALPHA=0/RN=3.C/MACH NO.=0.2/IWRITE=0

INPUT AIRFOIL POINTS

UPPER SURFACE		LOWER SURFACE	
XU	ZU	XL	ZL
0.0	0.0	0.0	0.0
0.012500	0.026700	0.012500	-0.012300
0.025000	0.036100	0.025000	-0.017100
0.050000	0.049100	0.050000	-0.022600
0.075000	0.058000	0.075000	-0.026100
0.100000	0.064300	0.100000	-0.029200
0.150000	0.071900	0.150000	-0.035000
0.200000	0.075000	0.200000	-0.039700
0.250000	0.074C00	0.250000	-0.042800
0.300000	0.075500	0.300000	-0.044400
0.400000	0.071400	0.400000	-0.044800
0.500000	0.061400	0.500000	-0.041700
0.600000	0.054700	0.600000	-0.036700
0.700000	0.043600	0.700000	-0.030000
0.800000	0.030800	0.800000	-0.021600
0.900000	0.016800	0.900000	-0.012300
0.950000	0.009200	0.950000	-0.007000
1.000000	0.001300	1.000000	-0.001300

NACA 23012 AIRFOIL/ALPHA=0/RN=3.C/MACH NO.=0.2/IWRITE=0

DISTRIBUTED AIRFOIL POINTS

UPPER SURFACE		LOWER SURFACE	
XU	ZU	XL	ZL
0.001562	0.004283	0.999994	-0.001301
0.005084	0.012948	0.953537	-0.006610
0.011934	0.025865	0.907505	-0.011530
0.024014	0.035987	0.860744	-0.016157
0.039488	0.044576	0.812696	-0.020469
0.057524	0.052210	0.764660	-0.024671
0.077592	0.058774	0.715153	-0.028036
0.099321	0.064163	0.671088	-0.032112
0.122524	0.068569	0.632369	-0.034717
0.147233	0.071597	0.592982	-0.037106
0.173675	0.073929	0.554524	-0.039185
0.202774	0.075110	0.517126	-0.040978
0.234705	0.075853	0.480190	-0.042445
0.248279	0.075991	0.444219	-0.043836
0.303436	0.075427	0.408438	-0.044666
0.338303	0.074418	0.373269	-0.045190
0.373269	0.072901	0.338303	-0.045212
0.408438	0.070704	0.303436	-0.044682
0.444215	0.067706	0.268279	-0.043609
0.480190	0.063119	0.234705	-0.041990
0.517126	0.060018	0.202774	-0.039914
0.554524	0.058292	0.173675	-0.037363
0.592982	0.055314	0.147233	-0.034700
0.632369	0.051599	0.122524	-0.031079
0.671088	0.046984	0.099321	-0.029121
0.715153	0.041770	0.077592	-0.026440
0.764660	0.035461	0.057524	-0.023864
0.812696	0.029089	0.039488	-0.020953
0.860744	0.022487	0.024014	-0.016494
0.907505	0.015678	0.011934	-0.011905
0.953537	0.008651	0.005084	-0.005908
0.999994	0.001301	0.001562	-0.001948
		0.0	0.0

Figure A-3. Sample output of the 2-D characteristics program with IWRITE=0 (note that only upper surface inviscid and boundary layer solution information is shown here).

INVISCID FLOW SOLUTION

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0

MACH NUMBER = 0.20000   ANGLE OF ATTACK =   0.0   DEG.

UPPER SURFACE

ITERATION NUMBER 0

X	Z	(V/VO)INC.	(V/VO)COMPR.	ML	CP
0.00156	0.00420	0.04319	0.04275	0.00852	1.00014
0.00500	0.01245	0.46200	0.45836	0.09138	0.79613
0.01193	0.02586	0.84240	0.83988	0.16778	0.29543
0.02401	0.03599	1.09624	1.09647	0.21947	-0.20164
0.03949	0.04458	1.18249	1.18736	0.23786	-0.40815
0.05792	0.05221	1.23555	1.24229	0.24900	-0.54034
0.07759	0.05877	1.26977	1.27764	0.25622	-0.62887
0.09932	0.06416	1.28023	1.29810	0.26033	-0.68036
0.12292	0.06899	1.29438	1.30346	0.26142	-0.69413
0.14723	0.07160	1.28600	1.29473	0.29065	-0.67177
0.17360	0.07393	1.26001	1.27704	0.25606	-0.62686
0.20277	0.07511	1.24493	1.25203	0.25090	-0.56436
0.23471	0.07583	1.22585	1.23223	0.24696	-0.51571
0.26826	0.07599	1.21233	1.21823	0.24412	-0.48173
0.30344	0.07543	1.19001	1.20444	0.24132	-0.44864
0.33036	0.07442	1.18087	1.19396	0.23920	-0.42373
0.37327	0.07290	1.18194	1.18679	0.23779	-0.40680
0.40844	0.07078	1.17645	1.17906	0.23618	-0.38867
0.44421	0.06771	1.14380	1.14745	0.22970	-0.31563
0.48019	0.06312	1.09372	1.09594	0.21936	-0.20067
0.51713	0.06002	1.08252	1.08444	0.21704	-0.17570
0.55452	0.05828	1.10668	1.10904	0.22201	-0.22943
0.59296	0.05531	1.11493	1.11732	0.22369	-0.24777
0.63237	0.05159	1.10234	1.10479	0.22113	-0.22007
0.67109	0.04698	1.08224	1.08415	0.21698	-0.17907
0.71513	0.04177	1.06417	1.06563	0.21324	-0.13956
0.76466	0.03544	1.04739	1.04844	0.20977	-0.09911
0.81270	0.02909	1.02870	1.02932	0.20591	-0.05945
0.86074	0.02249	1.00898	1.00917	0.20185	-0.01841
0.90791	0.01568	0.98317	0.98263	0.19654	0.03399
0.95394	0.00865	0.94344	0.94230	0.18839	0.11149
0.99999	0.00130	0.92171	0.92028	0.18394	0.15330

LAMINAR BOUNDARY LAYER SUMMARY

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0

UPPER SURFACE
ITERATION NUMBER 0

STAGNATION TEMPERATURE   = 518.69   DEGREES RANKINE   STAGNATION PRESSURE   = 4478.688   LB/SQ FT
FREESTREAM MACH NUMBER   = 0.20000   AIRFOIL CHORD   = 1.00000   FT
REYNOLDS NUMBER   = 3.00000   MILLION   PRANDTL NUMBER   = 0.77000
HEAT TRANSFER FACTOR K   = 1.00000   SEPARATION CORRELATION NO.   = 0.04937
SPEED OF SOUND   = 1116.419 FEET/SECOND   KINEMATIC VISCOSITY   = 0.0000743 SQ FT/SEC
ANGLE OF ATTACK (ALPHA)   = 0.0   DEGREES
STAGNATION AT X/C   = 0.0013063   STAGNATION AT Z/C   = 0.0036359

X/C	S/C	N	DM/D(S/C)	H	THETA/C	DELS/C	DELTAS/C	CF	CP
0.001562	0.000697	0.008516	11.982389	2.500737	0.000017	0.000063		-0.000003	1.00014
0.005834	0.010050	0.041384	7.442142	2.506000	0.000022	0.000095		0.000967	0.79613
0.011934	0.024671	0.107779	4.289056	2.496595	0.000026	0.000064		0.002794	0.29543
0.024014	0.040430	0.219472	2.224560	2.480503	0.000029	0.000073		0.004001	-0.20164
0.039600	0.050127	0.237063	0.815860	2.464336	0.000039	0.000096		0.004126	-0.40815
0.097924	0.077711	0.244000	0.439911	2.462024	0.000047	0.000117		0.003727	-0.54034
0.077502	0.094829	0.254217	0.265139	2.497791	0.000054	0.000137		0.003519	-0.62887
0.099321	0.121212	0.260333	0.116874	2.440255	0.000064	0.000157		0.002824	-0.68036
0.122524	0.144825	0.261424	-0.010993	2.431207	0.000073	0.000178		0.002205	-0.69413
0.147233	0.169721	0.259650	-0.102307	2.409275	0.000084	0.000202		0.001565	-0.67177
0.173675	0.196265	0.256056	-0.153993	2.385283	0.000096	0.000229		0.000936	-0.62686
0.202774	0.223388	0.250977	-0.151215	2.369037	0.000110	0.000261		0.000950	-0.56436
0.234709	0.257327	0.244899	-0.105716	2.375357	0.000123	0.000293		0.000569	-0.51571
0.268279	0.290900	0.244118	-0.082107	2.377635	0.000136	0.000322		0.000560	-0.48173

SCRIT/C = 0.11799729   RECRIT =   454.51   STRAN/C = 0.28501159   RTRAN = 920.40   THETA1/C = 0.00013359

INSTABILITY TRANSITION HAS OCCURRED AT X/C = 0.262390   S/C = 0.285012

TURBULENT BOUNDARY LAYER SUMMARY FOR EQUIVALENT AIRFOIL   (GORADIA METHOD)

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0

UPPER SURFACE
ITERATION NUMBER 0

STAGNATION TEMPERATURE   = 518.69   DEGREES RANKINE   STAGNATION PRESSURE   = 4478.688   LB/SQ FT
FREESTREAM MACH NUMBER   = 0.20000   AIRFOIL CHORD   = 1.00000   FT
REYNOLDS NUMBER   = 3.00000   MILLION   PRANDTL NUMBER   = 0.77000
HEAT TRANSFER FACTOR K   = 1.00000   TRANSITION POINT, S/C   = 0.28501
(INITIAL MOMENTUM THICKNESS)/C   = 0.00013   INIT. INCOMPR. FORM FACTOR   = 1.35474
ANGLE OF ATTACK (ALPHA)   = 0.0   DEGREES

X/C	S/C	N	DM/(S/C)	H	THETA/C	DELS/C	DELTAS/C	CF	CP
0.262390	0.285012	0.244604	-0.082907	1.354743	0.000134	0.000181			-0.481733
0.268279	0.290900	0.244118	-0.082107	1.366482	0.000147	0.000201	0.001298	0.006670	-0.449644
0.303436	0.326061	0.241324	-0.070153	1.532023	0.000226	0.000346	0.001644	0.004584	-0.423725
0.338303	0.360943	0.239199	-0.051209	1.617521	0.000300	0.000489	0.002057	0.003596	-0.466804
0.373269	0.395942	0.237747	-0.042960	1.515099	0.000371	0.000563	0.002747	0.003945	-0.388667
0.404630	0.431174	0.236181	-0.110713	1.391617	0.000663	0.000616	0.003764	0.004311	-0.313626
0.444215	0.467083	0.229701	-0.232464	1.370185	0.000547	0.000740	0.004798	0.004207	-0.200672
0.480190	0.503349	0.219364	-0.176170	1.360901	0.000695	0.000946	0.006191	0.003697	-0.179704
0.517124	0.540415	0.217041	0.034563	1.332398	0.000781	0.001040	0.007297	0.003600	-0.229435
0.554524	0.577052	0.222012	0.080749	1.311455	0.000780	0.001033	0.007670	0.003942	-0.247772
0.592902	0.616426	0.223606	-0.009642	1.305090	0.000832	0.001086	0.008206	0.003974	-0.220067
0.632369	0.659988	0.221152	-0.085651	1.306720	0.000926	0.001210	0.009008	0.003778	-0.175065
0.671080	0.694079	0.214982	-0.096362	1.308531	0.001044	0.001366	0.010221	0.003931	-0.135370
0.715193	0.739391	0.213241	-0.077359	1.307617	0.001171	0.001531	0.011485	0.003329	-0.099116
0.764660	0.780258	0.209258	-0.074661	1.306360	0.001308	0.001710	0.012862	0.003144	-0.059949
0.812694	0.837714	0.209912	-0.001703	1.307045	0.001459	0.001907	0.014326	0.002950	-0.050414
0.860744	0.886213	0.201844	-0.098253	1.310473	0.001623	0.002127	0.015830	0.002703	0.003992
0.907969	0.933468	0.196540	-0.143903	1.322367	0.001851	0.002421	0.017443	0.002508	0.111969
0.953537	0.980033	0.188391	-0.134970	1.334863	0.002152	0.002872	0.020026	0.002190	0.193305
0.999994	1.027067	0.183944	-0.054105	1.348601	0.002301	0.003211	0.021643	0.002082	

Figure A-3. Continued.

292

LOAD SUMMARY SHEET--ITERATION NUMBER 0

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0

MACH NUMBER   = 0.20000   REFERENCE CHORD (BASIS FOR ALL COEFFICIENT NORMALIZATION) = 1.00000 FT.

REYNOLDS NO.   = 3.00000 MILLION   ANGLE OF ATTACK (VALUE CF IALPHA = 0)   = 0.0   DEG.
                                          (IALPHA=0--ANGLE DEFINED W.R.T. REFERENCE LINE)
LONGEST CHORDLINE = 1.00000 FEET   (IALPHA=1--ANGLE DEFINED W.R.T. LONGEST CHORDLINE)

ANGLE BETWEEN LONGEST CHORDLINE AND REFERENCE LINE = 0.0   DEG. (POSITIVE FOR REFERENCE LINE BELOW THE LONGEST CHORDLINE)

```
                                        ************************************************
                                        *                                              *
NORMAL FORCE COEFFICIENTS               * LIFT COEFFICIENTS                             *
           (CN)PRESS. =  0.13344        *            (CL)PRESS. =  0.13344              *
           (CN)SHEAR  = -0.00003        *            (CL)SHEAR  = -0.00003              *
           CN         =  0.13341        *            CL         =  0.13341              *
                                        *                                              *
AXIAL FORCE COEFFICIENTS                * DRAG COEFFICIENTS                             *
           (CA)PRESS. =  0.0            *            (CD)PRESS. =  0.0                  *
           (CA)SHEAR  =  0.00574        *            (CD)SHEAR  =  0.00574              *
           CA         =  0.00574        *            CD         =  0.00574              *
                                        *                                              *
MOMENT COEFFICIENTS                     * MOMENT COEFFICIENTS                           *
 (ABOUT NOSE)  (CM)PRESS. = -0.04252    *            (CM)ABOUT NOSE    = -0.04237       *
               (CM)SHEAR  =  0.00015    *            (CM)QUARTER CHORD = -0.00902       *
               CM         = -0.04237    *                                              *
                                        ************************************************
```

DRAG COEFFICIENT COMPUTED BY SQUIRE-YOUNG FORMULA = 0.00637

ZEROTH ITERATION PRESSURE DRAG = 0.00194

TRANSITION POINTS: X/C(LOWER) = 0.39225   X/C(UPPER) = 0.26239

LAMINAR BOUNDARY LAYER SUMMARY

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0   UPPER SURFACE
                                                                                          ITERATION NUMBER 4

```
STAGNATION TEMPERATURE    = 518.69     DEGREES RANKINE     STAGNATION PRESSURE       = 4478.688   LB/SQ FT
FREESTREAM MACH NUMBER    = 0.20000                        AIRFOIL CHORD             = 1.00000    FT
REYNOLDS NUMBER           = 3.00000    MILLION             PRANDTL NUMBER            = 0.77000
HEAT TRANSFER FACTOR K    = 1.00000                        SEPARATION CORRELATION NO.= 0.069637
SPEED OF SOUND            = 1116.419   FEET/SECOND         KINEMATIC VISCOSITY       = 0.0000743 SQ FT/SEC
ANGLE OF ATTACK (ALPHA)   = 0.0        DEGREES
STAGNATION AT X/C         = 0.0013530                      STAGNATION AT Z/C         = 0.0037539
```

X/C	S/C	M	DM/D(S/C)	H	THETA/C	DELS/C	DELTAS/C	CF	CP
0.001562	0.000570	0.006945	11.988713	2.500720	0.000017	0.000043		-0.000002	1.00875
0.005084	0.009923	0.049079	7.450247	2.504625	0.000022	0.000055		0.000543	0.80221
0.011934	0.024544	0.166428	4.290575	2.496690	0.000026	0.000064		0.002726	0.30671
0.024014	0.040303	0.218224	2.232097	2.480362	0.000029	0.000073		0.004725	-0.18038
0.039488	0.058001	0.236769	0.622052	2.464285	0.000039	0.000096		0.004079	-0.39545
0.057524	0.077584	0.248006	0.463543	2.461936	0.000047	0.000117		0.003690	-0.52832
0.077592	0.098649	0.255203	0.267474	2.457661	0.000056	0.000137		0.003289	-0.61729
0.099321	0.121085	0.259439	0.118551	2.448127	0.000064	0.000157		0.002800	-0.66913
0.122524	0.144698	0.260567	-0.009007	2.431306	0.000073	0.000178		0.002191	-0.68332
0.147233	0.169594	0.258853	-0.099084	2.409523	0.000084	0.000202		0.001541	-0.66176
0.173675	0.196130	0.255524	-0.152524	2.385482	0.000096	0.000229		0.000937	-0.61779
0.202774	0.225261	0.250257	-0.153313	2.360759	0.000110	0.000260		0.000531	-0.55559
0.234705	0.257200	0.246084	-0.111018	2.372191	0.000124	0.000293		0.000523	-0.50921
0.268279	0.290773	0.243049	-0.083411	2.376044	0.000136	0.000323		0.000517	-0.46903

SCRIT/C = 0.11828500   RECRIT = 456.30   STRAN/C = 0.28427839   RTRAN = 919.04   THETA1/C = 0.00013387

INSTABILITY TRANSITION HAS OCCURRED AT   X/C = 0.261784   S/C = 0.284278

TURBULENT BOUNDARY LAYER SUMMARY FOR EQUIVALENT AIRFOIL   (GORADIA METHOD)

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0   UPPER SURFACE
                                                                                          ITERATION NUMBER 4

```
STAGNATION TEMPERATURE        = 518.69   DEGREES RANKINE     STAGNATION PRESSURE       = 4478.688   LB/SQ FT
FREESTREAM MACH NUMBER        = 0.20000                      AIRFOIL CHORD             = 1.00000    FT
REYNOLDS NUMBER               = 3.00000  MILLION             PRANDTL NUMBER            = 0.77000
HEAT TRANSFER FACTOR K        = 1.00000                      TRANSITION POINT, S/C     = 0.28428
(INITIAL MOMENTUM THICKNESS)/C= 0.00013                      INIT. INCOMPR. FORM FACTOR= 1.35291
ANGLE OF ATTACK (ALPHA)       = 0.0      DEGREES
```

X/C	S/C	M	DM/(S/C)	H	THETA/C	DELS/C	DELTAS/C	CF	CP
0.261784	0.284278	0.243600	-0.086115	1.352907	0.000134	0.000181			
0.268279	0.290773	0.243049	-0.083411	1.365986	0.000149	0.000203	0.001314	0.006604	-0.469033
0.303434	0.325934	0.240374	-0.063882	1.526860	0.000227	0.000347	0.001662	0.004502	-0.437479
0.330303	0.360816	0.238560	-0.043746	1.611150	0.000301	0.000484	0.002069	0.003613	-0.416360
0.373249	0.395815	0.237319	-0.030436	1.511716	0.000371	0.000561	0.002752	0.003954	-0.401831
0.408438	0.431047	0.235067	-0.106423	1.390843	0.000442	0.000615	0.003761	0.004509	-0.345038
0.444215	0.466956	0.229644	-0.227090	1.369104	0.000545	0.000746	0.004787	0.004214	-0.314093
0.480190	0.503222	0.219438	-0.173104	1.359771	0.000691	0.000940	0.006165	0.003712	-0.201474
0.517126	0.540288	0.217125	0.031501	1.332047	0.000776	0.001034	0.007260	0.003690	-0.176594
0.554524	0.577725	0.221656	0.082690	1.311716	0.000786	0.001032	0.007650	0.003938	-0.227728
0.592982	0.614299	0.223310	-0.012770	1.305539	0.000833	0.001088	0.008206	0.003959	-0.243649
0.632349	0.655861	0.220756	-0.084437	1.306960	0.000927	0.001212	0.009105	0.003764	-0.215750
0.671088	0.694852	0.216700	-0.092814	1.308351	0.001044	0.001365	0.010222	0.003525	-0.172051
0.715153	0.739224	0.213140	-0.072805	1.306945	0.001168	0.001526	0.011449	0.003332	-0.134391
0.764440	0.789131	0.209921	-0.069186	1.305291	0.001301	0.001698	0.012819	0.003162	-0.100683
0.812696	0.837587	0.206355	-0.075554	1.305115	0.001444	0.001885	0.014243	0.002986	-0.063975
0.860744	0.886086	0.202596	-0.092798	1.307091	0.001601	0.002094	0.015694	0.002802	-0.025907
0.907505	0.933341	0.197507	-0.137908	1.318651	0.001800	0.002374	0.017275	0.002556	0.024557
0.953537	0.979906	0.189699	-0.132586	1.330318	0.002105	0.002800	0.019793	0.002245	0.099625
0.999994	1.026940	0.185130	-0.061677	1.342957	0.002334	0.003135	0.021416	0.002053	0.142302

Figure A-3. Continued.

293

LOAD SUMMARY SHEET--ITERATION NUMBER 4

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0

MACH NUMBER   = 0.20000   REFERENCE CHORD (BASIS FOR ALL COEFFICIENT NORMALIZATION) =   1.00000 FT.

REYNOLDS NO.   = 3.0CC00 MILLION ANGLE OF ATTACK (VALUE CF IALPHA = 0)   =   0.0   DEG.
                                        (IALPHA=0--ANGLE DEFINED W.R.T. REFERENCE LINE)
LONGEST CHORDLINE = 1.00000 FEET   (IALPHA=1--ANGLE DEFINED W.R.T. LONGEST CHORDLINE)

ANGLE BETWEEN LONGEST CHORDLINE AND REFERENCE LINE = 0.0   DEG. (POSITIVE FOR REFERENCE LINE BELOW THE LONGEST CHORDLINE)

NORMAL FORCE COEFFICIENTS   * LIFT COEFFICIENTS   *
             (CN)PRESS. = 0.12541   *   (CL)PRESS. = 0.12541   *
             (CN)SHEAR = -0.00003   *   (CL)SHEAR = -0.00003   *
              CN   = 0.12538   *   CL   = 0.12538   *

AXIAL FORCE COEFFICIENTS   * DRAG COEFFICIENTS   *
             (CA)PRESS. = 0.00085   *   (CO)PRESS. = 0.00085   *
             (CA)SHEAR = 0.00577   *   (CO)SHEAR = 0.00577   *
              CA   = 0.00662   *   CO   = 0.00662   *

MOMENT COEFFICIENTS   * MOMENT COEFFICIENTS   *
   (ABOUT NOSE) (CM)PRESS. = -0.04000   *   (CM)ABOUT NOSE = -0.03985   *
                 (CM)SHEAR = 0.00015   *   (CM)QUARTER CHORD = -0.00850 *
                  CM   = -0.03985   *

DRAG COEFFICIENT COMPUTED BY SQUIRE-YOUNG FORMULA = 0.00643

ZEROTH ITERATION PRESSURE DRAG = 0.00194

TRANSITION POINTS: X/C(LOWER) = C.38373   X/C(UPPER) = 0.26178

LONGEST CHORDLINE SYSTEM		BASIC REFERENCE SYSTEM						
X/C	Z/C	X/C	Z/C	S/C	CP	CF	DELTAS/C	V/VO
0.99999	-0.00130	0.99999	-0.00130	0.0	0.14230	0.00226	0.00233	0.92612
0.95354	-0.00661	0.95354	-0.00661	0.04676	0.09809	0.00247	0.00213	0.94969
0.90751	-0.01153	0.90751	-0.01153	0.09305	0.03639	0.00275	0.00194	0.98062
0.86074	-0.01616	0.86074	-0.01616	0.14004	0.00072	0.00295	0.00174	0.99964
0.81270	-0.02047	0.81270	-0.02047	0.18828	-0.02467	0.00310	0.00154	1.01226
0.76466	-0.02467	0.76466	-0.02467	0.23650	-0.05181	0.00324	0.00133	1.02558
0.71515	-0.02884	0.71515	-0.02884	0.28619	-0.08024	0.00343	0.00113	1.03935
0.67109	-0.03211	0.67109	-0.03211	0.33037	-0.10322	0.00359	0.00100	1.05034
0.63237	-0.03472	0.63237	-0.03472	0.36918	-0.12012	0.00371	0.00090	1.05836
0.59298	-0.03711	0.59298	-0.03711	0.40864	-0.13451	0.00383	0.00080	1.06513
0.55452	-0.03918	0.55452	-0.03918	0.44715	-0.14662	0.00393	0.00070	1.07174
0.51713	-0.04098	0.51713	-0.04098	0.48459	-0.16106	0.00391	0.00062	1.07752
0.48019	-0.04246	0.48019	-0.04246	0.52158	-0.17639	0.00372	0.00054	1.08461
0.44421	-0.04384	0.44421	-0.04384	0.55756	-0.19617	0.00402	0.00043	1.09369
0.40844	-0.04467	0.40844	-0.04467	0.59335	-0.21291	0.00506	0.00030	1.10132
0.37327	-0.04519	0.37327	-0.04519	0.62852	-0.23032	0.00044	0.00038	1.10920
0.33830	-0.04521	0.33830	-0.04521	0.66349	-0.24519	0.00057	0.00036	1.11588
0.30344	-0.04468	0.30344	-0.04468	0.69836	-0.25229	0.00070	0.00034	1.11906
0.26828	-0.04361	0.26828	-0.04361	0.73353	-0.25448	0.00080	0.00032	1.12004
0.23471	-0.04199	0.23471	-0.04199	0.76714	-0.25240	0.00090	0.00030	1.11911
0.20277	-0.03991	0.20277	-0.03991	0.79914	-0.24303	0.00103	0.00028	1.11491
0.17368	-0.03736	0.17368	-0.03736	0.82835	-0.22327	0.00112	0.00027	1.10602
0.14723	-0.03471	0.14723	-0.03471	0.85493	-0.19055	0.00116	0.00025	1.09478
0.12252	-0.03188	0.12252	-0.03188	0.87980	-0.17563	0.00024	0.00024	1.08426
0.09932	-0.02912	0.09932	-0.02912	0.90316	-0.16422	0.00102	0.00022	1.07899
0.07759	-0.02644	0.07759	-0.02644	0.92506	-0.18051	0.00084	0.00019	1.08642
0.05752	-0.02386	0.05752	-0.02386	0.94529	-0.23901	0.00093	0.00015	1.11511
0.03949	-0.02095	0.03949	-0.02095	0.96356	-0.31952	0.00105	0.00012	1.14470
0.02401	-0.01699	0.02401	-0.01699	0.97953	-0.36248	0.00381	0.00008	1.16725
0.01193	-0.01191	0.01193	-0.01191	0.99264	-0.22978	0.00570	0.00006	1.10896
0.00508	-0.00591	0.00508	-0.00591	1.00174	0.13021	0.00450	0.00005	0.93262
0.00156	-0.00195	0.00156	-0.00195	1.00704	0.38123	0.00506	0.00003	0.78662
0.0	0.0	0.0	0.0	1.00956	0.80170	0.00100	0.00003	0.44531
0.00156	0.00428	0.00156	0.00428	1.01410	1.00875	0.00000	0.00004	0.09354
0.00508	0.01295	0.00508	0.01295	1.02345	0.80221	0.00054	0.00005	0.44474
0.01193	0.02586	0.01193	0.02586	1.03807	0.30671	0.00273	0.00006	0.83264
0.02401	0.03599	0.02401	0.03599	1.05383	-0.18838	0.00473	0.00007	1.09013
0.03949	0.04458	0.03949	0.04458	1.07153	-0.39449	0.00408	0.00010	1.18129
0.05752	0.05221	0.05752	0.05221	1.09111	-0.52032	0.00369	0.00012	1.23625
0.07759	0.05877	0.07759	0.05877	1.11223	-0.61729	0.00329	0.00014	1.27173
0.09932	0.06414	0.09932	0.06414	1.13461	-0.66913	0.00280	0.00016	1.29195
0.12252	0.06855	0.12252	0.06855	1.15823	-0.68352	0.00219	0.00018	1.29763
0.14723	0.07160	0.14723	0.07160	1.18312	-0.66176	0.00154	0.00020	1.28909
0.17368	0.07393	0.17368	0.07393	1.20967	-0.61779	0.00094	0.00023	1.27192
0.20277	0.07511	0.20277	0.07511	1.23879	-0.55559	0.00053	0.00026	1.24723
0.23471	0.07585	0.23471	0.07585	1.27073	-0.50521	0.00052	0.00029	1.22687
0.26828	0.07599	0.26828	0.07599	1.30430	-0.46903	0.00660	0.00020	1.21204
0.30344	0.07543	0.30344	0.07543	1.33946	-0.43748	0.00450	0.00035	1.19895
0.33830	0.07442	0.33830	0.07442	1.37434	-0.41636	0.00361	0.00048	1.19011
0.37327	0.07290	0.37327	0.07290	1.40934	-0.40183	0.00355	0.00056	1.18399
0.40844	0.07078	0.40844	0.07078	1.44458	-0.38504	0.00451	0.00061	1.17688
0.44421	0.06771	0.44421	0.06771	1.48048	-0.31409	0.00421	0.00075	1.14634
0.48019	0.06312	0.48019	0.06312	1.51675	-0.20147	0.00371	0.00094	1.09612
0.51713	0.06002	0.51713	0.06002	1.55362	-0.17659	0.00369	0.00103	1.08471
0.55452	0.05829	0.55452	0.05829	1.59129	-0.22773	0.00394	0.00103	1.10803
0.59298	0.05531	0.59298	0.05531	1.62983	-0.24365	0.00396	0.00109	1.11519
0.63237	0.05159	0.63237	0.05159	1.66939	-0.21575	0.00376	0.00121	1.10261
0.67109	0.04698	0.67109	0.04698	1.70838	-0.17205	0.00352	0.00137	1.08261
0.71515	0.04177	0.71515	0.04177	1.75275	-0.13439	0.00333	0.00153	1.06508
0.76466	0.03546	0.76466	0.03546	1.80266	-0.10068	0.00316	0.00181	1.04913
0.81270	0.02909	0.81270	0.02909	1.85112	-0.06397	0.00299	0.00208	1.03149
0.86074	0.02249	0.86074	0.02249	1.89961	-0.02591	0.00280	0.00235	1.01287
0.90751	0.01568	0.90751	0.01568	1.94687	0.02456	0.00256	0.00261	0.98765
0.95354	0.00865	0.95354	0.00865	1.99343	0.09962	0.00225	0.00287	0.94888
0.99999	0.00130	0.99999	0.00130	2.04047	0.14230	0.00205	0.00314	0.92612

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=0
FSMACH = 0.20000

ALPHA	CL	CO	CM(NOSE)	CM(1/4-CHORD)
0.0	0.125380	0.006621	-0.039047	-0.008502

Figure A-3. Continued.

***** CASE INPUT *****

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1

NRU NRL IWRITE IALPHA IPUNCH
10   10   1   0   0

        XU = 0.0   0.125000E-01 0.250000E-01 0.500000E-01 0.750000E-01 0.100000E 00 0.150000E 00 0.200000E 00
              0.250000E 00 0.300000E 00 0.400000E 00 0.500000E 00 0.600000E 00 0.700000E 00 0.800000E 00 0.900000E 00
              0.950000E 00 0.100000E 01
        YU = 0.0   0.267000E-01 0.361000E-01 0.491000E-01 0.580000E-01 0.642990E-01 0.719000E-01 0.750000E-01
              0.760000E-01 0.759000E-01 0.714000E-01 0.614000E-01 0.547000E-01 0.436000E-01 0.308000E-01 0.168000E-01
              0.920000E-02 0.130000E-02
        XL = 0.0   0.125000E-01 0.250000E-01 0.500000E-01 0.750000E-01 0.100000E 00 0.150000E 00 0.200000E 00
              0.250000E 00 0.300000E 00 0.400000E 00 0.500000E 00 0.600000E 00 0.700000E 00 0.800000E 00 0.900000E 00
              0.950000E 00 0.100000E 01
        YL = 0.0   -0.123000E-01-0.171000E-01-0.226000E-01-0.261000E-01-0.292000E-01-0.350000E-01-0.397000E-01
             -0.426000E-01-0.446000E-01-0.448000E-01-0.417000E-01-0.367000E-01-0.300000E-01-0.216000E-01-0.123000E-01
             -0.7000CCOE-02-0.130000E-02

NA = 1   ANGLES OF ATTACK W.R.T. REFERENCE LINE (IALPHA=0) OR W.R.T. LONGEST CHORDLINE (IALPHA=1)

         0.0

NM = 1   FSMACH   = 0.200000E 00

CREF = 0.1000C0E 01   SF = 0.100000E 01   TO = 0.518690E 03   RN = 0.300000E 01   PR = 0.770000E 00   KF = 0.100000E 01

              LTRAN   XTRAN   ZTRAN

UPPER SURFACE 0   0.0   0.0
LOWER SURFACE 0   0.0   0.0

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1
INPUT AIRFOIL POINTS

UPPER SURFACE		LOWER SURFACE	
XU	YU	XL	YL
0.0	0.0	0.0	0.0
0.012500	0.026700	0.012500	-0.012300
0.025000	0.036100	0.025000	-0.017100
0.050000	0.049100	0.050000	-0.022600
0.075000	0.058000	0.075000	-0.026100
0.100000	0.064300	0.100000	-0.029200
0.150000	0.071900	0.150000	-0.035000
0.200000	0.075000	0.200000	-0.039700
0.250000	0.076000	0.250000	-0.042600
0.300000	0.075500	0.300000	-0.044600
0.400000	0.071400	0.400000	-0.044800
0.500000	0.061400	0.500000	-0.041700
0.600000	0.054700	0.600000	-0.036700
0.700000	0.043600	0.700000	-0.030000
0.800000	0.030800	0.800000	-0.021600
0.900000	0.016800	0.900000	-0.012300
0.950000	0.009200	0.950000	-0.007000
1.000000	0.001300	1.000000	-0.001300

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1
DISTRIBUTED AIRFOIL POINTS

UPPER SURFACE		LOWER SURFACE	
XU	YU	XL	YL
0.001562	0.006203	0.999994	-0.001301
0.005084	0.012948	0.993537	-0.006610
0.011934	0.025065	0.907505	-0.011530
0.024014	0.035907	0.860744	-0.016157
0.039488	0.044576	0.812696	-0.020464
0.057924	0.052210	0.764660	-0.024671
0.077592	0.059774	0.715153	-0.028836
0.099321	0.064163	0.671088	-0.032112
0.122924	0.068549	0.632369	-0.034717
0.147233	0.071597	0.592982	-0.037106
0.173675	0.073929	0.554524	-0.039185
0.202774	0.075110	0.517124	-0.040978
0.234705	0.075853	0.480190	-0.042465
0.266279	0.075491	0.444215	-0.043836
0.303436	0.075427	0.408438	-0.044686
0.338303	0.074418	0.373269	-0.045190
0.373269	0.072901	0.338303	-0.045712
0.408438	0.070784	0.303436	-0.044682
0.444215	0.067704	0.268279	-0.043609
0.480190	0.063119	0.234705	-0.041490
0.517124	0.060018	0.202774	-0.039914
0.554524	0.058292	0.173675	-0.037363
0.592982	0.055314	0.147233	-0.034708
0.632369	0.051589	0.122924	-0.031879
0.671088	0.046584	0.099321	-0.029121
0.715153	0.041770	0.077592	-0.026440
0.764660	0.035461	0.057924	-0.023864
0.812696	0.029089	0.039488	-0.020953
0.860744	0.022487	0.024014	-0.016994
0.907905	0.015678	0.011934	-0.011905
0.953537	0.008651	0.005084	-0.009908
0.999994	0.001301	0.001562	-0.001948
		0.0	0.0

LAMINAR BOUNDARY LAYER SUMMARY

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1

UPPER SURFACE
ITERATION NUMBER 4

STAGNATION TEMPERATURE   = 518.64   DEGREES RANKINE   STAGNATION PRESSURE   = 4478.688   LB/SQ FT
FREESTREAM MACH NUMBER   = 0.20000   AIRFOIL CHORD   = 1.00000   FT
REYNOLDS NUMBER   = 3.00000 MILLION   PRANDTL NUMBER   = 0.77000
HEAT TRANSFER FACTOR K   = 1.00000   SEPARATION CORRELATION NO.   = 0.069637
SPEED OF SOUND   = 1116.419 FEET/SECOND   KINEMATIC VISCOSITY   = 0.0000743 SQ FT/SEC
ANGLE OF ATTACK (ALPHA) = 0.0   DEGREES
STAGNATION AT X/C   = 0.0013530   STAGNATION AT X/C   = 0.0037539

X/C	S/C	N	DH/D(S/C)	H	THETA/C	DELS/C	DELTAS/C	CF	CP
0.001562	0.000978	0.806945	11.968713	2.500720	0.000017	0.000043		-0.000002	1.00875
0.005084	0.008923	0.898079	7.498247	2.506425	0.000022	0.000055		0.000543	0.80221
0.011934	0.024544	0.116428	4.290575	2.496498	0.000026	0.000064		0.007726	0.30671
0.024014	0.040303	0.210224	2.232097	2.480362	0.000029	0.000073		0.004725	-0.18638
0.039488	0.058001	0.216769	0.622852	2.464285	0.000039	0.000096		0.004079	-0.39545
0.057924	0.077504	0.248006	0.463543	2.461936	0.000047	0.000117		0.003690	-0.52032
0.077592	0.098699	0.255203	0.267674	2.457661	0.000056	0.000137		0.003289	-0.61729
0.099321	0.121085	0.259439	0.110551	2.448127	0.000064	0.000157		0.002800	-0.68913
0.122924	0.144698	0.260567	-0.009607	2.431506	0.000073	0.000178		0.002391	-0.68352
0.147233	0.169994	0.258053	-0.099884	2.409523	0.000086	0.000202		0.001541	-0.66178
0.173675	0.196130	0.255324	-0.192524	2.385482	0.000096	0.000229		0.000937	-0.61779
0.202774	0.225263	0.250297	-0.153313	2.368759	0.000110	0.000260		0.000531	-0.55959
0.234705	0.257200	0.246084	-0.111018	2.372191	0.000124	0.000293		0.000523	-0.50921
0.266279	0.296773	0.243049	-0.083411	2.374044	0.000136	0.000323		0.000517	-0.46903

SCRIT/C = 0.11020900   RECRIT = 456.30   STRAN/C = 0.28427039   RTRAN = 919.04   THETA1/C = 0.00013387

INSTABILITY TRANSITION HAS OCCURRED AT X/C = 0.261784   S/C = 0.284278

Figure A-4. Sample output of the 2-D characteristics program with IWRITE=1.

TURBULENT BOUNDARY LAYER SUMMARY FOR EQUIVALENT AIRFOIL   (GORADIA METHOD)

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1

UPPER SURFACE
ITERATION NUMBER 4

STAGNATION TEMPERATURE	=	510.69	DEGREES RANKINE	STAGNATION PRESSURE	=	4470.688	LB/SQ FT
PREESTREAM MACH NUMBER	=	0.20000		AIRFOIL CHORD	=	1.00000	FT
REYNOLDS NUMBER	=	3.00000	MILLION	PRANDTL NUMBER	=	0.77000	
HEAT TRANSFER FACTOR K	=	1.00000		TRANSITION POINT, S/C	=	0.20420	
(INITIAL MOMENTUM THICKNESS)/C	=	0.00013		INIT. INCOMPR. FORM FACTOR	=	1.35291	
ANGLE OF ATTACK (ALPHA)	=	0.0	DEGREES				

X/C	S/C	R	DR/(S/C)	H	THETA/C	DELS/C	DELTAS/C	CP	CP

LAMINAR BOUNDARY LAYER SUMMARY

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1

LOWER SURFACE
ITERATION NUMBER 4

STAGNATION TEMPERATURE	=	510.69	DEGREES RANKINE	STAGNATION PRESSURE	=	4470.688	LB/SQ FT
PREESTREAM MACH NUMBER	=	0.20000		AIRFOIL CHORD	=	1.00000	FT
REYNOLDS NUMBER	=	3.00000	MILLION	PRANDTL NUMBER	=	0.77000	
HEAT TRANSFER FACTOR K	=	1.00000		SEPARATION CORRELATION NO.	=	0.049637	
SPEED OF SOUND	=	1116.419	FEET/SECOND	KINEMATIC VISCOSITY	=	0.0000743	SQ FT/SEC
ANGLE OF ATTACK (ALPHA)	=	0.0	DEGREES				
STAGNATION AT X/C	=	0.0013530		STAGNATION AT Z/C	=	0.0037539	

X/C	S/C	R	DR/(DIS/C)	H	THETA/C	DELS/C	DELTAS/C	CP	CP

SCRIT/C = 0.04167026   RECRIT = 272.44   STRAN/C = 0.39547205   ATRAN = 1027.64   THETA1/C = 0.00016254

INSTABILITY TRANSITION HAS OCCURRED AT   X/C = 0.383732   S/C = 0.395472

TURBULENT BOUNDARY LAYER SUMMARY FOR EQUIVALENT AIRFOIL   (GORADIA METHOD)

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1

LOWER SURFACE
ITERATION NUMBER 4

STAGNATION TEMPERATURE	=	510.69	DEGREES RANKINE	STAGNATION PRESSURE	=	4470.688	LB/SQ FT
PREESTREAM MACH NUMBER	=	0.20000		AIRFOIL CHORD	=	1.00000	FT
REYNOLDS NUMBER	=	3.00000	MILLION	PRANDTL NUMBER	=	0.77000	
HEAT TRANSFER FACTOR K	=	1.00000		TRANSITION POINT, S/C	=	0.39547	
(INITIAL MOMENTUM THICKNESS)/C	=	0.00016		INIT. INCOMPR. FORM FACTOR	=	1.36016	
ANGLE OF ATTACK (ALPHA)	=	0.0	DEGREES				

X/C	S/C	R	DR/(S/C)	H	THETA/C	DELS/C	DELTAS/C	CP	CP

Figure A-4.   Continued.

296

LOAD SUMMARY SHEET--ITERATION NUMBER 4

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1

MACH NUMBER   = 0.20000   REFERENCE CHORD (BASIS FOR ALL COEFFICIENT NORMALIZATION) = 1.00000 FT.

REYNOLDS NO.   = 3.00000 MILLION ANGLE OF ATTACK (VALUE OF IALPHA = 0)   = 0.0   DEG.
                                         (IALPHA=0--ANGLE DEFINED W.R.T. REFERENCE LINE)
LONGEST CHORDLINE = 1.00000 FEET   (IALPHA=1--ANGLE DEFINED W.R.T. LONGEST CHORDLINE)

ANGLE BETWEEN LONGEST CHORDLINE AND REFERENCE LINE = 0.0   DEG. (POSITIVE FOR REFERENCE LINE BELOW THE LONGEST CHORDLINE)

```
NORMAL FORCE COEFFICIENTS                    LIFT COEFFICIENTS
        (CN)PRESS. =  0.13541                        (CL)PRESS. =  0.13541
        (CN)SHEAR  = -0.00003                        (CL)SHEAR  = -0.00003
        CN         =  0.13538                        CL         =  0.13538

AXIAL FORCE COEFFICIENTS                     DRAG COEFFICIENTS
        (CA)PRESS. =  0.00085                        (CD)PRESS. =  0.00085
        (CA)SHEAR  =  0.00577                        (CD)SHEAR  =  0.00577
        CA         =  0.00662                        CD         =  0.00662

MOMENT COEFFICIENTS                          MOMENT COEFFICIENTS
  (ABOUT NOSE)  (CM)PRESS. = -0.04000                (CM)ABOUT NOSE   = -0.03985
                (CM)SHEAR  =  0.00015                (CM)QUARTER CHORD = -0.00050
                CM         = -0.03985
```

DRAG COEFFICIENT COMPUTED BY SQUIRE-YOUNG FORMULA = 0.00643

ZEROTH ITERATION PRESSURE DRAG = 0.00194

TRANSITION POINTS: X/C(LOWER) = 0.30379   X/C(UPPER) = 0.26170

LONGEST CHORDLINE SYSTEM		BASIC REFERENCE SYSTEM		S/C	CP	CF	DELTAS/C	V/VD
X/C	Z/C	X/C	Z/C					

NACA 23012 AIRFOIL/ALPHA=0/RN=3.0/MACH NO.=0.2/IWRITE=1
PSMACH = 0.20000

ALPHA	CL	CD	CM(NOSE)	CM(1/4-CHORD)
0.0	0.125380	0.006621	-0.039047	-0.000502

Figure A-4. Continued.

297

Figure A-5. Sample output of the 2-D characteristics program with IWRITE=2.

NACA 23012 AIRFOIL/ALPHA=0,2,4,6,8,10,12,14/RN=3.0/MACH NO.=0.2/IWRITE=3

```
NXU  NXL  IWRITE  IALPHA  IPUNCH
18   18     3       0       0

XU =  0.0          0.125CCOE-01  C.250000E-01  0.500000E-01  0.750000E-01  0.100000E 00  0.150000E 00  0.200000E 00
      0.25CCCCOE CO  0.30CCCCOE 00  C.4C0000E 00  0.500000E 00  0.600000E 00  0.700000E 00  0.800000E 00  0.900000E 00
      0.95CCCOE CO  0.1CCCCOE 01

ZU =  0.0          0.267000E-01  0.361000E-01  0.491000E-01  0.580C00E-01  0.642999E-01  0.719000E-01  0.750000E-01
      0.760CCOE-C1  0.755C00E-01  C.714000E-01  0.614000E-01  0.547000E-01  0.436000E-01  0.308000E-01  0.168000E-01
      0.92CCOE-C2  0.130CCOE-02

XL =  0.C.         0.125C00E-01  C.250000E-01  0.500000E-01  0.750000E-01  0.100000E 00  0.150000E 00  0.200000E 00
      0.250000E CO  0.300000E C0  0.400000E C0  0.500000E 00  0.600000E 00  0.700000E 00  0.800000E 00  0.900000E 00
      0.95CCCOE CO  0.1CCCCOE 01

ZL =  0.0          -0.123000E-01  -0.171000E-01  -0.226000E-01  -0.261000E-01  -0.292000E-01  -0.350000E-01  -0.397000E-01
      -0.428CCOE-01  -0.446C00E-01  -0.448000E-01  -0.417000E-01  -0.300000E-01  -0.216000E-01  -0.123000E-01
      -0.70CCCOE-02  -0.130CCOE-02
```

NA = 8   ANGLES OF ATTACK W.R.T. REFERENCE LINE (IALPHA=0) OR W.R.T. LONGEST CHORDLINE (IALPHA=1)

```
C.0          0.2CCCCOE 01   0.4CCCCOE 01   0.600000E 01   0.800000E 01   0.1000000E 02
0.120000E 02   0.14CCCOE 02
```

NM = 1   FSMACH   = 0.2CCCCCE 00

CREF = 0.1000COE 01   SF = 0.100CCOE C1   TO = C.518690E 03   RN = 0.300000E 01   PR = 0.770000E 00   KF = 0.100000E 01

```
                LTRAN   XTRAN   ZTRAN

UPPER SURFACE     0      0.C     0.0
LOWER SURFACE     0      0.C     0.0
```

NACA 23012 AIRFOIL/ALPHA=0,2,4,6,8,10,12,14/RN=3.0/MACH NO.=0.2/IWRITE=3
FSMACH = C.2COCO

ALPHA	CL	CD	CM(NOSE)	CM(1/4-CHORD)
0.0	0.125380	C.006621	-0.039847	-0.008502
2.C00000	0.348239	0.006935	-0.097525	-0.010458
4.C00000	0.570817	0.007488	-0.154459	-0.011972
6.C0000C	0.788949	C.008154	-0.209180	-0.012810
8.C0000C	1.004408	0.010101	-0.263183	-0.014173
10.C0000C	1.178342	0.014641	-0.302996	-0.012251
12.C0000C	1.354321	C.018874	-0.342340	-0.010178
14.C00000	1.562696	C.025908	-0.397864	-0.017228

Figure A-6. Sample output of the 2-D characteristics program with IWRITE=3.

299

APPENDIX B - Polynomial Fit of Two Dimensional Data

User Instructions

This program is written in Fortran IV and is designed to run in single precision on an IBM 370-165 computer. Execution requires 54,000 bytes of core storage and approximately six seconds to fit one set of two-dimensional airfoil data giving four polynomial curve fits of degree four. For each airfoil the program requires the following input data:

(1) The 80 characters of the array TITLE which are used as a header for identifying output. Since the program allows more than one airfoil to be analyzed in a given run, TITLE is used as a control variable to end execution. Termination of execution is achieved by following the last set of airfoil data to be analyzed by a title card having only the word END in the first three spaces (see the last card of the sample data set in Figure B-2).

(2) The number NUM. The largest allowable value of NUM is 20. This variable specifies the number of angles of attack which follow.

(3) The first element values of the arrays AL, CL, CD, CM. These are the angle of attack and the two-dimensional coefficients of lift, drag, and pitching moment for that angle of attack. Similar cards with successive array elements follow until the number of points specified by NUM are read in.

In addition to the previous input data specification an internal switch is provided to suppress the plot of the input points and the curve fit function. When SWITCH is set to zero, plots are produced; however, when SWITCH is set to one, the plots are suppressed. Another internal switch PUNCH which operates similarly allows for the fitted coefficients to be punched into card form for use with other programs.

Statements (1) through (3) represent a complete data set for a particular airfoil. The format specification for this data is given in Figure B-1. A sample data set of the 23012 airfoil is shown in Figure B-2. The output of this particular data set is shown in Figure B-3.

300

Figure B-1. Format specification of input data for the polynomial fit of 2-D data program.

TITLE

END

AL(NUM)   CL(NUM)   CD(NUM)   CM(NUM)
12.000   1.3580   0.18434E-01   -0.10255E-01

10.000   1.1850   0.14048E-01   -0.12825E-01

8.0000   1.0056   0.89529E-02   -0.14259E-01

6.0000   0.79169   0.80974E-02   -0.13172E-01

4.0000   0.57321   0.70861E-02   -0.12139E-01

2.0000   0.34962   0.68436E-02   -0.10349E-01

etc.
0.0   0.12565   0.64448E-02   -0.65043E-02

AL(2)   CL(2)   CD(2)   CM(2)
-2.0000   -0.9499E-01   0.75489E-02   -0.66656E-02

AL(1)   CL(1)   CD(1)   CM(1)
-4.0000   -0.31963   0.78733E-02   -0.60069E-02

NUM
9

TITLE

23012 SERIES AIRFOIL / ALPHA=-4,-2,0,2,4,6,8,10,12 / RN=3.49 / MACH NO =.2

Figure B-2.   Example data set for the polynomial fit of 2-D data program.

302

```
      INTEGER*4 SWITCH,PUNCH
      DIMENSION AL(20),CL(20),CD(20),CM(20),CCL(22),CCD(22),CCM(22),CAL(
     122),TITLE(20),TITLE1(4),TITLE2(4),TITLE3(4),TITLE4(4)
      COMMON /JREAD/JREAD,JPUNCH,JWRITE
      DATA END/4HEND /,TITLE1/4HCL V,4HERSU,4HS AL,4HPHA /,TITLE2/4HCD V
     1,4HERSU,4HS CL,4H    /,TITLE3/4HCM V,4HERSU,4HS CL,4H    /,TITLE4/
     14HALPH,4HA VE,4HRSUS,4H CL /
C
C     SWITCH...AN INTEGER WHICH CAN TAKE ON THE FOLLOWING VALUES
C        SWITCH=0 INDICATES THAT A PLOT IS DESIRED
C        SWITCH=1 INDICATES THAT THE PLOT IS NOT DESIRED
C     PUNCH...AN INTEGER WHICH CAN TAKE ON THE FOLLOWING VALUES
C        PUNCH=0 INDICATES THAT A PUNCHED DECK IS DESIRED
C        PUNCH=1 INDICATES THAT A PUNCHED DECK IS NOT DESIRED
C
      PUNCH=0
      SWITCH=0
C
C     SET INPUT-OUTPUT UNIT NUMBERS
C
      JREAD=1
      JPUNCH=2
      JWRITE=3
C
C     SET THE DEGREE FOR POLYNOMIAL CURVE FIT
C
      N=4
      NP1=N+1
C
C     INPUT AIRFOIL IDENTIFICATION CARD AND THE NUMBER OF DATA PTS.
C
    5 READ (JREAD,20) TITLE,NUM
      IF (TITLE(1).EQ.END) GO TO 125
      WRITE (JWRITE,30)
      WRITE (JWRITE,45)
      WRITE (JWRITE,55)
      WRITE (JWRITE,50)
      WRITE (JWRITE,60) TITLE,NUM
      WRITE (JWRITE,50)
      IF (NUM.GT.NP1) GO TO 10
      WRITE (JWRITE,35)
      GO TO 5
   10 WRITE (JWRITE,40)
C
C     INPUT DATA POINTS
C
      DO 15 I=1,NUM
      READ (JREAD,65) AL(I),CL(I),CD(I),CM(I)
   15 WRITE (JWRITE,70) AL(I),CL(I),CD(I),CM(I)
      WRITE (JWRITE,50)
      WRITE (JWRITE,45)
C
C     PERFORM LEAST SQUARES CURVE FIT
C
      CALL LSQFIT(AL,CL,NUM,CCL,N,SWITCH,TITLE1)
      CALL LSQFIT(CL,CD,NUM,CCD,N,SWITCH,TITLE2)
      CALL LSQFIT(CL,CM,NUM,CCM,N,SWITCH,TITLE3)
      CALL LSQFIT(CL,AL,NUM,CAL,N,SWITCH,TITLE4)
      AL1=AL(1)
      ALM=AL(NUM)
      CL1=CL(1)
      CLM=CL(NUM)
```

```
C
C     OUTPUT IDENTIFICATION CARD AND POLYNOMIAL COEFFICIENTS
C
      WRITE (JWRITE,30)
      WRITE (JWRITE,85)
      WRITE (JWRITE,80)
      WRITE (JWRITE,75)
      WRITE (JWRITE,115)
      WRITE (JWRITE,90)
      WRITE (JWRITE,110) TITLE
      WRITE (JWRITE,90)
      WRITE (JWRITE,120)
      WRITE (JWRITE,95)  (CCL(I),I=1,NP1),AL1,ALM
      WRITE (JWRITE,100) (CCD(I),I=1,NP1),CL1,CLM
      WRITE (JWRITE,105) (CCM(I),I=1,NP1),CL1,CLM
      WRITE (JWRITE,65)  (CAL(I),I=1,NP1),CL1,CLM
      WRITE (JWRITE,85)
      IF (PUNCH.EQ.1) GO TO 5
      WRITE (JPUNCH,25) TITLE
      WRITE (JPUNCH,65) (CCL(I),I=1,NP1),AL1,ALM
      WRITE (JPUNCH,65) (CCD(I),I=1,NP1),CL1,CLM
      WRITE (JPUNCH,65) (CCM(I),I=1,NP1),CL1,CLM
      WRITE (JPUNCH,65) (CAL(I),I=1,NP1),CL1,CLM
      GO TO 5
   20 FORMAT (20A4,-/I10)
   25 FORMAT (20A4)
   30 FORMAT (1H1)
   35 FORMAT (//1X,7HTHE NUMBER OF POINTS READ IN MUST BE AT LEAST TWO
     1GREATER THAN THE VARIABLE - N)
   40 FORMAT (20X,1H*,26X,5HALPHA,7X,2HCL,7X,2HCD,7X,2HCM,4X,26X,1H*)
   45 FORMAT (20X,84(1H*))
   50 FORMAT (20X,1H*,84X,1H*)
   55 FORMAT (20X,1H*,25X,34HTWO DIMENSIONAL AIRFOIL DATA INPUT,25X,1H*)
   60 FORMAT (20X,1H*,2X,20A4,2X,1H*/20X,1H*,25X,30HTHE NUMBER OF DATA P
     1OINTS IS =,I4,25X,1H*)
   65 FORMAT (5G14.5)
   70 FORMAT (20X,1H*,22X,4F10.6,22X,1H*)
   75 FORMAT (3H 0,40X,'TWO DIMENSIONAL CURVE FIT FUNCTION DATA',35X,1H
     1*)
   80 FORMAT (2X,1H*,114X,1H*)
   85 FORMAT (2X,116(1H*))
   90 FORMAT (2X,1H*      ,15HCL VERSUS ALPHA,5F10.5,13X,7HDOMAIN=,F10.4,5H
     1 TO ,F10.4,3H  *)
   95 FORMAT (2X,1H*      ,15HCD VERSUS CL   ,5F10.5,13X,7HDOMAIN=,F10.4,5H
     1 TO ,F10.4,3H  *)
  100 FORMAT (2X,1H*      ,15HCM VERSUS CL   ,5F10.5,13X,7HDOMAIN=,F10.4,5H
     1 TO ,F10.4,3H  *)
  105 FORMAT (2X,1H*      ,15HALPHA VERSUS CL,5F10.5,13X,7HDOMAIN=,F10.4,5H
     1 TO ,F10.4,3H  *)
  110 FORMAT (2X,1H*,10X,20A4,16X,1H*)
  115 FORMAT (3H 0,40X,39HOF THE FORM Y=C(0)+C(1)*X+C(2)*X**2+....35X,1
     1H*)
  120 FORMAT (2X,1H*,22X,4HC(0),6X,4HC(1),6X,4HC(2),6X,4HC(3),6X,4HC(4),
     146X,1H*)
  125 RETURN
      END
C
      SUBROUTINE LSQFIT(X,Y,N,C,M,SWITCH,TITLE)
```

303

```
C   THIS SUBROUTINE SETS UP DATA FOR THE LEAST SQUARES CURVE FIT
C   WRITTEN BY KEVIN R. JONES AND AVAILABLE FROM NCSU COMP. CENTER

      REAL*8PCLY
      INTEGER*2 I8/2H  /,SWITCH*4
      COMMON /INOUT/JREAD,JPUNCH,JWRITE
      DIMENSION YY(500),C(IN),X(IN),Y(IN),TITLE(4)
      DIMENSION X(IFLIP)3
      COMMON /DATA8/V(2,500),H(2,500),NN(2)
      XLOW=X(1)
      XHIGH=X(1)
      YLOW=Y(1)
      YHIGH=Y(1)
      DO 5 I=2,M
      IF (X(I).LT.XLOW)XLOW=X(I)
      IF (Y(I).LT.YLOW)YLOW=Y(I)
      IF (X(I).GT.XHIGH)XHIGH=X(I)
      IF (Y(I).GT.YHIGH)YHIGH=Y(I)
    5 IF (Y(I).GT.YHIGH)YHIGH=Y(I)
      DO 10 KK=1,3
   10 IF(IFLIP(KK))=0
      IF (SWITCH.EQ.1) GO TO 30
      NN(1)=13
      IF (XHIGH.EQ.XLOW).OR.(YHIGH.EQ.YLOW)) GO TO 25
      IF (XHIGH.GT.XLOW) GO TO 15
      XXX=XHIGH
      XHIGH=XLOW
      XLOW=XXX
   15 IF (YHIGH.GT.YLOW) GO TO 20
      XXX=YHIGH
      YHIGH=YLOW
      YLOW=XXX
   20 XXX=120./(ABS(XHIGH-XLOW)
      DXX=1./XXX
      YYY=60./(ABS(YHIGH-YLOW)
      XSCALE=DXX*20.
      YINCH=6./YYY
      XINCH=DXX*10.
      GC TO 30
   25 SWITCH=1
      IFLIP(1)=1
      NN(2)=1
   30 IF (M.EQ.500)IFLIP(2)=1
      DO 35 J=1,M
      V(2,J)=XXX*(X(J)-XLOW)
      H(2,J)=YYY*(Y(J)-YLOW)
   35 WRITE (JWRITE,100) TITLE
      IF (IFLIP(1).EQ.1) WRITE (JWRITE,70)
      IF (IFLIP(2).EQ.1) WRITE (JWRITE,115)
      IF (IFLIP(3).EQ.1) WRITE (JWRITE,75)
      CALL LSQ(X,Y,M,C,N)
      SUM=0.0
      DO 45 J=1,M
      YY(J)=POLY(X(J),N,C,M)
      IF (Y(J).NE.0.)SUM=SUM+((YY(J)-Y(J))**2/Y(J))**2
   45 CONTINUE
      SUM=SUM/(M-1)
      WRITE (JWRITE,110) SUM
      NP1=N+1
      WRITE (JWRITE,80)
      DO 50 K=1,NP1
      KM1=K-1
   50 WRITE (JWRITE,85) (8,KM1,C(K)
      WRITE (JWRITE,120)
      M1=M/2
      M2=N/2+1
      DO 55 I=1,M1
      M2M2=1
   55 WRITE (JWRITE,95) X(I),Y(I),YY(I),X(M2),YY(M2),YY(M2)
      IF ((2*M/2)-M1.NE.0) WRITE (JWRITE,95) X(M),YY(M),YY(M)
      IF (SWITCH.EQ.1) WRITE (JWRITE,135)
      IF (SWITCH) 140,60,140
   60 T=XLOW
      WRITE (JWRITE,125)
      DO 65 J=1,13
      M(1,J)=YYY*(POLY(T,M,C,N)-YLOW)
      V(1,J)=XXX*(T-XLOW)
   65 T=T+DXX
      WRITE (JWRITE,130) TITLE,YHIGH,XLOW,XHIGH
      CALL GPLOT
   70 FORMAT (//1X,68HINVALID PLOT RANGE HAS BEEN SPECIFIED...THE PLOT H
     1AS BEEN SUPPRESSED)
   75 FORMAT (//1X,72HDEGREE OF POLYNOMIAL HAS BEEN REDUCED TO AGREE WIT
     1H THE NUMBER OF POINTS.//)
   80 FORMAT (//1X,T48,24HTHE FIT COEFFICIENTS ARE//)
   85 FORMAT (A2,2HC(,I2,2H)=,G13.6)
   90 FORMAT (///1X,8HDEGREE= ,I2//1X,23HNUMBER OF DATA POINTS= ,I3)
   95 FORMAT (3(4X,G14.7),8X,2HI),2X,3(4X,G14.7))
  100 FORMAT (1H1,T49,4A4)
  105 FORMAT (1X,2HY=,G13.6,6H,    X=,G13.6,T118,2HX=,G13.6)
  110 FORMAT (11HOVARIANCE= ,G12.6)
  115 FORMAT (//30H TOO MANY DATA POINTS SUPPLIED,5X,34HONLY THE FIRST 5
     100 POINTS ACCEPTED)
  120 FORMAT (/1HO,6X,7HX-VALUE,11X,7HY-VALUE,11X,5HY-FIT,26X,7HX-VALUE,
     111X,7HY-VALUE,10X,5HY-FIT/)
  125 FORMAT (///1X,13HPLOT OF Y..(X,1H*,18HS)    AND   Y-FIT..(8,1H*,11HS)
     1 FOLLOWS:)
  130 FORMAT (1H1,T49,4A4//1X,2HY=,G13.6,10X,23HGRID SQUARE DIMENSIONS:,
     12X,2HX=,G13.6,T75,6HYSCALE=,T85,2HX=,G13.6,11H UNITS/INCH/1X,T51,2H
     1Y=,G13.6,T85,2HY=,G13.6,11H UNITS/INCH)
  135 FORMAT (///50H THE PLOT HAS BEEN SUPPRESSED DUE TO USER REQUEST.)
  140 RETURN
      END
```

```
C   THIS SUBROUTINE PERFORMS A LEAST SQUARES CURVE FIT ON M POINTS
C   CONTAINED IN X VERSUS Y INTO A POLYNOMIAL OF DEGREE N AND
C   COEFFICIENTS C.

      SUBROUTINE LSQ(X,Y,M,C,N)
      REAL*8A(21,22),B(500),YSUM,SUM
      DIMENSION C(IN),X(IN),Y(IN)
      COMMON /INOUT/JREAD,JPUNCH,JWRITE
      IT=2
      N=N+1
      M=M
      NP1=N+1
      NM1=N-1
      N2N=2N-1
      DO 5 J=1,M
    5 B(J)=1.0
      L1=1
      DO 25 L=1,N2N
```

```
      POLY=C(NP1)
      DO 5 J=1,N
      K=NP1-J
    5 POLY=POLY*T+C(K)
      RETURN
      END
C
      SUBROUTINE GPLOT
C
C     THIS SUBROUTINE PLOTS DATA ON THE PRINTER
C
      COMMON /INOUT/JREAD,JPUNCH,JWRITE
      COMMON /DATAS/V(2,500),W(2,500),NN(2)
      DIMENSION IPRINT(131),ICHAR(6)
      DATA ICHAR/4HP  ,4HX  ,4H   ,4HI  ,4HI  ,4H+  /
      LCT=1
      DO 40 KJ=1,61
      DO 5 K=1,131
    5 IPRINT(K)=ICHAR(3)
      DO 10 K=1,121,20
   10 IPRINT(K)=ICHAR(5) GO TO 25
      IF (KJ.NE.LCT) GO TO 25
      LCT=LCT+10
      DO 15 K=1,131
   15 IPRINT(K)=ICHAR(4)
      DO 20 K=1,121,20
   20 IPRINT(K)=ICHAR(6)
   25 FL=60.5-KJ
      FU=61.5-KJ
      J=1
   30 M=NN(J)
      DO 35 K=1,M
      JX=V(1,K)+1.49999
      IF (W(J,K).GE.FU.OR.W(J,K).LT.FL.OR.JX.LT.1.OR.JX.GT.131) GO TO 35
      IPRINT(JX)=ICHAR(J)
   35 CONTINUE
      J=J+1
      IF (J.LE.2) GO TO 30
   40 WRITE (JWRITE,45) IPRINT
   45 FORMAT (1X,131A1)
      RETURN
      END
```

```
      YSUM=0.00
      SUM=0.00
      DO 10 J=1,N
      YSUM=YSUM+B(J)*V(J)
      SUM=SUM+B(J)
   10 B(J)=B(J)*X(J)
      L2=L
      IF (L.LE.N) GO TO 15
      L1=L1+1
      L2=N
   15 GO TO 20
      A(L,NP1)=YSUM
   20 DO 25 LL=L1,L2
      L3=L-LL+L2
   25 A(LL,L3)=SUM
      KK=0
      MN=0
   30 MN=MN+1
      MNN=MN+1
      MNN=MN+1
      IF (A(MN,MN).EQ.0.) GO TO 70
      DO 40 I=MN,N
      IF (A(I,MN).EQ.0.) GO TO 40
      DO 35 J=MN,NP1
      A(I,J)=A(I,J)/A(I,MN)
      CALL OVERFL(IT)
      IF (IT.EQ.1.OR.IT.EQ.3) GO TO 70
   35 CONTINUE
   40 CONTINUE
      KK=KK+1
      IF (KK.GT.NN1) GO TO 55
      DO 50 I=MNN,N
      IF (A(I,MN).EQ.0.) GO TO 50
      DO 45 J=MNN,NP1
   45 A(I,J)=A(I,J)-A(MN,J)
   50 CONTINUE
      GO TO 30
   55 C(NP1)=1.0
      DO 65 K=1,NP
      SUM=0.00
      J=NP1-K
      L=J+1
      DO 60 I=L,NP1
   60 SUM=SUM-A(J,I)*C(I)
   65 C(J)=SUM
      N=N-1
      RETURN
   70 WRITE (JWRITE,75)
      N=N-1
   75 FORMAT (1X,36HNO FIT IS POSSIBLE WITH THIS PROGRAM/1X,18HPROBLEM T
     1ERMINATED)
      RETURN
      END
```

```
      REAL FUNCTION POLY *8(T,M,C,N)
C
C     THIS SUBROUTINE EVALUATES A POLYNOMIAL OF DEGREE N WITH
C     COEFFICIENTS C AT POINT T.
C
      DIMENSION C(M)
      NP1=N+1
```

305

Sample Output

Figure B-3. Sample output of the polynomial fit of 2-D data program.

CD VERSUS CL

DEGREE= 4

NUMBER OF DATA POINTS= 5

VARIANCE= 0.107926E-02

THE FIT COEFFICIENTS ARE

C(0)= 0.7C7623E-02
C(1)=-0.201755E-02
C(2)= 0.195355E-02
C(3)= 0.573362E-03
C(4)= 0.269209E-02

X-VALUE	Y-VALUE	Y-FIT			X-VALUE	Y-VALUE	Y-FIT
-0.3196300	0.7873300E-02	0.7930178E-02	I I		0.5732100	0.7086098E-02	0.6960C 08E-02
-0.9459300E-01	0.7544899E-02	0.7285273E-02	I I		0.7916400	0.8097399E-02	0.8045' 26E-02
0.1256500	0.6444BC1E-02	0.6855320E-02	I I		1.005600	0.9952899E-02	0.1035 b17E-01
0.3496200	0.6843600F-02	0.6674223E-02	I I		1.185000	0.1404600E-01	0.1369 056E-01
1.358000	0.1843400E-01	0.1852990E-01	I I				

PLOT OF Y..(X'S) AND Y-FIT..(#'S) FOLLOWS:

CD VERSUS CL

Y= 0.184340E-01   GRID SQUARE DIMENSIONS: X= 0.274605   SCALE: X= 0.139802   UNIT: ./INC H
                                               Y= 0.199820E-02   Y= 0.119892E-02 UNIT: ./INC H

Y= 0.644480E-02,   X=-0.319630   X= 1.35800
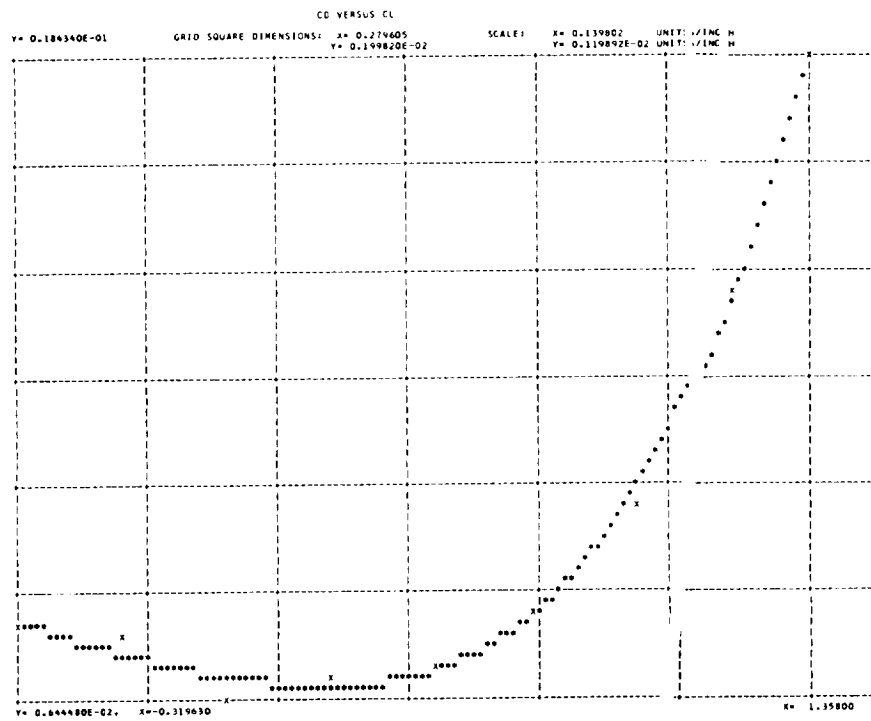
Figure B-3. Continued.

CM VERSUS CL

DEGREE= 4

NUMBER OF DATA POINTS= 9

VARIANCE= 0.27601 75E-03

THE FIT COEFFICIENTS ARE

C(0)=-0.751338E- 22
C(1)=-0.764033E-(02
C(2)=-0.954403E-C 3
C(3)=-0.2050118-0 2
C(4)= 0.424931E-0. 1

X-VALUE	Y-VALUE	Y-FIT		X-VALUE	Y-VALUE	Y-FIT
-0.319630	-0.5086758 E-02	-0.5041111E-02	II	0.5792160	-0.1213900E-01	-0.1215819E-01
-0.9449500E-01	-0.6665800 E-02	-0.6789356E-02	II	0.7916900	-0.1317200E-01	-0.1355218E-01
0.1256900	-0.8304301 E-02	-0.8497734E-02	II	1.005000	-0.1625900E-01	-0.1391044E-01
0.3496200	-0.1054300 E-01	-0.1034224E-01	II	1.185000	-0.1282500E-01	-0.1292013E-01
1.350000	-0.1025500 E-01	-0.1026347E-01	II			

PLOT OF Y..(X'S) AND Y-FIT..(''S) FOLLOWS:
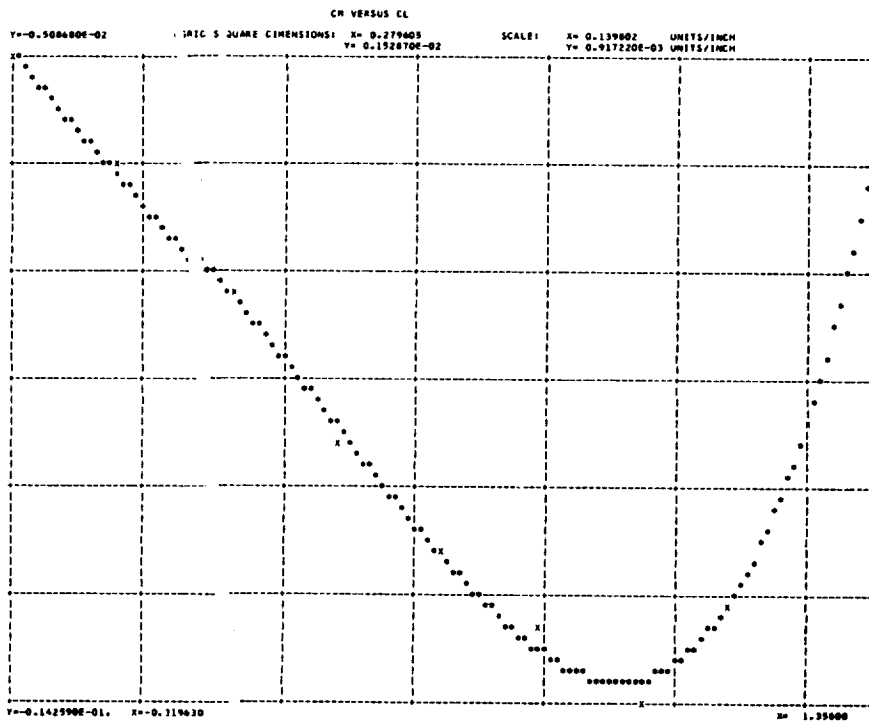
CM VERSUS CL

Figure B-3. Continued.

ALPHA VERSUS CL

DEGREE= 4

NUMBER OF DATA POINTS= 9

VARIANCE= 0.105174E-04

THE FIT COEFFICIENTS ARE

C(0)= -1.13585
C(1)= 9.02351
C(2)=-0.558624E-01
C(3)=-0.497145
C(4)= 0.659838

X-VALUE	Y-VALUE	Y-FIT		X-VALUE	Y-VALUE	Y-FIT
-0.3194300	-4.000000	-4.002621		0.5732100	4.000000	3.995767
-0.9499300E-01	-2.000000	-1.993043		0.7916900	6.000000	5.985488
0.1256500	0.0000000	-0.3748405E-02		1.005600	8.000000	8.050905
0.3496200	2.000000	2.000735		1.185000	10.00000	9.952423
1.358000	12.00000	12.01409				

PLOT OF Y..(X*S) AND Y-FIT..(*'S) FOLLOWS:

Figure B-3. Continued.

APPENDIX C - Airfoil-to-Complete-Wing Program

User Instructions

This program is written in Fortran IV and is designed to run in single precision on an IBM 370-165 computer. Execution requires 60,000 bytes of core storage and approximately eight seconds to produce the three-dimensional lift, drag, and pitching moment for a given wing-body configuration. For each configuration the program requires the following input data:

(1) The aspect ratio ASPEC, the thickness ratio of the tip TAUT, the thickness ratio of the root TAUR, the taper ratio TAPER, the geometric twist TWIST in degrees (If geometric twist is specified, the aerodynamic twist TWISA must be set to a value of 100.), the number of spanwise stations R (R must be less than or equal to 20.), Reynolds number in millions based on wing mean aerodynamic chord REYND, and a criterion for convergence of the lift distribution DISCR.

(2) Fuselage height to wing span ratio A, fuselage width to wing span ratio B, the height of the wing above the fuselage centerline H, again as a ratio to wing span, wing-body incidence angle ALPHR in degrees, x-coordinate of the moment reference point X, z-coordinate of the moment reference point Z, the aerodynamic twist TWISA in degrees (If aerodynamic twist is specified, the geometric twist TWIST must be set to a value of 100.).

(3) The number of airfoil families (two tables per family) to be read in with this configuration IFAM, a control parameter for reading in wing geometric parameters ISWIT(1), a control parameter for printing out intermediate calculations as they are performed ISWIT(2), a control parameter for printing out matrices ISWIT(3), and an indicator that the tip airfoil is or is not of the same family as the root IRT. A yes action is implied when the control parameter or indicator is set to one; otherwise, the appropriate space is filled with a zero.

4) The 80 characters of the array NAME which are used as a header for identifying output.

(5) The 80 characters of the array TITLEI which serve as identification for the first airfoil table.

(6) The thickness ratio of the airfoil in the first table RTI.

(7) The five coefficients of the lift polynomial CCLRT1 for the airfoil in the first table.

(8) The domain for which the coefficients of CCLRT1 are valid XLO(1) and XHI(1).

310

(9) The five coefficients of the drag polynomial CCDRT1 for the airfoil in the first table.

(10) The domain for which coefficients of CCDRT1 are valid XLO(2) and XHI(2).

(11) The five coefficients of the moment polynomial CCMRT1 for the airfoil in the first table.

(12) The domain for which the coefficients of CCMRT1 are valid XLO(3) and XHI(3).

(13) The five coefficients of the alpha polynomial CALRT1 for the airfoil in the first table.

(14) The domain for which the coefficients of CALRT1 are valid XLO(4) and XHI(4).

(15) A duplication of (4) through (14) for each additional airfoil until the correct number of airfoil coefficient tables are stored.

(16) The 20 elements of the array ALPHB representing the angles of attack for which three-dimensional lift, drag, and moment coefficients are to be calculated. One element of this array must contain the value 99.0 to insure a later return to the main portion of the program.

The program allows for additional configurations to be calculated during the same run. This may be accomplished simply by repeating the previous input. The program will continue execution until it encounters an ASPEC value of 99.0 followed by a blank card.

Statements (1) through (16) represent a complete data set for a particular configuration. The format specification for this data is given in Figure C-1. A sample data set using a 23020 root-23012 tip wing is shown in Figure C-2. The output of this particular data set is given in Figure C-3.

ASPEC	TAUT	TAUR	TAPER	TWIST	R	REYND	DISCR
F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2

A	B	H	ALPHR	X	Z	TWISA	
F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	F10.2	

IFAM	ISWIT(1)	ISWIT(2)	ISWIT(3)	IRT			
I10	I10	I10	I10	I10			

NAME
20A4

TITLE
20A4

RTI							
G16.5							

CCLRTI(1)				CCLRTI(5)
G16.5	G16.5	G16.5	G16.5	G16.5

XLO(1)	XHI(1)		
G16.5	G16.5		

CCDRTI(1)				CCDRTI(5)
G16.5	G16.5	G16.5	G16.5	G16.5

XLO(2)	XHI(2)		
G16.5	G16.5		

CCMRTI(1)				CCMRTI(5)
G16.5	G16.5	G16.5	G16.5	G16.5

XLO(3)	XHI(3)		
G16.5	G16.5		

CALRTI(1)				CALRTI(5)
G16.5	G16.5	G16.5	G16.5	G16.5

XLO(4)	XHI(4)		
G16.5	G16.5		

ADDITIONAL AIRFOIL COEFFICIENT TABLES

ALPHV(1)									
F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0

									ALPHV(20)
F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0	F8.0

Figure C-1. Format specification of input data for the airfoil-to-complete-wing program.

312

313

```
C
      DIMENSION DIST(19),YDA(19),YDAX(19),YX(19),ALPHZ(19),CLO(19),TITLE
     1L(20),TITLE2(20),TITLE3(20),TITLE4(20)
      COMMON ISWIT(3),CI(9),EPS(19),TRANS(19),REY(19),ETA(19),Y1(19),BETA
     1(19,19),TFAC,TRIX(19),TAU(19),ASPEC,TAPER,REYNO,DISCR,PIER,CAB,
     19,EDGE,ALPHR,NP,A,B,H,TAUT,TAUR,TWIST,R,BM,THISA,X,Z,ACC,NAME(20)
     1,CVAL(19),ALPHU(19),CBC(19),ALPHE(19),ALPHB,YD
      COMMON /FUNC/CCLRT1(5),CCDRT2(5),CCDRT1(5),CCMRT1(5),CCM
     1RT2(5),CALRT1(5),CALRT2(5),CCLTT1(5),CCLTT2(5),CCDTT1(5),CCDTT2(5)
     1,CCMTT1(5),CCMTT2(5),CALTT1(5),CALTT2(5),IFAM,IRT,RT1,RT2,TT1,TT2
      COMMON /DOMAIN/XLO(16),XMI(16)
      COMMON /IHOUT/JREAD,JWRITE
      ACCS(X)=ARCOS(X)
      DATA CLO/1990./
C
C              SET INPUT-OUTPUT UNIT NUMBERS
C
    5 JREAD=1
      JWRITE=3
C
C              INPUT DATA SECTION
C
      READ (JREAD,260) ASPEC,TAUT,TAUR,TAPER,TWIST,R,REYNO,DISCR,A,B,H,A
     1LPHR,X,Z,THISA
      IF (ASPEC-99.) 15,10,10
   10 CALL EXIT
   15 CONTINUE
      WRITE (JWRITE,285)
      WRITE (JWRITE,290)
      WRITE (JWRITE,260)
      WRITE (JWRITE,265)
      WRITE (JWRITE,270)
      WRITE (JWRITE,280)
C
      READ (JREAD,275) IFAM,ISWIT,IRT
      READ (JREAD,325) NAME
      IF (IFAM-1) 30,20,20
C
C              INPUT ROOT AIRFOIL FAMILY
C
   20 READ (JREAD,325) TITLE1
      READ (JREAD,295) CCLRT1,XLO(1),XMI(1)
      READ (JREAD,295) CCDRT1,XLO(2),XMI(2)
      READ (JREAD,295) CCMRT1,XLO(3),XMI(3)
      READ (JREAD,295) CALRT1,XLO(4),XMI(4)
      WRITE (JWRITE,280)
      WRITE (JWRITE,330) TITLE1
      WRITE (JWRITE,335) RT1
      WRITE (JWRITE,340)
      READ (JREAD,300) CCLRT2,XLO(5),XMI(5)
      READ (JREAD,305) CCDRT2,XLO(6),XMI(6)
      READ (JREAD,305) CCMRT2,XLO(7),XMI(7)
      READ (JREAD,315) CALRT2,XLO(8),XMI(8)
      WRITE (JWRITE,280) TITLE2
      WRITE (JWRITE,335) RT2
      WRITE (JWRITE,340)

      WRITE (JWRITE,300) CCLRT2,XLO(5),XMI(5)
      WRITE (JWRITE,305) CCDRT2,XLO(6),XMI(6)
      WRITE (JWRITE,310) CCMRT2,XLO(7),XMI(7)
      WRITE (JWRITE,315) CALRT2,XLO(8),XMI(8)
      IF (IFAM-1) 30,30,25
C
C              INPUT TIP AIRFOIL FAMILY
C
   25 READ (JREAD,325) TITLE3
      READ (JREAD,295) TT1
      READ (JREAD,295) CCLTT1,XLO(9),XMI(9)
      READ (JREAD,295) CCDTT1,XLO(10),XMI(10)
      READ (JREAD,295) CCMTT1,XLO(11),XMI(11)
      READ (JREAD,295) CALTT1,XLO(12),XMI(12)
      WRITE (JWRITE,280)
      WRITE (JWRITE,330) TITLE3
      WRITE (JWRITE,335) TT1
      WRITE (JWRITE,340)
      WRITE (JWRITE,300) CCLTT1,XLO(9),XMI(9)
      WRITE (JWRITE,305) CCDTT1,XLO(10),XMI(10)
      WRITE (JWRITE,310) CCMTT1,XLO(11),XMI(11)
      WRITE (JWRITE,315) CALTT1,XLO(12),XMI(12)
      READ (JREAD,325) TITLE4
      READ (JREAD,295) TT2
      READ (JREAD,295) CCLTT2,XLO(13),XMI(13)
      READ (JREAD,295) CCDTT2,XLO(14),XMI(14)
      READ (JREAD,295) CCMTT2,XLO(15),XMI(15)
      READ (JREAD,295) CALTT2,XLO(16),XMI(16)
      WRITE (JWRITE,280)
      WRITE (JWRITE,330) TITLE4
      WRITE (JWRITE,335) TT2
      WRITE (JWRITE,340)
      WRITE (JWRITE,300) CCLTT2,XLO(13),XMI(13)
      WRITE (JWRITE,305) CCDTT2,XLO(14),XMI(14)
      WRITE (JWRITE,310) CCMTT2,XLO(15),XMI(15)
      WRITE (JWRITE,315) CALTT2,XLO(16),XMI(16)
      WRITE (JWRITE,280)
C
C              SWITCH ZERO ON WILL READ IN VALUES FROM
C              CARDS FOR TAU, REY, C, EPS, EDGE, CAB, AND
C              ACC. FORMAT IS 16F5.0. ARRAYS ARE READ IN
C              ROWWISE
C
   30 WRITE (JWRITE,280)
      WRITE (JWRITE,290)
      CALL SETSW
      NP=R-1.
      JP=R./2.
      PIER=3.14159/R
      CALL DATSW(0,I)
      GO TO (35,40),I
   35 READ (JREAD,320) (TAU(I),I=1,NP)
      READ (JREAD,320) (REY(I),I=1,NP)
      READ (JREAD,320) (C(I),I=1,NP)
      READ (JREAD,320) (EPS(I),I=1,NP)
      READ (JREAD,320) EDGE
      READ (JREAD,320) CAB
      READ (JREAD,320) ACC
   40 CONTINUE
C
C              IF NO FUSE (FUSELAGE)
C
      IF (A) 145,145,45
   45 YD=BM*SQRT(1.-HH**2/A**2)
```

```
      ECC=SQRT(A**2-B**2)
      CALL DATSW(0,I)
      GO TO (55,50),I
   50 CRB=2.*(1.-YO)/(ASPEC*(1.+TAPER-2.*YO*TAPER))
   55 TFAC=1.-((YO*TAUR*CRB/(3.141590A*B))*4.,
      JPP=JP+1
C
C                   Y PRIME (I)
C
      DO 60 I=1,JPP
      X(I)=I-1
   60 YDA(I)=YO+(1.-YO)*COS(X(I)*PIER)
C
C          CHECK IF FUSE ELLIPTICAL OR CIRCULAR
C
      IF (A-B) 75,75,65
C
C              ELLIPTICAL FUSE
C
   65 CONTINUE
      DISTX=0.5*(SQRT(1.+(H-ECC)**2)+SQRT(1.+(H+ECC)**2))
      BWX=1./(A-B)*(A-B*DISTX/SQRT(DISTX**2-ECC**2))
      DO 70 I=1,JPP
      DIST(I)=0.5*(SQRT(YDA(I)**2+(H-ECC)**2)+SQRT(YDA(YDA(I)**2+(H+ECC)**2))
     1)
C
C               Y BAR PRIME (I)
C
   70 YDAK(I)=(YDA(I)/(A-B)*(A-B*DIST(I)/SQRT(DIST(I)**2-ECC**2)))/BWX
      GO TO 85
C
C          Y BAR PRIME (I) FOR CIRCULAR FUSE
C
   75 DO 80 I=1,JPP
   80 YDAK(I)=YDA(I)**(1.-A**2/(YDA(I)**2+H**2))
      BWX=1.-A**2/(1.+H**2)
C
C        COMMON TO ELLIPTIC AND CIRCULAR FUSE
C
   85 DO 90 I=1,JPP
C
C               Y BAR (I)
C
      AI=I-1
   90 YX(I)=COS(AI*PIER)
C
C                  Y (I)
C
   95 DO 95 I=2,JPP
      Y(I)=TERP(YCAX(I-1),YX(I),YDAX(I),YDA(I-1),YDA(I))
      DO 100 I=1,JP
      Y(I)=Y(I+1)
  100 YX(I)=YX(I+1)
      M=JP-1
      DO 105 I=1,M
      IRI=I-1
      IRI=IRI-1
      YX(IRI)=-YX(I)
  105 Y(IRI)=-Y(I)
      IF (A-B) 125,125,110
C
C            FUSE IS ELLIPTICAL
C
  110 DO 115 I=1,JP
      DIST(I)=0.5*(SQRT((Y(I)**2+(H-ECC)**2)+SQRT((Y(I)**2+(H+ECC)**2))
      Q=DIST(I)/(SQRT(DIST(I)**2-ECC**2)
      S=1.-(ECC*Y(I)/(DIST(I)**2-ECC**2))**2
      TRANS(I)=1./(A-B)*(A-B*Q/S)
      IRI=R
      IRI=IRI-1
  115 TRANS(IRI)=TRANS(I)
C
C          WING ON TOP OR BOTTOM OF FUSELAGE
C
      IF (A-H) 120,120,200
  120 TRANS(JP)=1.
  125 TRANS(I)=1.-A**2*(Y(I)**2-H**2)/((Y(I)**2+H**2)**2)
      IRI=R
      IRI=IRI-1
  130 TRANS(IRI)=TRANS(I)
      IF (A-H) 135,135,140
  135 TRANS(JP)=1.0
  140 BFX=0.
      GO TO 170
  145 DO 150 I=1,MP
      AI=I
      Y(I)=COS(AI*PIER)
      YX(I)=Y(I)
  150 TRANS(I)=1.
      YO=0.
C
C     SPECIAL CASE IF VALUES HAVE BEEN READ IN
C     DO NOT WANT TO COMPUTE CRB.
C     WILL NOT COMPUTE VALUES ALREADY READ IN
C
      CALL DATSW(0,I)
      GO TO (160,155),I
  155 CRB=2.*(1.-YO)/(ASPEC*(1.+TAPER-2.*YO*TAPER))
  160 TFAC=1.
      BWX=1.
  170 TSTAX=ACOS(BF*X/BWX)
      DO 195 I=1,JP
      AI=I
      C3=AI*PIER
      TSTX=TSTAX
  175 TSTAX=C3    195,175,180
      ISTAR=I
      GC TO 200
  180 AM=C3-TSTAX
      C1=(AI-1.)*PIER
      AK=TSTAX-C1
      IF (AM-AK) 185,185,190
  185 ISTAR=I
      TSTAX=C3
      GO TO 200
  190 ISTAR=I-1
      TSTAX=C1
      GO TO 200
  195 CONTINUE
  200 CALL DATSW(0,I)
      GC TO (215,205),I
  205 EDGE=SQRT(1.+4./(ASPEC**2))
      DO 210 I=1,MP
      C(I)=1.-(1.-TAPER)*(ABS(Y(I))-YO)/(1.-YO)
      TAU(I)=TAUR/C(I)*(1.-(1.-TAPER*TAUR)*(ABS(Y(I))-YO)/(1.-YO))
```

316

```
      CALL AAA(TAU,NP)
      WRITE (JWRITE,130)
      CALL AAA(REV,NP)
      WRITE (JWRITE,135)
      CALL AAA(C,NP)
      WRITE (JWRITE,140)
      CALL MAIN2
      CALL AAA(EPS,NP)
      RETURN
  105 FORMAT (10X,11HMATRIX BETA)
  110 FORMAT (10X,11HMATRIX TRII)
  115 FORMAT (1H1/1H0/35X,20A4/1X)
  120 FORMAT (11X/10X,17HSPANWISE STATIONS)
  125 FORMAT (10X,30HTHICKNESS / CHORD DISTRIBUTION)
  130 FORMAT (10X,24HSECTION REYNOLDS NUMBERS)
  135 FORMAT (10X,18HCHORD DISTRIBUTION)
  140 FORMAT (10X,18HTWIST DISTRIBUTION)
      END

      SUBROUTINE MAIN2

C                SUBROUTINE FOR THE CASE WITHOUT A FLAP

      DIMENSION CBG(19),ALPG(19),ALPHI(19),EPS(19),TRANS(19),REV(19),ETA(19),Y(19),BETA
     1(19,19),TFAC,TRIX(19,19),TAU(19),ASPEC,TAPER,REYND,DISCR,PIER,CRB,
     2C,EDGE,ALPHR,NP,A,B,M,TAUT,TALR,TWIST,R,BMX,TWISA,X,Z,ACC,NAME(20)
     3,CVAL(19),ALPHU(19),CBC(19),ALPHE(19),ALPHB,YC
      COMMON /INOUT/JREAD,JWRITE
      DATA CL0/1990./
      WRITE (JWRITE,140) NAME
      WRITE (JWRITE,145) A,B,ASPEC,M,ALPHR,TAUT
      WRITE (JWRITE,150) TALR,TWIST,R,TWISA,TAPER,REYND,X,Z,DISCR
      WRITE (JWRITE,155)
      WRITE (JWRITE,220)
      WRITE (JWRITE,210)
      WRITE (JWRITE,140)
      WRITE (JWRITE,210)
      WRITE (JWRITE,165)
      IKT=1
C                             INPUT OF ALPHA VALUE
      READ (JREAD,135) ALPHV
    5 ALPHB=ALPHV(IKT)
C                             TEST FOR END OF ALPHA VALUES
      IF (ALPHB-99.) 15,10,15
   10 WRITE (JWRITE,210)
      WRITE (JWRITE,215)
      RETURN
   15 ITR=0
      DO 20 K=1,NP
   20 ALPG(K)=ALPHB+ALPHR*EPS(K)+ALPHB*TFAC*(TRANS(K)-1.)
C                SWITCH NUMBER 3 IS USED FOR AN INTERNAL
C                DUMP OF ARRAYS COMPUTED DURING ITERATION
C                PROCESS

C                                              FIND CL(K)
      CALL DATSW(3,JUNK)
      GO TO (25,30),JUNK
   25 WRITE (JWRITE,170)
      CALL AAA(ALPG,NP)
   30 CALL BRIDG(CVAL,1,NP,1,ALPG)
      DO 40 K=1,NP
      AK=K
   40 CBG(K)=CVAL(K)*(ASPEC/(ASPEC*1.8))*C(K)*CRB*(0.5*(1.+TAPER)*SIN(AK
     1*PTER)/(3.1415*C(K)))
C                                              CHECK FOR DUMP
      CALL DATSW(3,JUNK)
      GO TO (45,50),JUNK
   45 WRITE (JWRITE,175)
      CALL AAA(CVAL,NP)
C                    LOOK UP ALPHA VALUE FOR ZERO LIFT
   50 CALL BRIDG(ALPHZ,1,NP,4,CL0)
      DO 65 K=1,NP
      SIG=0.
      DO 60 K=1,NP
   60 SIG=CBG(K)*BETA(M,K)+SIG
      ALPHU(K)=SIG*(1.+TFAC*(TRANS(K)-1.))
      ALPHI(K)=ALPG(K)-ALPHU(K)
      ALPHE(K)=(ALPHI(K)-ALPHZ(K)*(1.-EDGE))/EDGE
   65 CONTINUE
C                LOOK UP CL VALUES FOR A LIST OF ALPHA
C                                              VALUES
      CALL BRIDG(CVAL,1,NP,1,ALPHE)
      DO 75 K=1,NP
      CBC(K)=CVAL(K)*C(K)*CRB/BMX
   75 DELTA(K)=CBC(K)-CBG(K)
C                                              CHECK FOR DUMP
      CALL DATSW(3,JUNK)
      GO TO (80,85),JUNK
   80 WRITE (JWRITE,180)
      CALL AAA(CBG,NP)
      WRITE (JWRITE,185)
      CALL AAA(ALPHU,NP)
      WRITE (JWRITE,190)
      CALL AAA(ALPHE,NP)
      WRITE (JWRITE,175)
      CALL AAA(CVAL,NP)
      WRITE (JWRITE,195)
      CALL AAA(CBC,NP)
      WRITE (JWRITE,200)
      CALL AAA(DELTA,NP)
C                                              CHECK TOLERANCE
   85 DO 95 K=1,NP
      IF (ABS(DELTA(K))-DISCR) 90,90,100
   90 IF (K-NP) 95,130,130
   95 CONTINUE
```

317

```
      SUMM=C(I)*C(I)+2.*ETA(I)+SUMM
   40 D=ASPEC*SUM*2
      CDLIFT=D*SUM1
      CDI=0.017+59*D*SUM2
      CDP=CDP*SUM3
      CD=CDI+CDP
      ZM=ASPEC*M*PCR0/ACO*SUM
      WRITE (JWRITE,90) ALPM0,COLIFT,CDP,CDI,CD,ZM
      RETURN
   50 FORMAT (10X,35HSECTION PITCHING MOMENT COEFFICIENT)
   55 FORMAT (10X,4HCDC)
   60 FORMAT (11X/10X,33HEFFECTIVE SECTION ANGLE OF ATTACK)
   65 FORMAT (10X,32HSECTION PROFILE DRAG COEFFICIENT)
   70 FORMAT (10X,32HSECTION INDUCED DRAG COEFFICIENT)
   75 FORMAT (10X,40HDISTRIBUTION OF SECTION LIFT COEFFICIENT)
   80 FORMAT (32X,3HP  ,6F10.6,3H  *)
   85 FORMAT (32X,1HP,5X,5HALPMA,7X,2HCL,7X,3HCDP,7X,3HCDI,8X,2HCD,8X,2H
     1CM,5X,1HP)
      END

      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(361),L(19),M(19)
C     SEARCH FOR LARGEST ELEMENT
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF (ABS(BIGA)-ABS(A(IJ))) 5,10,10
    5 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   10 CONTINUE
C     INTERCHANGE ROWS
      J=L(K)
      IF (J-K) 25,25,15
   15 KI=K-N
      DO 20 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   20 A(JI)=HOLD
C     INTERCHANGE COLUMNS
   25 I=M(K)

   30 JP=N*(I-1)
      DO 35 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   35 A(JI)=HOLD
C     DIVIDE COLUMN BY MINUS PIVOT (VALUE OF
C     PIVOT ELEMENT IS CONTAINED IN BIGA)
      IF (I-K) 45,45,30
   40 IF (ABS(BIGA)-1.E-20) 48,48,45
   45 D=0.0
      RETURN
   48 DO 55 I=1,N
   50 IF (I-K) 55,60,55
   55 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   60 CONTINUE
C     REDUCE MATRIX
      DO 75 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 75 J=1,N
      IJ=IJ+N
   65 IF (I-K) 65,75,65
   70 IF (J-K) 70,75,70
      KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   75 CONTINUE
C     DIVIDE ROW BY PIVOT
      KJ=K-N
      DO 85 J=1,N
      KJ=KJ+N
   80 IF (J-K) 80,85,80
      A(KJ)=A(KJ)/BIGA
   85 CONTINUE
C     PRODUCT OF PIVOTS
      D=D*BIGA
C     REPLACE PIVOT BY RECIPROCAL
      A(KK)=1.0/BIGA
   90 CONTINUE
C     FINAL ROW AND COLUMN INTERCHANGE
      K=N
   95 K=K-1
      IF (K) 130,130,100
  100 I=L(K)
      IF (I-K) 115,115,105
  105 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
```

```
                                                    SUBROUTINE SETSW
C                                                                             SET   1
C            SUBROUTINE CHECKS SWITCH SETTINGS ON DATA                        SET   2
C            CARD AND LISTS OPTIONS SELECTED                                  SET   3
C                                                                             SET   4
      COMMON ISWIT(3)                                                         SET   5
      COMMON /INOUT/JREAD,JWRITE                                             SET   6
      K=0                                                                     SET   7
      WRITE (JWRITE,45)                                                       SET   8
      IF (ISWIT(1)) 10,10,5                                                   SET   9
    5 K=K+1                                                                   SET  10
      WRITE (JWRITE,50)                                                       SET  11
   10 IF (ISWIT(2)) 20,20,15                                                  SET  12
   15 K=K+1                                                                   SET  13
      WRITE (JWRITE,55)                                                       SET  14
   20 IF (ISWIT(3)) 30,30,25                                                  SET  15
   25 K=K+1                                                                   SET  16
      WRITE (JWRITE,60)                                                       SET  17
   30 IF (K) 40,35,40                                                         SET  18
   35 WRITE (JWRITE,65)                                                       SET  19
   40 RETURN                                                                  SET  20
C                                                                             SET  21
   45 FORMAT (1H0/1H0,50X,29HSPECIAL FEATURES FOR THIS RUN/1H0)               SET  22
   50 FORMAT (51X,14HREAD IN VALUES/1H0)                                      SET  23
   55 FORMAT (51X,34HTRACE (OUTPUT OF KEY CALCULATIONS)/1H0)                  SET  24
   60 FORMAT (51X,28HPRINT MATRICES BETA AND TRIM/1H0)                        SET  25
   65 FORMAT (51X,31H0NO SWITCH SETTINGS   REGULAR RUN)                       SET  26
      END                                                                     SET  27
                                                                              SET  28
```

```
      SUBROUTINE DATSW(/N/,/ICOND/)                                           DAT   1
C                                                                             DAT   2
C            SUBROUTINE MAKES DECISION BASED ON THE                           DAT   3
C            SETTING OF THE SWITCH   SET=1,NOT SET=0                          DAT   4
C            ICOND=1 IF SWITCH IS SET                                         DAT   5
C            ICOND=2 IF SWITCH IS NOT SET                                     DAT   6
C                                                                             DAT   7
      COMMON ISWIT(3)                                                         DAT   8
      IF (N-3) 5,10,15                                                        DAT   9
    5 IF (ISWIT(1)) 20,20,25                                                  DAT  10
   10 IF (ISWIT(2)) 20,20,25                                                  DAT  11
   15 IF (ISWIT(3)) 20,20,25                                                  DAT  12
   20 ICOND=2                                                                 DAT  13
      RETURN                                                                  DAT  14
   25 ICOND=1                                                                 DAT  15
      RETURN                                                                  DAT  16
      END                                                                     DAT  17
```

```
      SUBROUTINE BRIDG(CC21,/NS/,/NP/,/LOW/,XX)                               BDG   1
C                                                                             BDG   2
C            SUBROUTINE TO INTERPOLATE BETWEEN TABLES                         BDG   3
C                                                                             BDG   4
      DIMENSION XX(19),CC21(19),CC22(19)                                      BDG   5
      COMMON ISWIT(3),C1(19),EPS(19),TRANS(9),REY(19),ETA(19),Y(19),BETA     BDG   6
     1(19,19),TFAC,TRIX(19,19),TAU(19),ASPEC,TAPER,REYNO,DISCR,PIR,CRB,      BDG   7
     2EDGE,ALPHA,MO,A,B,M,TAUT,TAUR,TMIST,A,BWS,THISA,XYZ,ACC,NAME(20)        BDG   8
     3,GVAL(19),ALPNU(19),CRC(19),ALPHC(19),ALPHS,TO                         BDG   9
      COMMON /FUNC/CCLRT1(5),CCLRT2(5),CCLRT1(5),CCOAT2(5),CCOAT1(5),CCR     BDG  10
     1RT2(5),CALRT1(5),CALRT2(5),CCLRT1(5),CCLDT1(5),CCDT2(5),CCR            BDG  11
     2,CCART1(5),CCART2(5),CALTT1(5),CALTT2(5),IFAM,IRT,RT1,RT2,TT1,TT2      BDG  12
```

```
  110 A(KI)=-A(JI)                                                           MIN 106
  115 A(JI)=HOLD                                                             MIN 107
      J=N(K)                                                                 MIN 108
      IF (J-K) 95,95,120                                                    MIN 109
  120 K=K-N                                                                  MIN 110
      DO 125 I=1,N                                                          MIN 111
      KI=KI+N                                                               MIN 112
      HOLD=A(KI)                                                            MIN 113
      JI=KI-K+J                                                             MIN 114
      A(KI)=-A(JI)                                                          MIN 115
  125 A(JI)=HOLD                                                            MIN 116
      GO TO 95                                                              MIN 117
  130 RETURN                                                                MIN 118
      END                                                                   MIN 119
                                                                            MIN 120
```

```
      SUBROUTINE AAA(CLIST,NV)                                               AAA   1
C                                                                           AAA   2
C            SUBROUTINE TO OUTPUT A ONE DIMENSIONAL                         AAA   3
C            ARRAY                                                          AAA   4
      DIMENSION CLIST(NV)                                                   AAA   5
      COMMON /INOUT/JREAD,JWRITE                                           AAA   6
      WRITE (JWRITE,5) (J,CLIST(J),J=1,NV)                                  AAA   7
      WRITE (JWRITE,10)                                                     AAA   8
      RETURN                                                                AAA   9
    5 FORMAT (5(1X,1H(,I2,2H) ,E16.8))                                      AAA  10
   10 FORMAT (21LH )                                                        AAA  11
      END                                                                   AAA  12
```

```
      SUBROUTINE SSS(A,NP)                                                   SSS   1
C                                                                           SSS   2
C            SUBROUTINE TO OUTPUT A SQUARE TWO                              SSS   3
C            DIMENSIONAL ARRAY                                              SSS   4
      DIMENSION A(19,19)                                                    SSS   5
      COMMON /INOUT/JREAD,JWRITE                                           SSS   6
      WRITE (JWRITE,5) ((J,K,A(J,K),K=1,NP),J=1,NP)                         SSS   7
      WRITE (JWRITE,10)                                                     SSS   8
      RETURN                                                                SSS   9
    5 FORMAT (5(1X,1H(,I2,1H,,I2,2H) ,E16.8))                               SSS  10
   10 FORMAT (31LH0/1 )                                                     SSS  11
      END                                                                   SSS  12
```

```
      FUNCTION TER(R1,R2,R3,C1,C3)                                           TER   1
C                                                                           TER   2
C            INTERPOLATION FUNCTION                                         TER   3
C                                                                           TER   4
      IF (R1-R3) 10,5,10                                                    TER   5
    5 TER=C1                                                                 TER   6
      RETURN                                                                TER   7
   10 TER=C1+((C3-C1)*(R2-R1))/((R3-R1))                                     TER   8
      RETURN                                                                TER   9
      END                                                                   TER  10
```

320

```
BDG 13      COMMON /INCUT/JREAC,JWRITE
BDG 14      CLRT1(K)=FUNX(CCLRT1,X,1)
BDG 15      CDRT1(K)=FUNX(CCDRT1,X,2)
BDG 16      CMRT1(K)=FUNX(CCMRT1,X,3)
BDG 17      ALRT1(K)=FUNX(CALRT1,X,4)
BDG 18      CLRT2(K)=FUNX(CCLRT2,X,5)
BDG 19      CDRT2(K)=FUNX(CCDRT2,X,6)
BDG 20      CMRT2(K)=FUNX(CCMRT2,X,7)
BDG 21      ALRT2(K)=FUNX(CALRT2,X,8)
BDG 22      CLTT1(K)=FUNX(CCLTT1,X,9)
BDG 23      CDTT1(K)=FUNX(CCDTT1,X,10)
BDG 24      CMTT1(K)=FUNX(CCMTT1,X,11)
BDG 25      ALTT1(K)=FUNX(CALTT1,X,12)
BDG 26      CLTT2(K)=FUNX(CCLTT2,X,13)
BDG 27      CDTT2(K)=FUNX(CCDTT2,X,14)
BDG 28      CMTT2(K)=FUNX(CCMTT2,X,15)
BDG 29      ALTT2(K)=FUNX(CALTT2,X,16)
BDG 30      IF (IRT) 5,5,45
BDG 31    5 DO 40 K=NS,NP
BDG 32 C
BDG 33 C            GET TAU VALUE FROM ARRAY
BDG 34 C
BDG 35      TAU2=TAU(K)
BDG 36 C
BDG 37 C            PERFORM LOOK FOR LIFT, DRAG, PITCHING
BDG 38 C            MOMENT OR FLAP CASE
BDG 39 C
BDG 40      GO TO (15,20,25,30),LDM
BDG 41   15 YY=TERP(TT1,TAU,YT2,CLTT1(XK(K)),CLTT2(XK(K)))
BDG 42      GO TO 35
BDG 43   20 YY=TERP(TT1,TAU,YT2,CDTT1(XK(K)),CDTT2(XK(K)))
BDG 44      GO TO 35
BDG 45   25 YY=TERP(TT1,TAU,YT2,CMTT1(XK(K)),CMTT2(XK(K)))
BDG 46      GO TO 35
BDG 47   30 YY=TERP(TT1,TAU,YT2,ALTT1(XK(K)),ALTT2(XK(K)))
BDG 48 C
BDG 49 C            STORE VALUES FOUND IN LOOK UP
BDG 50 C
BDG 51   35 CCZ2(K)=YY
BDG 52 C
BDG 53 C            REPEAT FOR NUMBER OF VALUES TO BE FOUND
BDG 54 C
BDG 55   40 CONTINUE
BDG 56   45 DO 75 K=NS,NP
BDG 57      TAU2=TAU(K)
BDG 58 C
BDG 59 C            LOOK UP FOR SECOND LIFT, DRAG, PITCHING
BDG 60 C            MOMENT, OR FLAP CASE
BDG 61 C
BDG 62      GO TO (50,55,60,65),LDM
BDG 63   50 YY=TERP(RT1,TAU,RT2,CLRT1(XK(K)),CLRT2(XK(K)))
BDG 64      GO TO 70
BDG 65   55 YY=TERP(RT1,TAU,RT2,CDRT1(XK(K)),CDRT2(XK(K)))
BDG 66      GO TO 70
BDG 67   60 YY=TERP(RT1,TAU,RT2,CMRT1(XK(K)),CMRT2(XK(K)))
BDG 68      GO TO 70
BDG 69   65 YY=TERP(RT1,TAU,RT2,ALRT1(XK(K)),ALRT2(XK(K)))
BDG 70 C
BDG 71 C            STORE VALUES FOUND IN LOOK UP
BDG 72 C
BDG 73   70 CCZ1(K)=YY
BDG 74   75 CONTINUE
BDG 75      IF (IRT) 80,80,90
BDG 76   80 DO 85 K=NS,NP
BDG 77      YK=ABS(VK))
BDG 78   85 CCZ1(K)=TERP(VO,VK,1.,CCZ1(K),CCZ2(K))
BDG 79   90 RETURN
BDG 80      END
FNX 1       FUNCTION FUNX(C,X,I)
FNX 2       DIMENSION C(5)
FNX 3       COMMON /DOMAIN/XLO(16),XHI(16)
FNX 4       XH=XHI(I)
FNX 5       XL=XLO(I)
FNX 6       IF (X.LT.XL) GO TO 5
FNX 7       IF (X.GT.XH) GO TO 10
FNX 8       FUNX=FUN(C,X)
FNX 9       RETURN
FNX 10    5 XDEL=.05*(XH-XL)
FNX 11      FXL=FUN(C,XL)
FNX 12      XLP=XL+XDEL
FNX 13      FXLP=FUN(C,XLP)
FNX 14      SL=(FXLP-FXL)/XDEL
FNX 15      FUNX=SL*(X-XL)+FXL
FNX 16      RETURN
FNX 17   10 XDEL=.05*(XH-XL)
FNX 18      FXH=FUN(C,XH)
FNX 19      XHM=XH-XDEL
FNX 20      FXHM=FUN(C,XHM)
FNX 21      SH=(FXH-FXHM)/XDEL
FNX 22      FUNX=SH*(X-XH)+FXH
FNX 23      RETURN
FNX 24      END
FUN 1       FUNCTION FUN(C,X)
FUN 2       DIMENSION C(15)
FUN 3       FUN=C(1)+(C(2)+(C(3)+(C(4)+C(5)*X)*X)*X)*X
FUN 4       RETURN
FUN 5       END
```

321

Sample Output

```
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
•                        TWO DIMENSIONAL CURVE FIT FUNCTION DATA                      •
•                     OF THE FORM  Y=C(0)+C(1)*X+C(2)*X**2+...                         •
•                                                                                     •
•        2 012 SERIES AIRFOIL / ALPHA=-4,-2,0,2,4,6,8,10,12 / RN=3.49 / MACH NO.=.2    •
•                             THICKNESS RATIO= 0.12                                    •
•                   C(0)      C(1)      C(2)      C(3)      C(4)                        •
•  CL VERSUS ALPHA  0.12631   0.11213   0.00014  -0.00003  -0.00000   DOMAIN=  -4.0000 TO  12.0000 •
•  CD VERSUS CL     0.00708  -0.00202   0.00195   0.00057   0.00269   DOMAIN=  -0.3196 TO   1.3580 •
•  CM VERSUS CL    -0.00751  -0.00769  -0.00095  -0.00205   0.00429   DOMAIN=  -0.3196 TO   1.3580 •
•  ALPHA VERSUS CL -1.13580   9.02350  -0.05586  -0.49714   0.65984   DOMAIN=  -0.3196 TO   1.3580 •
•                                                                                     •
•        23021 SERIES AIRFOIL / ALPHA=-4,-2,0,2,4,6,8,10,12 / RN=3.49 / MACH NO.=.2    •
•                             THICKNESS RATIO= 0.21                                    •
•                   C(0)      C(1)      C(2)      C(3)      C(4)                        •
•  CL VERSUS ALPHA  0.13324   0.11788  -0.00013  -0.00015   0.00001   DOMAIN=  -4.0000 TO  12.0000 •
•  CD VERSUS CL     0.30775  -C.00047   0.00338   0.00154  -0.00003   DOMAIN=  -0.3295 TO   1.4304 •
•  CM VERSUS CL    -0.30176  -C.01808  -0.00624   0.01679  -0.00705   DOMAIN=  -0.3295 TO   1.4304 •
•  ALPHA VERSUS CL -1.12880   8.48720  -0.32278   1.16880  -0.42162   DOMAIN=  -0.3295 TO   1.4304 •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

SPECIAL FEATURES FOR THIS RUN

NO SWITCH SETTINGS REGULAR RUN

23020 ROOT SERIES - 23012 SERIES TIP

```
      SPANWISE STATIONS
( 1)  0.98768842   ( 2)  C.95105666   ( 3)  0.89100677   ( 4)  0.80901742   ( 5)  0.70710737
( 6)  0.58778608   ( 7)  0.45399171   ( 8)  0.30901823   ( 9)  0.15643668   (10)  0.22214408E-05
(11) -0.15643227   (12) -0.30901492   (13) -0.45398778   (14) -0.58778286   (15) -0.70710474
(16) -0.80901533   (17) -0.89100486   (18) -0.95105547   (19) -0.98768783   (

      THICKNESS / CHORD DISTRIBUTION
( 1)  0.12241757   ( 2)  0.12911910   ( 3)  0.13873547   ( 4)  0.14969081   ( 5)  0.16069812
( 6)  0.17094332   ( 7)  0.18003339   ( 8)  0.18786055   ( 9)  0.19447541   (10)  0.19999987
(11)  0.19447559   (12)  0.18786073   (13)  0.18003362   (14)  0.17094356   (15)  0.16069835
(16)  0.14969105   (17)  0.13873571   (18)  0.12911928   (19)  0.12241769   (

      SECTION REYNOLDS NUMBERS
( 1)  1.9141264   ( 2)  2.0173960   ( 3)  2.1866846   ( 4)  2.4178228   ( 5)  2.7051210
( 6)  3.0415030   ( 7)  3.4186869   ( 8)  3.8273869   ( 9)  4.2575340   (10)  4.6485426
(11)  4.2575464   (12)  3.8273954   (13)  3.4186974   (14)  3.0415125   (15)  2.7051287
(16)  2.4178295   (17)  2.1866903   (18)  2.0173998   (19)  1.9141293   (

      CHORD DISTRIBUTION
( 1)  0.40738696   ( 2)  0.42936599   ( 3)  0.46539593   ( 4)  0.51458955   ( 5)  0.57573557
( 6)  0.64732838   ( 7)  0.72760499   ( 8)  0.81458908   ( 9)  0.90613800   (10)  0.99999869
(11)  0.90614069   (12)  0.81458105   (13)  0.72760737   (14)  0.64733028   (15)  0.57573718
(16)  0.51458080   (17)  0.46539712   (18)  0.42936671   (19)  0.40738732   (

      TWIST DISTRIBUTION
( 1) -3.3942242   ( 2) -3.1010373   ( 3) -2.6803169   ( 4) -2.2010231   ( 5) -1.7194519
( 6) -1.2712250   ( 7) -0.87353462   ( 8) -0.53109646   ( 9) -0.24169743   (10) -0.311001998E-05
(11) -0.24168992   (12) -0.53108948   (13) -0.87352425   (14) -1.2712145   (15) -1.7194414
(16) -2.2010117   (17) -2.6803036   (18) -3.1010256   (19) -3.3942194   (
```

23020 ROOT SERIES - 23012 SERIES TIP

```
. /../../../../../../../../../../../../../../../../../../../../../../../../../../../../.

      BODY HEIGHT / SPAN . . . . . =    0.0        BODY WIDTH / SPAN. . . . . . =    0.0
      ASPECT RATIO . . . . . . . =   10.05        WING HEIGHT / SPAN . . . . =    0.0
      WING BODY INCIDENCE, DEG . =    0.0         TIP THICKNESS CHORD. . . . =    0.12
      ROOT THICKNESS CHORD . . . =    0.20        GEOMETRIC TWIST, DEG . . . =   -3.50
      NUMBER OF SPANWISE STATIONS. =  20.00       AERODYNAMIC TWIST, DEG . . =   -3.49
      TAPER RATIO. . . . . . . . =    0.40        REYNOLDS NUMBER. . . . . . =    3.49
      COORDINATES OF MOMENT REFERENCE POINT       X=   0.0         Z=   0.0
      VALUE OF DISCRIMINANT. . . . = 0.001000

./../../../../../../../../../../../../../../../../../../../../../../../../../../../../.
```

```
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
•            THREE DIMENSIONAL LIFT, DRAG, AND MOMENT DATA    •
•                                                            •
•    ALPHA       CL        CDP       CDI       CD       CM    •
•  -4.000000  -0.351687  0.008190  0.004400  0.012590  0.000474 •
•  -2.000000  -0.169874  0.007779  0.001311  0.009090 -0.001481 •
•   0.0        0.015495  0.007519  0.000352  0.007870 -0.004108 •
•   2.000000   0.202082  0.007467  0.001606  0.009073 -0.006987 •
•   4.000000   0.387578  0.007686  0.005057  0.012743 -0.009734 •
•   6.000000   0.570166  0.008247  0.010600  0.018846 -0.012090 •
•   8.000000   0.748508  0.009229  0.018074  0.027303 -0.013931 •
•  10.000000   0.921770  0.010713  0.027294  0.038007 -0.015250 •
•  12.000000   1.089635  0.012778  0.038080  0.050858 -0.016131 •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

Figure C-3. Sample output of the airfoil-to-complete-wing program.

APPENDIX D - PLOT Program

User Instructions

This program was originally written for a CDC 6000 computer (Ref. 113) and was then modified to run in single precision on an IBM 370-165. Given a set of input data the program generates the necessary instructions for automatic plotting of an airplane numerical model and can be used to draw three-view and oblique orthographic projections, as well as perspective projections. These plots are very useful in checking the validity of numerical model data. The program has an average execution time of 3 minutes and 20 seconds for a job yielding 8 different views of the same aircraft.

In order to be compatable with the potential flow program used in this report (see Appendix F), the body coordinates are specified with the body nose at, or near the origin of the coordinate system, and the body's longitudinal axis is extended along the X-axis (see Figure D-1). The origin of the Y-axis must lie in the XZ-plane of symmetry, and the Z-axis must be the vertical axis of the body.

Reference 113 does an excellent job of describing the input data cards; therefore, the data specification given below is taken directly from that description. Additional information on this program is available in the cited reference.

Configuration Cards

Since the airplane has to be symmetrical about the XZ-plane, only half of the airplane need be described to the computer. The convention used in presenting the input data is that the half of the airplane on the positive Y-side of the XZ-plane is presented. The program then uses this information to construct the complete airplane. The number of input cards depends on the number of components used to describe the configuration, whether a component has been described previously, and the amount of detail used to describe each component. The method of input is by FORTRAN "READ" statements.

Columns	FORTRAN Name	Description
01 to 03	JO	If JO=0, no reference area If JO=1, reference area to be read If JO=2, reference area same as previously read

Columns	FORTRAN Name	Description
04 to 06	J1	If J1=0, no wing data If J1=1, cambered wing data to be read If J1=-1, uncambered wing data to be read If J1=2, wing data same as previously read
07 to 09	J2	If J2=0, no fuselage data If J2=1, data for arbitrarily shaped fuselage to be read If J2=-1, data for circular fuselage to be read (with J6=0, fuselage will be cambered; with J6=-1, fuselage will be symmetrical with XY-plane; with J6=1, entire configuration will be symmetrical with XY-plane) If J2=2, fuselage data same as previously read
10 to 12	J3	If J3=0, no pod data If J3=1, pod data to be read If J3=2, pod data same as previously read
13 to 15	J4	If J4=0, no fin data If J4=1, fin data to be read If J4=2, fin data same as previously read
16 to 18	J5	If J5=0, no canard data If J5=1, canard data to be read If J5=2, canard data same as previously read
19 to 21	J6	Simplification code: If J6=0, indicates a cambered circular or arbitrary fuselage if J2 ≠ 0 If J6=1, complete configuration is symmetrical with respect to XY-plane, which implies uncambered circular fuselage if there is a fuselage If J6=-1, indicates uncambered circular fuselage with J2 ≠ 0
22 to 24	NWAF	Number of airfoil sections used to describe the wing; $2 \leq$ NWAF ≤ 20
25 to 27	NWAFOR	Number of ordinates used to define each wing airfoil section; $3 \leq$ NWAFOR ≤ 30

324

Columns	FORTRAN Name	Description
28 to 30	NFUS	Number of fuselage segments; $1 \leq$ NFUS ≤ 4
31 to 33	NRADX(1)	Number of points used to represent half-section of first fuselage segment; if fuselage is circular, the program computes indicated number of y- and z-ordinates; $3 \leq$ NRADX(1) ≤ 30
34 to 36	NFORX(1)	Number of stations for first fuselage segment; $4 \leq$ NFORX(1) ≤ 30
37 to 39 40 to 42	NRADX(2) NFORX(2)	Same as NRADX(1) and NFORX(1), but for second fuselage segment
43 to 45 46 to 48	NRADX(3) NFORX(3)	Same as NRADX(1) and NFORX(1), but for third fuselage segment
49 to 51 52 to 54	NRADX(4) NFORX(4)	Same as NRADX(1) and NFORX(1), but for fourth fuselage segment
55 to 57	NP	Number of pods described; NP ≤ 9
58 to 60	NPODOR	Number of stations at which pod radii are to be specified; $4 \leq$ NPODOR ≤ 30
61 to 63	NF	Number of fins (vertical tails) described; NF ≤ 6
64 to 66	NFINOR	Number of ordinates used to define each fin airfoil section; $3 \leq$ NFINOR ≤ 10
67 to 69	NCAN	Number of canards (horizontal tails) described; NCAN ≤ 2
70 to 72	NCANOR	Number of ordinates used to define each canard airfoil section; $3 \leq$ NCANOR ≤ 10; if NCANOR is given a negative sign, the program will expect to read lower ordinates also; otherwise, airfoil is assumed to be symmetrical

Cards 3, 4, . . . - remaining data input cards. - The remaining data input contain a detailed description of each component of the airplane. Each card contains up to 10 values, each value punched in a 7-column field with a decimal and may be identified in columns 73 to 80. The cards are arranged in the following order: reference area, wing data cards, fuselage data cards, pod (or nacelle) data cards, fin (vertical tail) data cards, and canard (or horizontal tail) data cards.

325

Reference area card: The reference area value is punched in columns 1 to 7 and may be identified as REFA in columns 73 to 80.

Wing data cards: The first wing data card (or cards) contains the locations in percent chord at which the ordinates of all the wing airfoils are to be specified. There will be exactly NWAFOR locations in percent chord given. Each card may be identified in columns 73 to 80 by the symbol XAFj where j denotes the number of the last location in percent chord given on that card. For example, if NWAFOR=16, there are 16 ordinates to be specified for every airfoil, and two data cards will be required. The first XAF card is identified as XAF 10 and the second as XAF 16.

The next wing data cards (there will be NWAF cards) each contain four numbers which give the origin and chord length of each of the wing airfoils that is to be specified. The cards representing the most inboard airfoil are given first, followed by the cards for successive airfoils. The information is arranged on each card as follows:

Columns	Description
1 to 7	x-ordinate of airfoil leading edge
8 to 14	y-ordinate of airfoil leading edge
15 to 21	z-ordinate of airfoil leading edge
22 to 28	airfoil streamwise chord length
73 to 80	card identification, WAFORGj where j denotes the particular airfoil; for example, WAFORG1 denotes first (most inboard) airfoil

If a cambered wing has been specified, the next set of wing data cards is the mean camber line (TZORD) cards. The first card contains up to 10 Δz values, referenced to the z-ordinate of the airfoil leading edge, at each of the specified percents of chord for the first airfoil. If more than 10 values are to be specified for each airfoil (there will be NWAFOR values), the remaining values are continued on successive cards. The remaining airfoils are described in the same manner, data for each airfoil starting on a new card, and the cards arranged in the order which begins with the most inboard airfoil and proceeds to the outboard. Each card may be identified in columns 73 to 80 as TZORDj, where j denotes the particular airfoil.

Next are the wing airfoil ordinate (WAFORD) cards. The first card contains up to 10 half-thickness ordinates of the first airfoil expressed as percent chord. If more than 10 ordinates are to be specified for each airfoil (there will be NWAFOR values), the remaining ordinates are continued on successive cards. The remaining airfoils are each described in the same

326

manner, and the cards are arranged in the order which begins with the most inboard airfoil and proceeds to the outboard. Each card may be identified in columns 73 to 80 as WAFORDj, where j denotes the particular airfoil.

Fuselage data cards: The first card (or cards) specifies the x values of the fuselage stations of the first segment. There will be NFORX(1) values and the cards may be identified in columns 73 to 80 by the symbol XFUSj where j denotes the number of the last fuselage station given on that card.

If the fuselage is circular and cambered, the next set of cards specifies the z locations of the center of the circular sections. There will be NFORX(1) values and the cards may be identified in columns 73 to 80 by the symbol ZFUSj where j denotes the number of the last fuselage station given on that card.

If the fuselage is circular, the next card (or cards) gives the fuselage cross-sectional areas, and may be identified in columns 73 to 80 by the symbol FUSARDj where j denotes the number of the last fuselage station given on that card. If the fuselage is of arbitrary shape, the y-ordinates for a half-section are given (NRADX(1) values) and identified in columns 73 to 80 as Yi where i is the station number. Following these are the corresponding z-ordinates (NRADX(1) values) for the half-section identified in columns 73 to 80 as Zi where i is the station number. Each station will have a set of Y and Z cards and the convention of ordering the ordinates from bottom to top is observed.

For each fuselage segment a new set of cards as described must be provided. The segment descriptions should be given in order of increasing values of x.

Pod data cards: The first pod or nacelle data card specifies the location of the origin of the first pod. The information is arranged on the card as follows:

Columns	Description
1 to 7	x-ordinate of origin of first pod
8 to 14	y-ordinate of origin of first pod
15 to 21	z-ordinate of origin of first pod
73 to 80	card identification, PODORGj where j denotes pod number

The next pod input data card (or cards) contains the x-ordinates, referenced to the pod origin, at which the pod radii (there will be NPODOR of them)

327

are to be specified. The first x-value must be zero, and the last x-value is the length of the pod. These cards may be identified in columns 73 to 80 by the symbol XPODj where j denotes the pod number. For example, XPOD1 represents the first pod.

The next pod input data cards give the pod radii corresponding to the pod stations that have been specified. These cards may be identified in columns 73 to 80 as PODRj where j denotes the pod number.

For each additional pod, new PODORG, XPOD, and PODR cards must be provided. Only single pods are described but the program assumes that if the y-ordinate is not zero an exact duplicate is located symmetrically with respect to the XZ-plane; a y-ordinate of zero implies a single pod.

Fin data cards: Exactly three data input cards are used to describe a fin. The information presented on the first fin data input card is as follows:

Columns	Description
1 to 7	x-ordinate of lower airfoil leading edge
8 to 14	y-ordinate of lower airfoil leading edge
15 to 21	z-ordinate of lower airfoil leading edge
22 to 28	chord length of lower airfoil
29 to 35	x-ordinate of upper airfoil leading edge
36 to 42	y-ordinate of upper airfoil leading edge
43 to 49	z-ordinate of upper airfoil leading edge
50 to 56	chord length of upper airfoil
73 to 80	card identification, FINORGj where j denotes fin number

The second fin data input card contains up to 10 locations in percent chord (exactly NFINOR of them) at which the fin airfoil ordinates are to be specified. The card may be identified in columns 73 to 80 as XFINj where j denotes the fin number.

The third fin data input card contains the fin airfoil half-thickness ordinates expressed in percent chord. Since the fin airfoil must be symmetrical,

328

only the ordinates on the positive y side of the fin chord plane are specified. The card identification, FINORDJ, may be given in columns 73 to 80, where J denotes the fin number.

For each fin, new FINORG, XFIN, and FINORD cards must be provided.

Only single fins are described but the program assumes that if the y-ordinate is not zero an exact duplicate is located symmetrically with respect to the XZ-plane; a y-ordinate of zero implies a single fin.

Canard data cards: If the canard (or horizontal tail) airfoil is symmetrical, exactly three cards are used to describe a canard, and the input is given in the same manner as for the fin. If, however, the canard airfoil is not symmetrical (indicated by a negative value of NCANOR), a fourth canard data input card will be required to give the lower ordinates. The information presented on the first canard data input card is as follows:

Columns	Description
1 to 7	x-ordinate of inboard airfoil leading edge
8 to 14	y-ordinate of inboard airfoil leading edge
15 to 21	z-ordinate of inboard airfoil leading edge
22 to 28	chord length of inboard airfoil
29 to 35	x-ordinate of outboard airfoil leading edge
36 to 42	y-ordinate of outboard airfoil leading edge
43 to 49	z-ordinate of outboard airfoil leading edge
50 to 56	chord length of outboard airfoil
73 to 80	card identification, CANORGJ where J denotes the canard number

The second canard data input card contains up to 10 locations in percent chord (exactly NCANOR of them) at which the canard airfoil ordinates are to be specified. The card may be identified in columns 73 to 80 as XCANJ where J denotes the canard number.

The third canard data input card contains the upper half-thickness ordinates, expressed in percent chord, of the canard airfoil. This card may be identified in columns 73 to 80 as CANORDJ where J denotes the canard number. If the canard

329

airfoil is not symmetrical, the lower ordinates are presented on a second CANORD card. The program expects both upper and lower ordinates to be punched as positive values in percent chord.

For another canard, new CANORG, XCAN, and CANORD cards must be provided.

Plot Cards

A single card contains all the necessary information for one plot. The available options and the necessary input for each are described in the succeeding sections.

Orthographic projections. - For orthographic projections, the card should be set up as follows (See Figure D-5):

Columns	FORTRAN Name	Description
1	HORZ	"X", "Y", or "Z" for horizontal axis
3	VERT	"X", "Y", or "Z" for vertical axis
5 to 7	TEST1	Word "OUT" for deletion of hidden lines; otherwise, leave blank
8 to 12	PHI	Roll angle, degrees (See Figure D-1)
13 to 17	THETA	Pitch angle, degrees (See Figure D-1)
18 to 22	PSI	Yaw angle, degrees (See Figure D-1)
48 to 52	PLOTSZ	PLOTSZ determines the size of plot (scale factor is computed using PLOTSZ and maximum dimension of configuration)
53 to 55	TYPE	Word "ORT"
72	KODE	If KODE=0, continue reading plot cards If KODE=1, after processing this plot, read new configuration description

An attempt is made to center the given configuration within the specified field. If the desired plot size is greater than 28 inches, centering is attempted within 28 inches so care must be taken in choosing the view. Minimum values are adjusted so that body axis lines with no rotation angles coincide with grid lines on the plotter paper. Therefore, the plotter pen should always be positioned exactly 1 inch from the side of the plotting space and on the intersection of heavy grid lines at the start of plotting.

330

Plan, front, and side views (stacked). - For plan, front, and side views, the card should be set up as follows (See Figure D-4):

Columns	FORTRAN Name	Description
8 to 12	PHI	y-origin on paper of plan view, inches
13 to 17	THETA	y-origin on paper of side view, inches
18 to 11	PSI	y-origin on paper of front view, inches
48 to 52	PLOTSZ	PLOTSZ determines size of plot (a scale factor is computed using PLOTSZ and maximum dimension of configuration)
53 to 55	TYPE	Word "VU3"
72	KODE	If KODE=0, continue reading plot cards If KODE=1, after processing this plot, read new configuration description

Perspective views. - For perspective views, the card should be set up as follows (See Figure D-10):

Columns	FORTRAN Name	Description
8 to 12	PHI	x of view point (location of viewer) in data coordinate system
13 to 17	THETA	y of view point in data coordinate system
18 to 22	PSI	z of view point in data coordinate system
23 to 27	XF	x of focal point (determines direction and focus) in data coordinate system
28 to 32	YF	y of focal point in data coordinate system
33 to 37	ZF	z of focal point in data coordinate system
38 to 42	DIST	Distance from eye to viewing plane, inches
43 to 47	FMAG	Viewing-plane magnification factor; it controls size of projected image

331

Columns	FORTRAN Name	Description
48 to 52	PLOTSZ	Diameter of viewing plane, inches; DIST and PLOTSZ together determine a cone which is field of vision; PLOTSZ value is also relative to type of viewer which is to be used.
53 to 55	TYPE	Word "PER"
72	KODE	If KODE=0, continue reading plot cards If KODE=1, after processing this plot, read new configuration description.

Stereo frames suitable for viewing in a stereoscope. - For stereo frames suitable for viewing in a stereoscope, the input is identical to that for the perspective views except that the word "STE" is used in columns 53 to 55.

Specification of the cards above represent a complete set of data for a particular body. The format specification for this data is given in the above text. A sample data set of a Cessna 182 light aircraft is shown in Figure D-2. The output of this particular data set, with different plot cards, is shown in Figures D-3 through D-14.

Since plotting software is different at almost every computing facility, this section of the appendix is included to help the user specify the appropriate software plotting instructions expected by the plot program. To institute a plotting procedure at any computing facility the user's program must first be linked with the plotter. In the original plot program linkage was established by the statement: CALL CALCOMP. The user must provide the necessary instructions to produce the same result at his facility. This instruction is given on card COO 35 in the Program Listing presented in the next section of this appendix. In addition, at most installations an instruction must also be given to close the plotter data set (i.e. turn the plotter off) after plotting has been completed. In the original plot program this was accomplished by the statement: CALL CALPLT(0.,0.,999). The user must provide an equivalent instruction for his installation as seen on card COO 67 in the Program Listing.

The actual plotting in the program is accomplished using three basic subroutines (CALPLT, NOTATE, and LINE) from the CalComp software package. For the program to operate properly, the user must either provide the original CalComp routines or provide three equivalent dummy subroutines as was necessary at the N. C. State computing facility. Given below is a description of the arguments to, and the results produced by these subroutines. Also included are listings of the equivalent dummy subroutines used at N. C. State to produce the same results as the original CalComp subroutines.

Subroutine CALPLT

Purpose:
To move the plotter pen to a new location with the pen either up or down, and to turn off the plotter.

Use:
CALL CALPLT(X,Y,IPEN)

where

X,Y   are the floating point values for pen movement.

IPEN=2   pen is moved in a lowered position.
=3   pen is moved in a raised position.

Negative IPEN (-2 or -3) will assign $X = 0$, $Y = 0$ as the location of the pen after moving the pen to X,Y (create a new reference point or origin).

333

IPEN=999   Turns the plotter off, the X and Y
values are ignored.

Restrictions: All X and Y coordinates must be expressed as
floating point inches (actual page dimensions)
in deflection from the origin.

(Equivalent N. C. State Routine)

```
SUBROUTINE CALPLT(A,B,I)
DIMENSION A(1),B(1)
IF (I.LT.0) GO TO 5
J=I-2
CALL PLOT(A,B,J)
RETURN
5 J=IABS(I)
J=J-2
CALL PLOT(A,B,J)
CALL ORIGIN(A,B,1.0)
RETURN
END
```

Subroutine NOTATE

Purpose: To draw alphanumeric information for annotation
and labeling.

Use: CALL NOTATE(X,Y,HEIGHT,BCD,THETA,N)

where

X,Y   are the floating point page co-
ordinates of the first character.

For alphanumeric characters, the
coordinates of the lower left-hand
corner of the characters are
specified.

HEIGHT   specifies the height in floating point
inches for a full-size character.

BCD   is the string of characters to be
drawn and is usually written in the
form: nHXXXX--- (the same way an alpha
message is written using FORTRAN for-

334

BCD con't

mat statements). Instead of specifying
alpha information as above, one may give
the beginning storage location of an
array containing alphanumeric infor-
mation.

THETA   is the angle in floating point degrees
             at which the information is to be
             drawn. Zero degrees will print hori-
             zontally reading from left to right, 90°
             will print the line vertically reading
             from bottom to top, 180° will print the
             line horizontally reading from right to
             left (i.e., upside down), and 270°
             will print vertically reading from top
             to bottom.

N   is the number of characters, including
             blanks, in the label.

(Equivalent N. C. State Routine)

```
    SUBROUTINE NOTATE(A,B,C,D,E,I)
    DIMENSION D(1),DD(21)
    DATA STOP/4H   _/
    J=I/4
    XI=I
    XR=XI/4
    IF ((XR-J).GT.0.1)J=J+1
    DO 5 K=1,J
  5 DD(K)=D(K)
    DD(J+1)=STOP
    CALL SYMBOL(A,B,C,DD,E)
    RETURN
    END
```


Subroutine LINE

Purpose: To draw a continuous line through a set of suc-
              cessive data points where the minimum values and
              scale factors are stored at the end of the data
              arrays.

Use: CALL LINE(XARRAY,YARRAY,N,K,J,L,S)

335

where

XARRAY and YARRAY are the names of arrays containing the X values and Y values, respectively, to be plotted. Values must be in floating point.

N	is the number of points to be plotted.
K = 1	this value of K is constant in the plot program.
J = 0	for line plot. Only line plots are used in the plot program.
L	is an integer describing symbol to be used. This variable is not used but a space for it in the calling sequence must be provided.
S	is the desired symbol height. This variable is not used but a space for it in the calling sequence must be provided.

Restrictions: LINE expects the adjusted minimums and scale factors. These two parameters (two for the XARRAY and two for the YARRAY) are automatically calculated and provided by the plotting program at the ends of the XARRAY and YARRAY respectively. The points actually plotted by LINE are

(XARRAY(J) - XARRAY(N+1))/XARRAY(N+2)   for J=1,N

and

(YARRAY(J) - YARRAY(N+1))/YARRAY(N+2)   for J=1,N.

(Equivalent N. C. State Routine)

```
SUBROUTINE LINE(A,B,I,J,K,L,S)
DIMENSION A(1),B(1),X(31),Y(31)
XMIN=A(I+1)
XSCALE=A(I+2)
YMIN=B(I+1)
YSCALE=B(I+2)
DO 5 II=1,I
X(II)=(A(II)-XMIN)/XSCALE
```

336

```
5 Y(II)=(B(II)-YMIN)/YSCALE
  CALL PLOT(X,Y,I)
  RETURN
  END
```
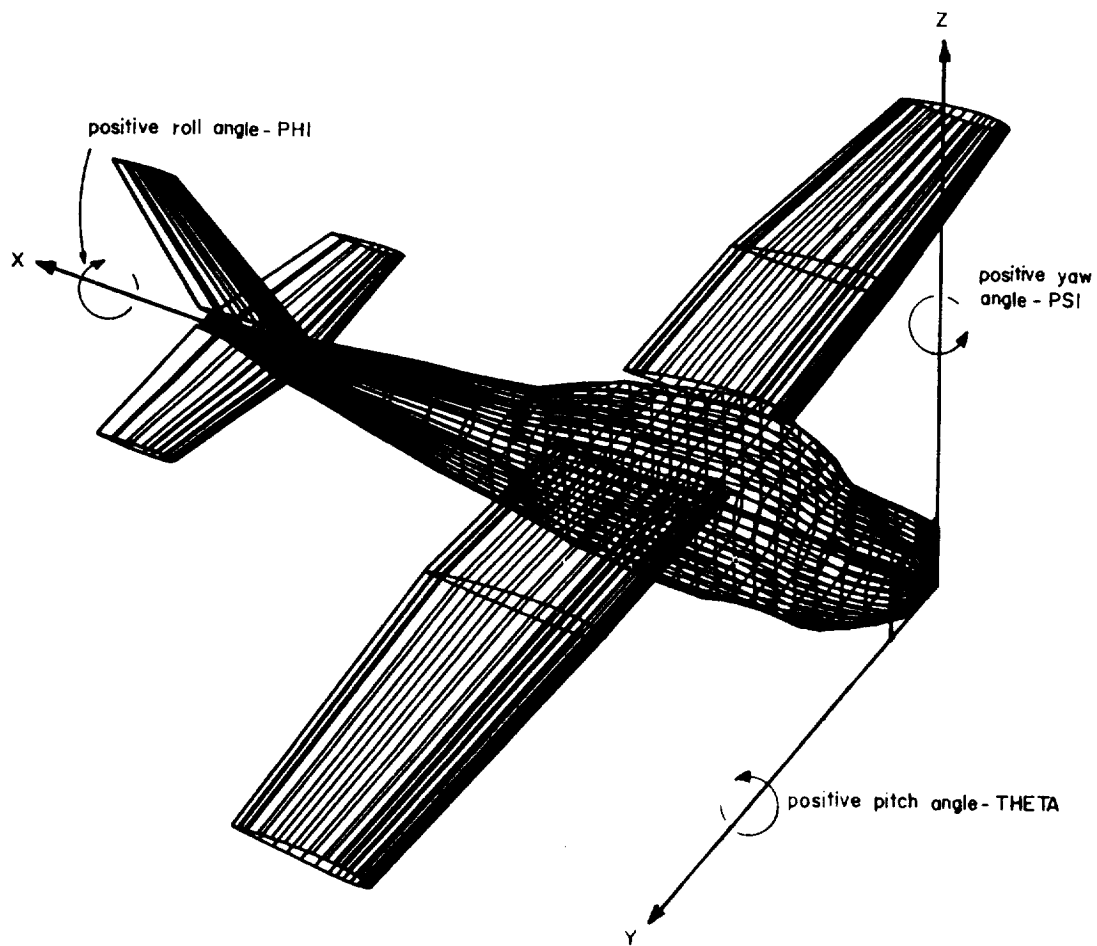
Figure D-1.   Orientation of body with respect to body reference
axes for the PLOT programs.

Figure D-2. Sample Cessna 182 data set for the PLOT program.

339
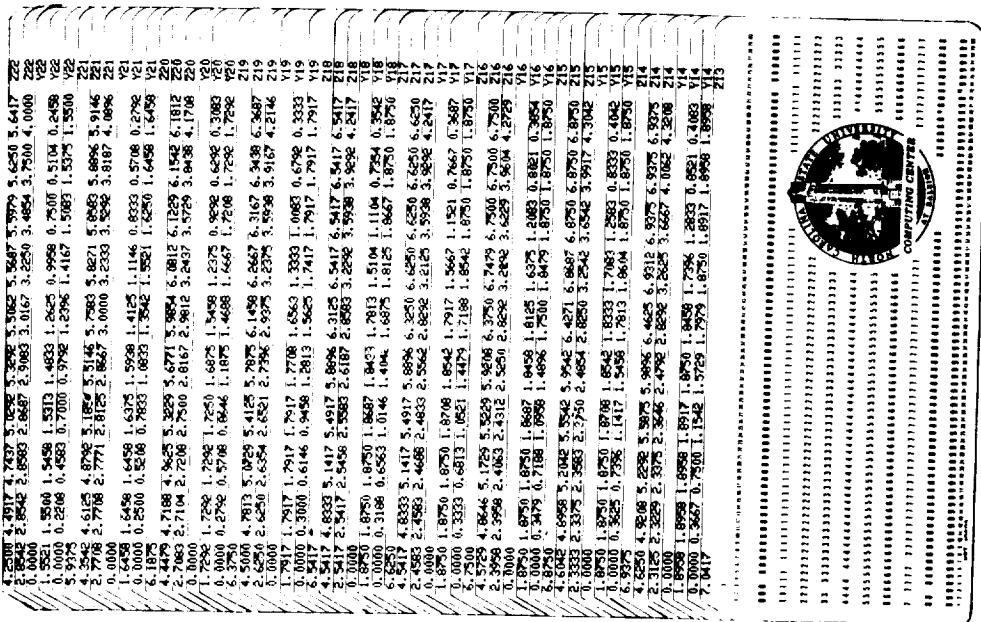
Figure D-2. Continued.

340
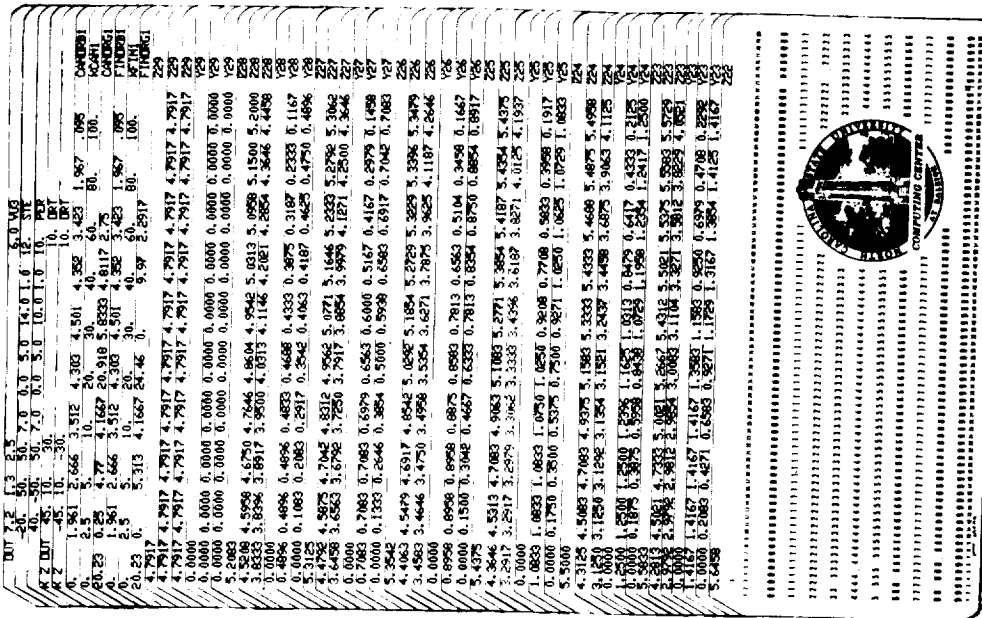
```
C00   1   C***  THIS PLOT PROGRAM WAS OBTAINED FROM NASA LANGLEY TO GIVE
C00   2   C***  N.C. STATE THE CAPABILITY TO GRAPHICALLY DISPLAY LIGHT AIRCRAFT.
C00   3   C***  THE ONLY MODIFICATIONS EMPLOYED WERE THOSE REQUIRED TO RUN THE
C00   4   C***  PROGRAM ON AN IBM COMPUTER.  AS A TOOL THIS PLOT PROGRAM IS
C00   5   C***  INVALUABLE FOR FINDING INPUT ERRORS IN OUR MODIFIED INVISCID BODY
C00   6   C***  PROGRAM.  WE WOULD LIKE TO THANK MR. CHARLES H. FOX JR. OF NASA
C00   7   C***  LANGLEY FOR HIS HELP IN OBTAINING THE PROGRAM.
C00   8   C***
C00   9   C***  REFERENCE:  DESCRIPTION OF A DIGITAL COMPUTER PROGRAM FOR AIRPLANE
C00  10   C***             CONFIGURATION PLOTS
C00  11   C***             CHARLOTTE B. CRAIDON
C00  12   C***             NASA TM X-2074
C00  13   C***             SEPTEMBER, 1970
C00  14   C***
C00  15   C***  AIRCRAFT CONFIGURATION PLOTS
C00  16         COMMON ABC(20),JO,J1,J2,J3,J4,J5,J6,NWAF,NWAFOR,NFUS,NRADX(4),NFOR
C00  17        1X(4),NP,NPODOR,NF,NFINOR,NCAN,NCANOR,JZTEST,NW,NC,ABCDE(20),HORZ,V
C00  18        1ERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG,PLOTSZ,TYPE,KODE,XMIN,X
C00  19        1MAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,ZMID,BIGD,ISP
C00  20         COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4
C00  21         JREAD=1
C00  22         JWRITE=3
C00  23         KFILE1=9
C00  24         KFILE2=10
C00  25         KFILE3=11
C00  26         KFILE4=12
C00  27   C***
C00  28   C***  IMPORTANT???
C00  29   C***
C00  30   C***  CALL PICSIZ(99.0,-11.0) IS THE APPROPRIATE CALL TO TURN THE PLOTTER
C00  31   C***  ON FOR THE N. C. STATE PLOTTING SOFTWARE.  THE CALL FOR THE
C00  32   C***  ORIGINAL CALCOMP PLOTTER SOFTWARE WAS CALL CALCOMP.  THE USER MUST
C00  33   C***  PROVIDE THE APPROPRIATE CALL FOR HIS INSTALLATION.
C00  34   C***
C00  35         CALL PICSIZ(99.0,-11.0)
C00  36         WRITE (JWRITE,5)
C00  37       5 FORMAT (1H1,17X4THPROGRAM D2290    PLOTS OF AIRCRAFT CONFIGURATION//
C00  38        1)
C00  39         DO 10 I=1,6
C00  40         WRITE (KFILE4)DUM
C00  41   C***         INPUT 1ST TWO CARDS
C00  42      15 FORMAT (20A4)
C00  43      20 READ (JREAD,15, END=55) ABC
C00  44         WRITE (JWRITE,30) ABC
C00  45      30 FORMAT (28X25HCONFIGURATION DESCRIPTION//1X20A4//)
C00  46         READ (JREAD,35) JO,J1,J2,J3,J4,J5,J6,NWAF,NWAFOR,NFUS,(NRADX(I),NF
C00  47        1ORX(I),I=1,4),NP,NPODOR,NF,NFINOR,NCAN,NCANOR
C00  48         WRITE (JWRITE,40) JO,J1,J2,J3,J4,J5,J6,NWAF,NWAFOR,NFUS,(NRADX(I),
C00  49        1NFORX(I),I=1,4),NP,NPODOR,NF,NFINOR,NCAN,NCANOR
C00  50      35 FORMAT (24I3)
C00  51      40 FORMAT (1X,24I3)
C00  52         CALL C8C10
C00  53   C***         INPUT CONFIGURATION DESCRIPTION AND INITIALIZE
C00  54   C***  PLOT CONFIGURATION
C00  55         WRITE (JWRITE,45)
C00  56      45 FORMAT (//36X9HPLOT DATA//)
C00  57      50 READ (JREAD,15, END=55) ABCDE
C00  58         GO TO 60
C00  59   C***  IMPORTANT???
C00  60   C***
```

```
C00  61   C***  CALL PICSIZ(0.0,0.0) IS THE APPROPRIATE CALL TO TURN THE PLOTTER
C00  62   C***  OFF FOR THE N. C. STATE PLOTTING SOFTWARE.  THE CALL FOR THE
C00  63   C***  ORIGINAL CALCOMP PLOTTER SOFTWARE WAS CALL CALPLT(0.,0.,999).
C00  64   C***  THE USER MUST PROVIDE THE APPROPRIATE CALL FOR HIS INSTALLATION.
C00  65   C***
C00  66      55 CALL PICSIZ(0.0,0.0)
C00  67         STOP
C00  68      60 WRITE (KFILE3,15) ABCDE
C00  69         REWIND KFILE3
C00  70         READ (KFILE3,65) HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG,
C00  71        1PLOTSZ,TYPE,KODE
C00  72         REWIND KFILE3
C00  73      65 FORMAT (2A2,A3,9F5.0,A3,16X,I1)
C00  74         WRITE (JWRITE,70) HORZ,VERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG
C00  75        1,PLOTSZ,TYPE,KODE
C00  76      70 FORMAT (1X,2A2,A3,9(1X,F10.5),A3,16X,I1)
C00  77         CALL C8C20
C00  78         IF (KODE.EQ.0) GO TO 50
C00  79         WRITE (JWRITE,75)
C00  80      75 FORMAT (1H1)
C00  81         GO TO 20
C00  82         END
```

```
C10   1         SUBROUTINE C8C10
C10   2   C***         INPUTS AND INITIALIZES CONFIGURATION DESCRIPTION
C10   3         COMMON ABC(20),JO,J1,J2,J3,J4,J5,J6,NWAF,NWAFOR,NFUS,NRADX(4),NFOR
C10   4        1X(4),NP,NPODOR,NF,NFINOR,NCAN,NCANOR,JZTEST,NW,NC,ABCDE(20),HORZ,V
C10   5        1ERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG,PLOTSZ,TYPE,KODE,XMIN,X
C10   6        1MAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,ZMID,BIGD,ISP
C10   7         COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4
C10   8         DIMENSION BLOCK(7500)
C10   9         DIMENSION XAF(30),WAFORG(20,4),WAFORD(20,3,30),TZORD(20,30)
C10  10         EQUIVALENCE (BLOCK,XAF),(BLOCK(131),WAFORG),(BLOCK(111),WAFORD),(BL
C10  11        1OCK(1911),TZORD)
C10  12         DIMENSION XFUS(30,4),ZFUS(30,4),FUSARD(30,4),SFUS(30,
C10  13        130,8)
C10  14         EQUIVALENCE (BLOCK,XFUS),(BLOCK(121),ZFUS),(BLOCK(241),FUSARD),(BL
C10  15        1OCK(361),FUSRAD),(BLOCK(241),SFUS)
C10  16         DIMENSION PODORG(9,3),XPOD(9,30),PODORD(9,30,30)
C10  17         EQUIVALENCE (BLOCK,PODORG),(BLOCK(28),XPOD),(BLOCK(298),PODORD),(B
C10  18        1LOCK(568),XPOD1)
C10  19         DIMENSION FINORG(6,2,4),XFIN(6,10),FINX2(6,2,10),FINX2(6,2,10),FI
C10  20        1N3(6,2,10)
C10  21         EQUIVALENCE (BLOCK,FINORG),(BLOCK(49),XFIN),(BLOCK(109),FINORD),(B
C10  22        1LOCK(229),FINX2),(BLOCK(349),FINX3)
C10  23         DIMENSION CANORG(2,2,4),XCAN(2,10),CANORD(2,2,10),CANOR1(2,2,10),C
C10  24        1ANORX(2,2,10)
C10  25         EQUIVALENCE (BLOCK,CANORG),(BLOCK(17),XCAN),(BLOCK(37),CANORD),(BL
C10  26        1OCK(77),CANOR1),(BLOCK(117),CANORX)
C10  27         DIMENSION ABCD(8)
C10  28         DIMENSION ALRT(31,3,2),VECRT(30,3),ANSIN(30),ANCOS(30)
C10  29         DATA NAN2/25/
C10  30         DATA PI/3.14159265/
C10  31         REWIND KFILE1
C10  32         REWIND KFILE2
C10  33         REWIND KFILE4
C10  34       5 FORMAT (8A10)
```

```
10 FORMAT (18A10)
15 FORMAT (10F7.0)
C***    REFERENCE AREA
C
      IF (J0.NE.2) GO TO 20
      READ (KFILE4)REFA
      WRITE (KFILE1)REFA
      GO TO 35
20    IF (J0.EQ.0) GO TO 30
      READ (JREAD,15) REFA
      WRITE (JWRITE,25) REFA
25    FORMAT (11X,10(1X,F10.5))
30    WRITE (KFILE1)REFA
      READ (KFILE4)DUM
C***    WING
35    IF (J1.NE.2) GO TO 40
      NWAFOR=IABS(NWAFOR)
      NW=NWAFOR
      READ (KFILE4)BLOCK
      WRITE (KFILE1)BLOCK
      GO TO 110
40    IF (J1.EQ.0) GO TO 205
      N=IABS(NWAFOR)
      NREC=(N+9)/10
      I1=-9
      I2=0
      DO 45 NN=1,NREC
      I1=I1+10
      I2=I2+10
      READ (JREAD,15) (XAF(I),I=I1,I7)
      WRITE (JWRITE,25) (XAF(I),I=I1,I12)
45    CONTINUE
      DO 50 I=1,NWAF
      READ (JREAD,15) (WAFCRG(I,J),J=1,4)
      WRITE (JWRITE,25) (WAFCRG(I,J),J=1,4)
50    CONTINUE
      IF (J1.LT.0) GO TO 65
      DO 60 NN=1,NWAF
      I1=-9
      I2=0
      DO 55 N1=1,NREC
      I1=I1+10
      I2=I2+10
      READ (JREAD,15) (TZORD(NN,I),I=I1,I12)
      WRITE (JWRITE,25) (TZORD(NN,I),I=I1,I12)
55    CONTINUE
60    CONTINUE
      GO TO 75
65    DO 70 I=1,NWAF
      DO 70 K=1,N
70    TZORD(I,K)=0.
75    L=1
      IF (NWAFOR.LT.OIL-2
      DO 85 NN=1,NWAF
      I1=-9
      I2=0
      DO 80 N1=1,NREC
      I1=I1+10
      I2=I2+10

      READ (JREAD,15) (WAFCRD(NN,K,I),I=I1,I12)
      WRITE (JWRITE,25) (WAFORD(NN,K,I),I=I1,I12)
80    CONTINUE
85    CONTINUE
      IF (NWAFOR.LT.0) GO TO 95
      DO 90 NN=1,NWAF
      DO 90 K=1,N
90    WAFORD(NN,2,K)=WAFORD(NN,1,K)
95    CONTINUE
      NWAFOR=IABS(NWAFOR)
      NW=NWAFOR
      J1=IABS(J1)
C***    CHANGE TO ACTUAL UNITS, COMPUTE MINIMUMS AND MAXIMUMS
      DO 105 I=1,NWAF
      E=.01*WAFORG(I,4)
      (3)=WAFORG(I,3)
      DO 100 J=1,NWAFOR
      WAFORD(I,1,J)=E*WAFORD(I,1,J)+E3*TZCRD(I,J)
      WAFORD(I,2,J)=E*WAFGRD(I,2,J)+E3*TZCRD(I,J)
100   WAFORD(I,3,J)=WAFORG(I,1)+E*XAF(J)
105   CONTINUE
110   XMIN=WAFORG(I,1)
      XMAX=WAFORG(I,1)
      YMAX=WAFORG(I,2)
      ZMIN=WAFORD(I,1,1)
      ZMAX=WAFORD(I,1,1)
      DO 120 N=1,NWAF
      XMAX=AMAX1(XMAX,WAFORD(N,3,NW))
      XMIN=AMIN1(XMIN,WAFORD(N,3,1))
      YMAX=AMAX1(YMAX,WAFORG(N,2))
      DO 115 NN=1,NW
      ZMAX=AMAX1(ZMAX,WAFORD(N,1,NN))
      ZMIN=AMIN1(ZMIN,WAFORD(N,2,NN))
115   CONTINUE
120   CONTINUE
      NL=NW-1
C***    SETUP 1ST LINE IN STREAMWISE DIRECTION
      DO 160 I=1,2
      DO 125 N=1,NW
      ALRT(N,1,2)=WAFORD(I,3,N)
      ALRT(N,2,2)=WAFORG(IN,2)
      ALRT(N,3,2)=WAFORD(I,1,N)
125   CONTINUE
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,NW),M3=1,3)
      DO 130 N=1,NW
130   ALRT(N,3,1)=ALRT(N,M3,2)
      DO 135 N=1,NW
      ALRT(N,1,2)=WAFORD(IN,3,N)
      ALRT(N,2,2)=WAFGRG(IN,2)
135   ALRT(N,3,2)=WAFCRD(IN,1,N)
      GO TO (140,145),L
140   CALL SURCC(NW,ALRT,VECRT)
      GO TO 150
145   CALL SURCL(NW,ALRT,VECRT)
150   CONTINUE
      WRITE (KFILE2)((VECRT(N,M3),N=1,NL),M3=1,3)
```

342

```
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,NM),M3=1,3)        C10 155
155   CONTINUE                                            C10 156
160   CONTINUE                                            C10 157
      NL=NNAF-1                                           C10 158
      DO 2CO I=1,2                                        C10 159
      DO 165 N=1,NNAF                                     C10 160
C***     SETUP 1ST LINE IN SPANWISE DIRECTION             C10 161
      ALRT(N,1,2)=NAFORD(N,3,2)                           C10 162
      ALRT(N,2,2)=NAFORG(N,2)                             C10 163
      ALRT(N,3,2)=NAFORD(N,1,1)                           C10 164
165   CONTINUE                                            C10 165
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,NNAF),M3=1,3)      C10 166
      DO 195 NM=7,NM                                      C10 167
      DO 170 N=1,NNAF                                     C10 168
      DO 170 M3=1,3                                       C10 169
      ALRT(N,M3,1)=ALRT(N,M3,2)                           C10 170
170   CONTINUE                                            C10 171
      DO 175 N=1,NNAF                                     C10 172
      ALRT(N,1,2)=NAFORD(N,3,NM)                          C10 173
      ALRT(N,2,2)=NAFORG(N,2)                             C10 174
      ALRT(N,3,2)=NAFORD(N,1,NM)                          C10 175
175   CONTINUE                                            C10 176
      GO TO (180,185),I                                   C10 177
180   CALL SURCL(NNAF,ALRT,VECRT)                         C10 178
      GO TO 190                                           C10 179
185   CALL SURCC(NNAF,ALRT,VECRT)                         C10 180
190   CONTINUE                                            C10 181
      WRITE (KFILE2)((VECRT(A,N3),N=1,NL(1,N3=1,3)        C10 182
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,NNAF),M3=1,3)      C10 183
195   CONTINUE                                            C10 184
200   CONTINUE                                            C10 185
      IF (J1.EQ.2) GO TO 210                              C10 186
205   WRITE (KFILE1)BLOCK                                 C10 187
      READ (KFILE4)DUM                                    C10 188
C***     FUSELAGE                                         C10 189
210   IF (J2.NE.2) GO TO 215                              C10 190
      READ (KFILE4)BLOCK                                  C10 191
      WRITE (KFILE1)BLOCK                                 C10 192
      GO TO 20C                                           C10 193
215   IF (J2.EQ.0) GO TO 425                              C10 194
      J2TEST=3                                            C10 195
      IF (J2.EQ.-1.AND.J6.EQ.-1)J2TEST=1                  C10 196
      IF (J2.EQ.-1.AND.J6.EQ.0)J2TEST=2                   C10 197
      IF (J6.EQ.1)J2TEST=1                                C10 198
      J2=1                                                C10 199
      DO 275 NFU=1,NFUS                                   C10 200
      NRAD=NRADX(NFU)                                     C10 201
      NFUSOR=NFORX(NFU)                                   C10 202
      N=NFUSOR                                            C10 203
      NREC=(N+9)/10                                       C10 204
      I1=-9                                               C10 205
      I2=0                                                C10 206
      DO 220 N1=1,NREC                                    C10 207
      I1=I1+10                                            C10 208
      I2=I2+10                                            C10 209
      READ (JREAD,15) (KFUS(I,NFU),I=I1,I2)               C10 210
      WRITE (JWRITE,25) (KFUS(I,NFU),I=I1,I2)             C10 211
220   CONTINUE                                            C10 212
      IF (J2TEST.NE.2) GO TO 230                          C10 213
      I1=-9                                               C10 214

      I2=0                                                C10 215
      DO 225 N1=1,NREC                                    C10 216
      I1=I1+10                                            C10 217
      I2=I2+10                                            C10 218
      READ (JREAD,15) (ZFUS(I,NFU),I=I1,I2)               C10 219
      WRITE (JWRITE,25) (ZFUS(I,NFU),I=I1,I2)             C10 220
225   CONTINUE                                            C10 221
      GO TO 240                                           C10 222
230   DO 235 I=1,M                                        C10 223
235   ZFUS(I,NFU)=0.                                      C10 224
240   IF (J2TEST.NE.3) GO TO 260                          C10 225
      NCARD=(NRAD+9)/10                                   C10 226
      DO 250 I=1,M                                        C10 227
      DO 250 M=1,2                                        C10 228
      KK=K+(NFU-1)*2                                      C10 229
245   CONTINUE                                            C10 230
      I1=-10                                              C10 231
      I2=0                                                C10 232
      DO 245 N1=1,NCARD                                   C10 233
      IF (NN.EC.NCARD)I1=MCD(NRAD,10)                     C10 234
      IF (I1.EQ.0)I1=10                                   C10 235
      I1=I1+10                                            C10 236
      I2=I2+11                                            C10 237
      READ (JREAD,15) (SFUS(I,LN,KK),I=I1,I2)             C10 238
      WRITE (JWRITE,25) (SFUS(I,LN,KK),I=I1,I2)           C10 239
245   CONTINUE                                            C10 240
250   CONTINUE                                            C10 241
255   CONTINUE                                            C10 242
      GO TO 275                                           C10 243
260   I1=-9                                               C10 244
      I2=0                                                C10 245
      DO 265 N1=1,NREC                                    C10 246
      I1=I1+10                                            C10 247
      I2=I2+10                                            C10 248
      READ (JREAD,15) (FUSARD(I,NFU),I=I1,I2)             C10 249
      WRITE (JWRITE,25) (FUSARD(I,NFU),I=I1,I2)           C10 250
265   CONTINUE                                            C10 251
      DO 270 I=1,M                                        C10 252
270   FUSRAD(I,NFU)=SQRT(FUSARD(I,NFU)/PI)                C10 253
275   CONTINUE                                            C10 254
C***     FUSELAGE MIN AND MAX                             C10 255
280   IF (J1.NE.0) GO TO 290                              C10 256
      XMIN=KFUS(1,1)                                      C10 257
      XMAX=KFUS(1,1)                                      C10 258
      IF (J2TEST.EQ.3) GO TO 285                          C10 259
      YMIN=FUSRAD(1,1)                                    C10 260
      ZMIN=-FUSRAD(1,1)+ZFUS(1,1)                         C10 261
      ZMAX=FUSRAD(1,1)+ZFUS(1,1)                          C10 262
      GO TO 290                                           C10 263
285   YMAX=SFUS(1,1,1)                                    C10 264
      ZMIN=SFUS(1,1,2)                                    C10 265
      ZMAX=SFUS(1,1,2)                                    C10 266
290   DO 310 N=1,NFUS                                     C10 267
      NRAD=NRADX(N)                                       C10 268
      NFUSOR=NFORX(N)                                     C10 269
      XMIN=AMIN1(XMIN,KFUS(1,N))                          C10 270
      XMAX=AMAX1(XMAX,XFUS(NFUSOR,N))                     C10 271
      DO 305 NN=1,NFUSOR                                  C10 272
      IF (J2TEST.EQ.3) GO TO 295                          C10 273
      YMAX=AMAX1(YMAX,FUSRAD(NN,N))                       C10 274
```

343

```
      NL1=NAN-1
      IF (J2TEST.EQ.3) GO TO 375
      FANG=(MRAD-1)*2
      DELE=6.2831853/FANG
      DO 370 M=1,NAN
      E=M-1
      ANSIN(N)=SIN(E*DELE+4.7123891)
  370 ANCOS(N)=COS(E*DELE+4.7123891)
  375 CONTINUE
      KK=1+(NFU-1)*2
      DO 390 M=1,NAN
      ALRT(M,1,2)=XFUS(1,NFU)
      IF (J2TEST.EQ.3) GO TO 380
      ALRT(M,2,2)=FUSRAD(1,NFU)*ANCOS(M)
      ALRT(M,3,2)=FUSRAD(1,NFU)*ANSIN(M)+ZFUS(1,NFU)
      GO TO 385
  380 ALRT(M,2,2)=SFUS(M,1,KK)
      ALRT(M,3,2)=SFUS(M,1,KK+1)
  385 CONTINUE
  390 CONTINUE
      WRITE (KFILE2)((ALRT(M,N3,2),M=1,NAN),N3=1,3)
      DO 415 MM=2,NFUSOR
      DO 395 M=1,NAN
      DO 395 M3=1,3
  395 ALRT(M,M3,1)=ALRT(M,M3,2)
      DO 410 M=1,NAN
      ALRT(M,1,2)=XFUS(MM,NFU)
      IF (J2TEST.EQ.3) GO TO 400
      ALRT(M,2,2)=FUSRAD(MM,NFU)*ANCOS(M)
      ALRT(M,3,2)=FUSRAD(MM,NFU)*ANSIN(M)+ZFUS(MM,NFU)
      GO TO 405
  400 ALRT(M,2,2)=SFUS(M,MM,KK)
      ALRT(M,3,2)=SFUS(M,MM,KK+1)
  405 CONTINUE
  410 CONTINUE
      CALL SURCC(NAN,ALRT,VECRT)
      WRITE (KFILE2)((VECRT(M,N3),M=1,NL1),N3=1,3)
      WRITE (KFILE2)((ALRT(M,N3,2),M=1,NAN),N3=1,3)
  415 CONTINUE
  420 CONTINUE
      IF (J2.EQ.2) GO TO 430
      WRITE (KFILE4)IBLOCK
      READ (KFILE4)DUM
C***  NACELLES
  430 IF (J3.NE.2) GO TO 435
      READ (KFILE4)IBLOCK
      WRITE (KFILE1)IBLOCK
      GO TO 455
  435 IF (J3.EQ.0) GO TO 535
      NMPODOR
      NREC=(N+9)/10
      DO 450 NM=1,NP
      READ (JREAD,15) (PODORG(NM,I),I=1,3)
      WRITE (JWRITE,25) (PODORG(NM,I),I=1,3)
      I1=-9
      I2=0
      DO 440 N1=1,NREC
      I1=I1+10
      I2=I2+10
```

```
      ZMAX=AMAX1(ZMAX,FUSRAD(MM,N)+ZFUS(MM,N))
      ZMIN=AMIN1(ZMIN,-FUSRAD(MM,N)+ZFUS(MM,N))
  295 KK=1+(M-1)*2
      DO 300 M8=1,MRAD
      YMAX=AMAX1(YMAX,SFUS(MM,MM,KK))
      YMIN=AMIN1(ZMIN,SFUS(MM,MM,KK+1))
      ZMAX=AMAX1(ZMAX,SFUS(MM,MM,KK+1))
  300 CONTINUE
  305 CONTINUE
  310 CONTINUE
C***  SETUP 1ST LINE IN STREAMWISE DIRECTION
      DO 365 NFU=1,NFUS
      MRAD=MRADK(NFU)
      NFUSOR=NFORK(NFU)
      NL1=NFUSOR-1
      NAN=MRAD
      IF (J2TEST.EQ.3) GO TO 320
      FANG=(MRAD-1)*2
      DELE=6.2831853/FANG
      DO 315 M=1,NAN
      E=M-1
  315 ANSIN(N)=SIN(E*DELE+4.7123891)
      ANCOS(N)=COS(E*DELE+4.7123891)
  320 CONTINUE
      KK=1+(NFU-1)*2
      DO 335 M=1,NFUSOR
      IF (J2TEST.EQ.3) GO TO 325
      ALRT(M,2,2)=FUSRAD(M,NFU)*ANCOS(1)
      ALRT(M,3,2)=FUSRAD(M,NFU)*ANSIN(1)+ZFUS(M,NFU)
      GO TO 330
  325 ALRT(M,2,2)=SFUS(1,M,KK)
      ALRT(M,3,2)=SFUS(1,M,KK+1)
  330 CONTINUE
  335 CONTINUE
      DO 360 MM=2,NAN
      DO 340 M=1,NFUSOR
      DO 340 M3=1,3
  340 ALRT(M,M3,1)=ALRT(M,M3,2)
      DO 355 M=1,NFUSOR
      IF (J2TEST.EQ.3) GO TO 345
      ALRT(M,2,2)=FUSRAD(M,NFU)*ANCOS(MM)
      ALRT(M,3,2)=FUSRAD(M,NFU)*ANSIN(MM)+ZFUS(M,NFU)
      GO TO 350
  345 ALRT(M,2,2)=SFUS(MM,M,KK)
      ALRT(M,3,2)=SFUS(MM,M,KK+1)
  350 CONTINUE
  355 CONTINUE
      CALL SURCL(NFUSOR,ALRT,VECRT)
      WRITE (KFILE2)((VECRT(M,N3,2),M=1,NL1),N3=1,3)
      WRITE (KFILE2)((ALRT(M,N3,2),M=1,NFUSOR),N3=1,3)
  360 CONTINUE
  365 CONTINUE
C***  SETUP 1ST LINE AROUND BODY
      DO 420 NFU=1,NFUS
      MRAD=MRADK(NFU)
      NFUSOR=NFORK(NFU)
      NAN=MRAD
```

344

```
      READ (JREAD,15) (XPOD(INN,I),I=I1,I2)
      WRITE (JWRITE,25) (XPOD(INN,I),I=I1,I2)
440   CONTINUE
      I1=9
      I2=0
      DO 445 N1=1,NREC
      I1=I1+10
      I2=I2+10
      READ (JREAD,15) (PODORD(INN,I),I=I1,I2)
      WRITE (JWRITE,25) (PODORD(INN,I),I=I1,I2)
445   CONTINUE
450   CONTINUE
C***  COMPUTE ACTUAL X,MINIMUM,MAXIMUM
455   DO 460 N=1,NP
      NL1=NPODOR
460   XPOD1(N,NN)=XPOD1(N,NN)+PODORG(N,1)
      IF (J1.NE.0.OR.J2.NE.0) GO TO 465
      XMIN=XPOD1(1,1)
      XMAX=XPOD1(1,1),NPODOR)
      ZMIN=PODORG(1,2)+PODORD(1,1)
      ZMAX=PODORG(1,3)+PODORD(1,1)
465   DO 475 N=1,NP
      XMIN=AMIN1(XMIN,XPOD1(N,1))
      DO 470 NN=1,NPODOR
      XMAX=AMAX1(XMAX,PODORD(N,NN)+PODORG(N,2))
      YMAX=AMAX1(YMAX,PODORD(N,NN)+PODORG(N,2))
      ZMIN=AMIN1(ZMIN,PODORG(N,3)+PODORD(N,NN))
      ZMAX=AMAX1(ZMAX,PODORG(N,3)+PODORD(N,NN))
470   CONTINUE
475   CONTINUE
      NANG1=NAN2+1
      FANG=NAN2
      DELE=6.2831853/FANG
      E=0.1
      ANSIN(N)=SIN(E)+DELE)
480   ANCOS(N)=COS(E)+DELE)
C***  SETUP 1ST LINE IN STREAMWISE DIRECTION
      DO 505 NP1=1,NP
      DO 485 N=1,NPODOR
      ALRT(N,1,2)=XPOD(NP1,N)+PODORG(NP1,1)
      ALRT(N,2,2)=PODORD(NP1,N)+ANCOS(NN)+PODORG(NP1,2)
      ALRT(N,3,2)=PODORD(NP1,N)+ANSIN(NN)+PODORG(NP1,3)
485   CONTINUE
      CALL SURCL(NPODOR,ALRT,VECRT)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NPODOR),N3=1,3)
      DO 500 NN=2,NANG1
      DO 490 N=1,NPODOR
490   ALRT(N,1,1)=ALRT(N,N3,2)
      DO 495 N=1,NPODOR
      ALRT(N,2,2)=PODORD(NP1,N)+ANCOS(NN)+PODORG(NP1,2)
      ALRT(N,3,2)=PODORD(NP1,N)+ANSIN(NN)+PODORG(NP1,3)
495   CONTINUE
      CALL SURCL(NPODOR,ALRT,VECRT)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NPODOR),N3=1,3)
500   CONTINUE
505   CONTINUE
```

```
C***  SETUP 1ST LINE AROUND PODS
      NL1=NANG1-1
      DO 530 NP1=1,NP
      DO 510 N=1,NANG1
      N=N
      ALRT(N,1,2)=XPOD(NP1,1)+PODORG(NP1,1)
      ALRT(N,2,2)=PODORD(NP1,1)+ANCOS(N)+PODORG(NP1,2)
      ALRT(N,3,2)=PODORD(NP1,1)+ANSIN(N)+PODORG(NP1,3)
510   CONTINUE
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NANG1),N3=1,3)
      DO 525 NN=2,NPODOR
      DO 515 N=1,NANG1
      DO 515 N3=1,3
      ALRT(N3,1)=ALRT(N,N3,2)
515   CONTINUE
      DO 520 N=1,NANG1
      N=N
      ALRT(N,1,2)=XPOD(NP1,NN)+PODORG(NP1,1)
      ALRT(N,2,2)=PODORD(NP1,NN)+ANCOS(N)+PODORG(NP1,2)
      ALRT(N,3,2)=PODORD(NP1,NN)+ANSIN(N)+PODORG(NP1,3)
520   CONTINUE
      CALL SURCC(NANG1,ALRT,VECRT)
      WRITE (KFILE2)((VECRT(N,N3),N=1,NL1),N3=1,3)
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,NANG1),N3=1,3)
525   CONTINUE
530   CONTINUE
      IF (J3.EQ.2) GO TO 540
      WRITE (KFILE1)BLOCK
      READ (KFILE4)DUM
C***  FINS
540   IF (J4.NE.2) GO TO 545
      READ (KFILE4)BLOCK
      WRITE (KFILE1)BLOCK
      GO TO 565
545   IF (J4.EQ.0) GO TO 660
      N=NFINOR
      DO 550 NN=1,NP
      READ (JREAD,15) (FINORG(NN,I,J),J=1,4),I=1,2)
      WRITE (JWRITE,25) (FINORG(NN,I,J),J=1,4),I=1,2)
      READ (JREAD,15) (XFIN(NN,I),I=1,N)
      READ (JREAD,15) (FINORD(NN,I,J),J=1,N)
      WRITE (JWRITE,25) (FINORD(NN,I,J),J=1,N)
550   CONTINUE
C***  CHANGE TO ACTUAL UNITS, COMPUTE MINIMUMS AND MAXIMUMS
      DO 560 LQ=1,NP
      DO 560 I=1,2
      J=3-I
      E=0.1(FINORG(LQ,J,4)
      E2=FINORG(LQ,J,2)
      DO 555 K=1,NFINOR
      EE=FINORD(LQ,1,K)+E
      FINORD(LQ,1,K)=E2+EE
      FIN2Z(LQ,J,K)=E2-EE
      FIN3S(LQ,J,K)=FINORG(LQ,J,1)+XFIN(INLQ,K)
555   CONTINUE
560   CONTINUE
565   IF (J1.NE.0.OR.J2.NE.0.OR.J3.NE.0) GO TO 570
      XMIN=FINORG(1,1,1)
      XMAX=FINORG(1,1,1)
      YMAX=FINORG(1,1,2)
```

```
      ZMIN=FINORG(I,1,3)
      ZMAX=FINORG(I,1,3)
      ZMIN=AMIN1(ZMIN,FINORG(N,1,3))
      ZMAX=AMAX1(ZMAX,FINORG(N,2,3))
  570 DO 575 N=1,NF
      DO 575 N2=1,2
      XMIN=AMIN1(XMIN,FINORG(N,N2,1))
      XMAX=AMAX1(XMAX,FINORG(N,N2,1))
      YMIN=AMIN1(YMIN,FINORD(N,N2,NN))
      YMAX=AMAX1(YMAX,FINORD(N,N2,NN))
  575 CONTINUE
      NL1=NFINOR-1
C***      SETUP LOWER AND UPPER LINES IN STREAMWISE DIRECTION
      DO 590 NF1=1,NF
      DO 580 N=1,NFINOR
      ALRT(N,1,N2)=FINX3(NF1,N2,N)
      ALRT(N,2,N2)=FINORD(NF1,N2,N)
      ALRT(N,3,N2)=FINORG(NF1,N2,3)
  580 CONTINUE
      CALL SURCL(NFINOR,ALRT,VECRT)
      WRITE (KFILE2)((ALRT(N,M3,1),N=1,NFINOR),M3=1,3)
      WRITE (KFILE2)(VECRT(N,M3),N=1,NL1,M3=1,3)
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,NFINOR),M3=1,3)
C***      CHANGE Y FOR INSIDE LINES
      DO 585 N2=1,2
      DO 585 N=1,NFINOR
      ALRT(N,2,N2)=FINX2(NF1,N2,N)
  585 CONTINUE
      CALL SURCC(NFINOR,ALRT,VECRT)
      WRITE (KFILE2)((ALRT(N,M3,1),N=1,NFINOR),M3=1,3)
      WRITE (KFILE2)(VECRT(N,M3),N=1,NL1,M3=1,3)
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,NFINOR),M3=1,3)
  590 CONTINUE
C***      SETUP LINES IN VERTICAL DIRECTION
      DO 615 NF1=1,NF
      DO 610 NN2=1,2
      DO 605 N2=1,2
      ALRT(N2,1,2)=FINX3(NF1,N2,1)
      IF (NN2.EQ.2) GO TO 595
      ALRT(N2,2,2)=FINORD(NF1,N2,1)
      GO TO 600
  595 ALRT(N2,2,2)=FINX2(NF1,N2,1)
  600 CONTINUE
      ALRT(N2,3,2)=FINORG(NF1,N2,3)
  605 CONTINUE
      WRITE (KFILE2)((ALRT(N,M3,2),N=1,2),M3=1,3)
      DO 610 NN=2,NFINOR
  610 CONTINUE
      ALRT(N2,3,1)=ALRT(N2,3,2)
  615 ALRT(N2,2,2)=FINX2(NF1,N2,1)
  620 CONTINUE
      ALRT(N2,3,2)=FINORG(NF1,N2,3)
```

<!-- page -->

```
  625 CONTINUE
      GO TO (630,635),MM2
  630 CALL SURCC(I2,ALRT,VECRT)
      GO TO 640
  635 CALL SURCL(I2,ALRT,VECRT)
  640 CONTINUE
      WRITE (KFILE2)(VECRT(I,N3),N3=1,3)
      WRITE (KFILE2)((ALRT(N2,N3,2),N2=1,2),N3=1,3)
  645 CONTINUE
  650 CONTINUE
  655 CONTINUE
      IF (J4.EQ.2) GO TO 665
  660 WRITE (KFILE1)BLOCK
      READ (KFILE4)DUR
C***      CANARDS
  665 IF (J5.NE.2) GO TO 670
      NCANR=IABS(NCANOR)
      NC=NCANOR
      READ (KFILE4)BLOCK
      WRITE (KFILE1)BLOCK
      GO TO 705
  670 IF (J5.EQ.0) GO TO 820
      N=IABS(NCANOR)
      DO 685 NN=1,NCAN
      READ (JREAD,15) ((CANORG(NN,I,J),J=1,4),I=1,2)
      WRITE (JWRITE,25) ((CANORG(I,J),J=1,4),I=1,2)
      READ (JREAD,15) (XCAN(NN,I),I=1,N)
      WRITE (JWRITE,25) (CANORD(NN,I,J),J=1,N)
      READ (JREAD,15) (CANORD(NN,I,J),J=1,N)
      IF (NCANOR.LT.0) GO TO 680
      DO 675 J=1,N
  675 CANOR1(NN,I,J)=CANORD(NN,I,J)
      GO TO 685
  680 READ (JREAD,15) (CANOR1(NN,I,J),J=1,N)
      WRITE (JWRITE,25) (CANOR1(NN,I,J),J=1,N)
  685 CONTINUE
      NCANR=IABS(NCANOR)
      NC=NCANOR
C***      CHANGE TO ACTUAL UNITS, COMPUTE MINIMUMS AND MAXIMUMS
      DO 700 NN=1,NCAN
      DO 695 K=1,2
      I=3-K
      E=.01*CANORG(NN,I,4)
      E3=CANORG(NN,1,3)
      DO 690 J=1,NCANOR
      CANORD(NN,I,J)=E*CANORD(NN,I,J)*E3
  690 CANOR1(NN,I,J)=E*CANOR1(NN,I,J)*E3
      CANORG(NN,I,J)=CANORG(NN,I,J)*E3
  695 CONTINUE
  700 CONTINUE
  705 IF (J1.NE.0.OR.J2.NE.0.OR.J3.NE.0.OR.J4.NE.0) GO TO 710
      XMIN=CANOR1(I,1,1)
      XMAX=CANOR1(I,1,NCANOR)
      YMIN=CANORG(I,1,1)
      YMAX=AMAX1(YMAX,CANORG(NCA,2,2))
  710 DO 725 NC=1,NCAN
      YMAX=AMAX1(YMAX,CANORG(NF1,N2,3))
      DO 720 N2=1,2
```

```
      XMIN=AMIN1(XMIN,CAMORX(MCA,N2,1))                              C10 635
      XMAX=AMAX1(XMAX,CAMORX(MCA,N2,MCAMOR))                         C10 636
      DO 715 NN=1,MCAMOR                                            C10 637
      ZMIN=AMIN1(ZMIN,CAMORD(MCA,N2,NN1)                            C10 638
      ZMAX=AMAX1(ZMAX,CAMORD(MCA,N2,NN1)                            C10 639
715   CONTINUE                                                     C10 640
720   CONTINUE                                                     C10 641
725   CONTINUE                                                     C10 642
      ML1=MC-1                                                      C10 643
C***  SETUP TWO LINES IN STREAMWISE DIRECTION FOR UPPER AND LOWER   C10 644
      DO 760 MCA=1,MCAN                                            C10 645
      DO 755 I=1,2                                                 C10 646
      DO 735 N2=1,2                                                C10 647
      DO 735 N=1,MC                                                C10 648
      ALRT(N,1,N2)=CAMORX(MCA,N2,N)                                C10 649
      ALRT(N,2,N2)=CAMORG(MCA,N2,2)                                C10 650
      IF (I.EQ.2) GO TO 730                                        C10 651
      ALRT(N,3,N2)=CAMORD(MCA,N2,N)                                C10 652
      GO TO 735                                                    C10 653
730   ALRT(N,3,N2)=CAMOR1(MCA,N2,N)                                C10 654
735   CONTINUE                                                     C10 655
      DO 740,745),I                                                C10 656
740   CALL SURCC(MC,ALRT,VECRT)                                    C10 657
      GO TO 750                                                    C10 658
745   CALL SURCL(MC,ALRT,VECRT)                                    C10 659
750   CONTINUE                                                     C10 660
      WRITE (KFILE2)((ALRT(N,1),N=1,MC),M3=1,3)                    C10 661
      WRITE (KFILE2)((VECRT(N,N3),N=1,NL),N3=1,3)                  C10 662
      WRITE (KFILE2)((ALRT(N,N3,2),N=1,MC),N3=1,3)                 C10 663
755   CONTINUE                                                     C10 664
760   CONTINUE                                                     C10 665
C***  SETUP LINES IN SPANWISE DIRECTION                            C10 666
      DO 815 MCA=1,MCAN                                            C10 667
      DO 810 I=1,2                                                 C10 668
      DO 770 N2=1,2                                                C10 669
      ALRT(N2,1,2)=CAMORX(MCA,N2,1)                                C10 670
      ALRT(N2,2,2)=CAMORG(MCA,N2,2)                                C10 671
      IF (I.EQ.2) GO TO 765                                        C10 672
      ALRT(N2,3,2)=CAMORD(MCA,N2,1)                                C10 673
      GO TO 770                                                    C10 674
765   ALRT(N2,3,2)=CAMOR1(MCA,N2,1)                                C10 675
770   WRITE (KFILE2)((ALRT(N2,M3,2),N2=1,2),N3=1,3)               C10 676
      DO 805 NN=2,MC                                               C10 677
      DO 775 N3=1,3                                                C10 678
      DO 775 N2=1,2                                                C10 679
      ALRT(N2,N3,1)=ALRT(N2,N3,2)                                  C10 680
775   CONTINUE                                                     C10 681
      DO 785 N2=1,2                                                C10 682
      ALRT(N2,1,2)=CAMORX(MCA,N2,NN)                               C10 683
      ALRT(N2,2,2)=CAMORG(MCA,N2,2)                                C10 684
      IF (I.EQ.2) GO TO 780                                        C10 685
      ALRT(N2,3,2)=CAMORD(MCA,N2,NN)                               C10 686
      GO TO 785                                                    C10 687
780   ALRT(N2,3,2)=CAMOR1(MCA,N2,NN)                               C10 688
785   CONTINUE                                                     C10 689
      GO TO (790,795),I                                            C10 690
790   CALL SURCL(2,ALRT,VECRT)                                     C10 691
      GO TO 800                                                    C10 692
795   CALL SURCC(2,ALRT,VECRT)                                     C10 693
800   CONTINUE                                                     C10 694

      WRITE (KFILE2)(VECRT(1,M3),N3=1,3)                           C10 695
      WRITE (KFILE2)((ALRT(N2,N3,2),N2=1,2),N3=1,3)               C10 696
805   CONTINUE                                                     C10 697
810   CONTINUE                                                     C10 698
815   CONTINUE                                                     C10 699
      IF (J5.EQ.2) GO TO 825                                       C10 700
820   WRITE (KFILE1)BLOCK                                          C10 701
      REWIND KFILE4                                                C10 702
      READ (KFILE4)DUM                                             C10 703
      REWIND KFILE4                                                C10 704
      WRITE (KFILE4)REFA                                           C10 705
825   REWIND KFILE1                                                C10 706
      READ (KFILE1)BLOCK                                           C10 707
      DO 830 K=1,5                                                 C10 708
      WRITE (KFILE4)BLOCK                                          C10 709
830   WRITE (KFILE1)BLOCK                                          C10 710
C***  FIND MAXIMUM DISTANCE AND MIDPOINT                           C10 711
      YMIN=YMAX                                                    C10 712
      XDIS=XMAX-XMIN                                               C10 713
      YDIS=YMAX-YMIN                                               C10 714
      ZDIS=ZMAX-ZMIN                                               C10 715
      BIGD=AMAX1(XDIS,YDIS,ZDIS)                                   C10 716
      XMID=.5*(XMAX-XMIN)+XMIN                                     C10 717
      YMID=0.                                                      C10 718
      ZMID=.5*(ZMAX-ZMIN)+ZMIN                                     C10 719
      RETURN                                                       C10 720
      END

      SUBROUTINE SURCL(NPT,FLINE,FVEC)                             SCL  1
C***  COMPUTES SURFACE UNIT NORMALS                                SCL  2
      DIMENSION FLINE(31,3,2),FVEC(30,3)                           SCL  3
      DO 10 N=2,NPT                                                SCL  4
      T1X=FLINE(N,1,2)-FLINE(N-1,1,1)                              SCL  5
      T2X=FLINE(N-1,1,2)-FLINE(N,1,1)                              SCL  6
      T1Y=FLINE(N,2,2)-FLINE(N-1,2,1)                              SCL  7
      T2Y=FLINE(N-1,2,2)-FLINE(N,2,1)                              SCL  8
      T1Z=FLINE(N,3,2)-FLINE(N-1,3,1)                              SCL  9
      T2Z=FLINE(N-1,3,2)-FLINE(N,3,1)                              SCL 10
      XNX=T2Y*T1Z-T1Y*T2Z                                          SCL 11
      YNY=T1X*T2Z-T2X*T1Z                                          SCL 12
      ZNZ=T2X*T1Y-T1X*T2Y                                          SCL 13
      FN=SQRT(XNX**2+YNY**2+ZNZ**2)                                SCL 14
      IF (FN.EQ.0.) GO TO 5                                        SCL 15
      FVEC(N-1,1)=XNX/FN                                           SCL 16
      FVEC(N-1,2)=YNY/FN                                           SCL 17
      FVEC(N-1,3)=ZNZ/FN                                           SCL 18
      GO TO 10                                                     SCL 19
5     FVEC(N-1,1)=0.                                               SCL 20
      FVEC(N-1,2)=0.                                               SCL 21
      FVEC(N-1,3)=0.                                               SCL 22
10    CONTINUE                                                     SCL 23
      RETURN                                                       SCL 24
      END                                                          SCL 25

      SUBROUTINE SURCC(NPT,FLINE,FVEC)                             SCC  1
C***  COMPUTES SURFACE UNIT NORMALS                                SCC  2
      DIMENSION FLINE(31,3,2),FVEC(30,3)                           SCC  3
```

347

```
      DO 10 N=2,NPT                                               SCC   4
      T1X=FLINE(N,1,2)-FLINE(N,1,1)                               SCC   5
      T2X=FLINE(N,1,1)-FLINE(N,1,2)                               SCC   6
      T1Y=FLINE(N,2,2)-FLINE(N,1,2)                               SCC   7
      T2Y=FLINE(N,2,1)-FLINE(N,2,2)                               SCC   8
      T1Z=FLINE(N,3,2)-FLINE(N,3,1)                               SCC   9
      T2Z=FLINE(N,3,1)-FLINE(N,3,2)                               SCC  10
      XNX=T2Y*T1Z-T2Z*T1Y                                         SCC  11
      YNY=T1X*T2Z-T2X*T1Z                                         SCC  12
      ZNZ=T2X*T1Y-T1X*T2Y                                         SCC  13
      FN=SQRT(ZNX**2+YNY**2+ZNZ**2)                               SCC  14
      IF (FN.EQ.0.) GO TO 5                                       SCC  15
      FVEC(N-1,1)=XNX/FN                                          SCC  16
      FVEC(N-1,2)=YNY/FN                                          SCC  17
      FVEC(N-1,3)=ZNZ/FN                                          SCC  18
      GO TO 10                                                    SCC  19
    5 FVEC(N-1,1)=0.                                              SCC  20
      FVEC(N-1,2)=0.                                              SCC  21
      FVEC(N-1,3)=0.                                              SCC  22
   10 CONTINUE                                                    SCC  23
      RETURN                                                      SCC  24
      END                                                         SCC  25

      SUBROUTINE CBC20                                            C20   1
C     CONTROL ROUTINE FOR VARIOUS TYPES OF PLOTS OF AN AIRCRAFT   C20   2
      COMMON ABC(20),JO,J1,J2,J3,J4,J5,J6,NMAF,NNAFOR,NFUS,NRADX(4),NFOR C20   3
     1K(4),NP,NPIDER,NF,NFINOR,NCAR,NCANOR,J2TEST,NW,NC,ABCDE(20),HORZ,V C20   4
     1ERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FRAG,PLOTSZ,TYPE,KODE,XMIN,X C20   5
     1MAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,ZMID,BIGD,ISP             C20   6
      COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4      C20   7
      DIMENSION ORG(3)                                           C20   8
      DATA TYPEO/3HORT/,TYPEP/3HPER/,TYPES/3HSTE/,TYPEV/3HVU3/    C20   9
      DATA X1/1HX/,Y1/1HY/,Z1/1HZ/                                C20  10
      REWIND KFILE2                                               C20  11
C     SAVE MIN AND MAX                                            C20  12
      XSAV=XMIN                                                   C20  13
      YSAV=YMIN                                                   C20  14
      ZSAV=ZMIN                                                   C20  15
      XMSAV=XMAX                                                  C20  16
      YMSAV=YMAX                                                  C20  17
      ZMSAV=ZMAX                                                  C20  18
      IF (TYPE.NE.TYPEV) GO TO 25                                 C20  19
      SCALE=BIGD/PLOTSZ                                           C20  20
      ORG(1)=PHI                                                  C20  21
      ORG(2)=THETA                                                C20  22
      ORG(3)=PSI                                                  C20  23
      PHI=0.                                                      C20  24
      THETA=0.                                                    C20  25
      PSI=0.                                                      C20  26
      YBIG=ORG(1)                                                 C20  27
      YORG=FLOAT(IFIX(YMAX/SCALE))+ORG(1)                         C20  28
      IF (YBIG.GT.ORG(2)) GO TO 5                                 C20  29
      YBIG=ORG(2)                                                 C20  30
      YORG=FLOAT(IFIX(ZMAX/SCALE))+ORG(2)                         C20  31
    5 IF (YBIG.GT.ORG(3)) GO TO 10                                C20  32
      YORG=FLOAT(IFIX(ZMAX/SCALE))+ORG(3)                         C20  33
      YBIG=ORG(3)                                                 C20  34
   10 CALL CALPLT(2.0,YORG,-3)                                    C20  35
C     ROTATE ON 3VIEW PLOTS                                       C20  36
      NCHAR=IFIX(6.*PLOTSZ)                                       C20  37
      IF (NCHAR.GT.80) GO TO 15                                   C20  38
      X=0.                                                        C20  39
      GO TO 20                                                    C20  40
   15 CONTINUE                                                    C20  41
      NDIF=(NCHAR-80)/2                                           C20  42
      X=FLOAT(NDIF)/6.                                            C20  43
      NCHAR=80                                                    C20  44
   20 CALL NOTATE(X,0.,2,ABC,0.,NCHAR)                            C20  45
      XMIN=0.                                                     C20  46
      YMIN=0.                                                     C20  47
      ZMIN=0.                                                     C20  48
      HORZ=X1                                                     C20  49
      VERT=Y1                                                     C20  50
      YORG=ORG(1)-YORG-1                                          C20  51
      CALL CALPLT(0.,YORG,-3)                                     C20  52
      CALL CBC21                                                  C20  53
      REWIND KFILE2                                               C20  54
      VERT=Z1                                                     C20  55
      YORG=ORG(2)-ORG(1)                                          C20  56
      CALL CALPLT(0.,YORG,-3)                                     C20  57
      CALL CBC21                                                  C20  58
      REWIND KFILE2                                               C20  59
      HORZ=Y1                                                     C20  60
      VERT=Y1                                                     C20  61
      YORG=ORG(3)-ORG(2)                                          C20  62
      YMIN=FLOAT(IFIX(YSAV/SCALE))*SCALE                          C20  63
      CALL CALPLT(0.,YORG,-3)                                     C20  64
      CALL CBC21                                                  C20  65
      X=FLOAT(IFIX(PLOTSZ+6.))                                    C20  66
      Y=1.-ORG(3)                                                 C20  67
      CALL CALPLT(X,Y,-3)                                         C20  68
   25 CONTINUE                                                    C20  69
      IF (TYPE.EQ.TYPES) GO TO 35                                 C20  70
C     NOTATE ID ON PLOT                                           C20  71
      X=0.                                                        C20  72
      NCHAR=IFIX(11.*PLOTSZ)+3                                    C20  73
      IF (NCHAR.LE.80) GO TO 30                                   C20  74
      NDIF=(NCHAR-80)/2                                           C20  75
      X=FLOAT(NDIF)/11.                                           C20  76
      NCHAR=80                                                    C20  77
   30 CALL NOTATE(X,.5,-.1,ABC,0.,NCHAR)                          C20  78
      CALL NOTATE(X,0.,-.1,ABCDE,0.,NCHAR)                        C20  79
   35 CONTINUE                                                    C20  80
      IF (TYPE.EQ.TYPEP.OR.TYPE.EQ.TYPES) GO TO 40               C20  81
C     ORTHOGRAPHIC                                                C20  82
      CALL CBC21                                                  C20  83
      X=FLOAT(IFIX(PLOTSZ+2.))                                    C20  84
      Y=0.                                                        C20  85
      GO TO 45                                                    C20  86
   40 ISP=1                                                       C20  87
      IF (TYPE.EQ.TYPES)ISP=2                                     C20  88
C     PERSPECTIVE OR STEREO                                       C20  89
      CALL CBC22                                                  C20  90
      X=PLOTSZ+2.                                                 C20  91
      IF (TYPE.EQ.TYPES)X=X+PLOTSZ                                C20  92
      Y=0.                                                        C20  93
C     END OF COMPLETE PLOT                                        C20  94
   45 CONTINUE                                                    C20  95
```

```
      CALL CALPLT(X,Y,-3)
C***  RESTORE MIN AND MAX
      XMIN=XSAV                                                    C20  96
      YMIN=YSAV                                                    C20  97
      ZMIN=ZSAV                                                    C20  98
      XMAX=XMSAV                                                   C20  99
      YMAX=YMSAV                                                   C20 100
      ZMAX=ZMSAV                                                   C20 101
      RETURN                                                       C20 102
C***  END OF PLTCOM                                                C20 103
      END                                                         C20 104
                                                                  C20 105
                                                                  C20 106

      SUBROUTINE CBC21                                             C21   1
C***  CONTROL ROUTINE FOR ORTHOGRAPHIC PROJECTIONS                C21   2
      COMMON ABC(20),J0,J1,J2,J3,J4,J5,J6,NNAF,NWAFDR,NFUS,NRADX(4),NFDR C21 3
     1X(4),NP,NPFDOOR,NF,NFINOR,NCAN,NCANOR,J2TEST,NW,NC,ABCDE(20),HORZ,V C21 4
     1ERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FMAG,PLOTSZ,TYPE,KODE,XMIN,X C21 5
     1MAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,ZMID,BIGD,ISP               C21   6
      DIMENSION A(2,3),C(3)                                        C21   7
      DATA XSEE/2HX /,YSEE/2HY /,ZSEE/2HZ /,XINTST/3HOUT/,CONV/.017453329 C21 8
     3/,HNUM2/2/,NAN2/24/                                          C21   9
      DATA HVU3/3HVU3/                                             C21  10
C***  INITIALIZE                                                   C21  11
      ITEST1=1                                                     C21  12
      DMAX=BIGD                                                    C21  13
      ITEST2=1                                                     C21  14
      IF (XINTST.NE.TEST1)ITEST1=0                                 C21  15
      IF (PSI.EQ.0..AND.THETA.EQ.0..AND.PHI.EQ.0.)ITEST2=0         C21  16
      SCALE=DMAX/PLOTSZ                                            C21  17
      THETA=CONV*THETA                                            C21  18
      PHI=CONV*PHI                                                 C21  19
      PSI=CONV*PSI                                                 C21  20
      IF (TYPE.EQ.HVU3) GO TO 5                                    C21  21
      XDIS=XMAX-XMIN                                               C21  22
      YDIS=YMAX-YMIN                                               C21  23
      ZDIS=ZMAX-ZMIN                                               C21  24
      XFIX=-.5*(DMAX-XDIS)                                         C21  25
      XMIN=XMIN-XFIX                                               C21  26
      YFIX=-.5*(DMAX-YDIS)                                         C21  27
      YMIN=YMIN-YFIX                                               C21  28
      ZFIX=-.5*(DMAX-ZDIS)                                         C21  29
      ZMIN=ZMIN-ZFIX                                               C21  30
      XMAX=XMAX-XFIX                                               C21  31
      YMAX=YMAX-YFIX                                               C21  32
      ZMAX=ZMAX-ZFIX                                               C21  33
C***  ADJUST MINIMUMS FOR GRID LINES                               C21  34
      XMIN=FLOAT(IFIX(XMIN/SCALE))*SCALE                           C21  35
      YMIN=FLOAT(IFIX(YMIN/SCALE))*SCALE                           C21  36
      ZMIN=FLOAT(IFIX(ZMIN/SCALE))*SCALE                           C21  37
    5 CONTINUE                                                     C21  38
C***  SETUP AXIS                                                   C21  39
      SINPSI=SIN(PSI)                                              C21  40
      SINTHE=SIN(THETA)                                            C21  41
      SINPHI=SIN(PHI)                                              C21  42
      COSPSI=COS(PSI)                                              C21  43
      COSTHE=COS(THETA)                                            C21  44
      COSPHI=COS(PHI)                                              C21  45
      IF (XSEE.NE.HORZ) GO TO 20                                   C21  46

C***  USE X FOR HORIZONTAL VARIABLE                                C21  47
      IF (ITEST2.EQ.0) GO TO 15                                    C21  48
      A(1,1)=COSTHE*COSPSI                                         C21  49
      A(1,2)=-SINPSI*COSPHI+SINTHE*COSPSI*SINPHI                   C21  50
      A(1,3)=-SINPSI*SINPHI+SINTHE*COSPSI*COSPHI                   C21  51
   15 HMIN=XMIN                                                    C21  52
      HMAX=XMAX                                                    C21  53
      HMID=XMID                                                    C21  54
      IHORZ=1                                                      C21  55
      GO TO 40                                                     C21  56
   20 IF (YSEE.NE.HORZ) GO TO 30                                   C21  57
C***  USE Y FOR HORIZONTAL VARIABLE                                C21  58
      IF (ITEST2.EQ.0) GO TO 25                                    C21  59
      A(1,1)=COSTHE*SINPSI                                         C21  60
      A(1,2)=COSPSI*COSPHI+SINTHE*SINPSI*SINPHI                    C21  61
      A(1,3)=-COSPSI*SINPHI+SINTHE*SINPSI*COSPHI                   C21  62
   25 HMIN=YMIN                                                    C21  63
      HMAX=YMAX                                                    C21  64
      HMID=YMID                                                    C21  65
      IHORZ=2                                                      C21  66
      GO TO 40                                                     C21  67
C***  USE Z FOR HORIZONTAL VARIABLE                                C21  68
   30 CONTINUE                                                     C21  69
      IF (ITEST2.EQ.0) GO TO 35                                    C21  70
      A(1,1)=-SINTHE                                               C21  71
      A(1,2)=COSTHE*SINPHI                                         C21  72
      A(1,3)=COSTHE*COSPHI                                         C21  73
   35 HMIN=ZMIN                                                    C21  74
      HMAX=ZMAX                                                    C21  75
      HMID=ZMID                                                    C21  76
      IHORZ=3                                                      C21  77
   40 IF (XSEE.NE.VERT) GO TO 50                                   C21  78
C***  USE X FOR VERTICAL VARIABLE                                  C21  79
      IF (ITEST2.EQ.0) GO TO 45                                    C21  80
      A(2,1)=COSTHE*COSPSI                                         C21  81
      A(2,2)=-SINPSI*COSPHI+SINTHE*COSPSI*SINPHI                   C21  82
      A(2,3)=-SINPSI*SINPHI+SINTHE*COSPSI*COSPHI                   C21  83
   45 VMIN=XMIN                                                    C21  84
      VMAX=XMAX                                                    C21  85
      VMID=XMID                                                    C21  86
      IVERT=1                                                      C21  87
      GO TO 70                                                     C21  88
   50 IF (YSEE.NE.VERT) GO TO 60                                   C21  89
C***  USE Y FOR VERTICAL VARIABLE                                  C21  90
      IF (ITEST2.EQ.0) GO TO 55                                    C21  91
      A(2,1)=COSTHE*SINPSI                                         C21  92
      A(2,2)=COSPSI*COSPHI+SINTHE*SINPSI*SINPHI                    C21  93
      A(2,3)=-COSPSI*SINPHI+SINTHE*SINPSI*COSPHI                   C21  94
   55 VMIN=YMIN                                                    C21  95
      VMAX=YMAX                                                    C21  96
      VMID=YMID                                                    C21  97
      IVERT=2                                                      C21  98
      GO TO 70                                                     C21  99
C***  USE Z FOR VERTICAL VARIABLE                                  C21 100
   60 CONTINUE                                                     C21 101
      IF (ITEST2.EQ.0) GO TO 65                                    C21 102
      A(2,1)=-SINTHE                                               C21 103
      A(2,2)=COSTHE*SINPHI                                         C21 104
      A(2,3)=COSTHE*COSPHI                                         C21 105
   65 VMIN=ZMIN                                                    C21 106
```

```
                                                                    C21 107
      VMAX=ZMAX                                                     C21 108
      VMID=ZMID                                                     C21 109
      IVERT=3                                                       C21 110
C***  CHECK PAPER PLANE                                             C21 111
70    IF (.NOT.(((IHORZ.EQ.1.AND.IVERT.EQ.2).OR.(IVERT.EQ.1.AND.IHORZ.EQ. C21 112
     12))) GO TO 75                                                 C21 113
      ITEST=1                                                       C21 114
      C(1)=-SINTHE                                                  C21 115
      C(2)=-COSTHE*SINPHI                                           C21 116
      C(3)=-COSTHE*COSPHI                                           C21 117
      GO TO 85                                                      C21 118
75    IF (.NOT.((IHORZ.EQ.1.AND.IVERT.EQ.3).OR.(IVERT.EQ.1.AND.IHORZ.EQ. C21 119
     13))) GO TO 80                                                 C21 120
      ITEST=2                                                       C21 121
      C(1)=-COSTHE*SINPSI                                           C21 122
      C(2)=COSPSI*COSPHI+SINTHE*SINPSI*SINPHI                       C21 123
      C(3)=-COSPSI*SINPHI+SINTHE*SINPSI*COSPHI                      C21 124
      GO TO 85                                                      C21 125
80    ITEST=1                                                       C21 126
      C(1)=-COSTHE*COSPSI                                           C21 127
      C(2)=-SINPSI*COSPHI+SINTHE*COSPSI*SINPHI                      C21 128
      C(3)=-SINPSI*SINPHI+SINTHE*COSPSI*COSPHI                      C21 129
85    CONTINUE                                                      C21 130
C***  CENTER WITHIN PAGE SIZE IF SIZE GREATER THAN 28 INCHES        C21 131
      IF (PLOTSZ.GT.28..AND.TYPE.NE.HVU3)VMIN=-13.*SCALE*FLOAT(IFIX(VMID C21 132
     1/SCALE))*SCALE                                                C21 133
C***  ROTATE MIDPOINT TO PLACE ROTATED VIEW CORRECTLY               C21 134
      IF (ITEST2.EQ.0) GO TO 90                                     C21 135
      AMID1=A(1,1)*XMID+A(1,2)*YMID+A(1,3)*ZMID                     C21 136
      AMID2=A(2,1)*XMID+A(2,2)*YMID+A(2,3)*ZMID                     C21 137
      HMID=HMIN-HMID+AMID1                                          C21 138
      VMIN=VMIN-VMID+AMID2                                          C21 139
90    CONTINUE                                                      C21 140
C***  BEGIN PLOTTING LINES                                          C21 141
C***  WING                                                          C21 142
      DO 100 I=1,2                                                  C21 143
      IF (J1.EQ.0) GO TO 110                                        C21 144
      DO 100 I=1,2                                                  C21 145
      CALL PLOTIT(NWAF,NW,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN,SCAL C21 146
     1E,A,C)                                                        C21 147
100   CONTINUE                                                      C21 148
      DO 105 I=1,2                                                  C21 149
      CALL PLOTIT(NWAF,NW,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN,SCAL C21 150
     1E,A,C)                                                        C21 151
105   CONTINUE                                                      C21 152
C***  FUSELAGE                                                      C21 153
110   IF (J2.EQ.0) GO TO 125                                        C21 154
      DO 115 NFU=1,NFUS                                             C21 155
      NANG1=NRADX(NFU)                                              C21 156
      NFUSOR=NFORX(NFU)                                             C21 157
      CALL PLOTIT(NANG1,NFUSOR,ITEST,ITEST1,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN C21 158
     1,SCALE,A,C)                                                   C21 159
115   CONTINUE                                                      C21 160
      DO 120 NFU=1,NFUS                                             C21 161
      NANG1=NRADX(NFU)                                              C21 162
      NFUSOR=NFORX(NFU)                                             C21 163
      CALL PLOTIT(NFUSOR,NANG1,ITEST,ITEST1,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN C21 164
     1,SCALE,A,C)                                                   C21 165
120   CONTINUE                                                      C21 166
C***  NACELLES                                                      C21 167
125   IF (J3.EQ.0) GO TO 140                                        C21 168

      NANG1=NRAD1                                                   C21 167
      DO 130 NP1=1,NP                                               C21 168
      CALL PLOTIT(NANG1,NPODOR,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN C21 169
     1,SCALE,A,C)                                                   C21 170
130   CONTINUE                                                      C21 171
      DO 135 NP1=1,NP                                               C21 172
      CALL PLOTIT(NPODOR,NANG1,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN C21 173
     1,SCALE,A,C)                                                   C21 174
135   CONTINUE                                                      C21 175
C***  FINS                                                          C21 176
140   IF (J4.EQ.0) GO TO 155                                        C21 177
      DO 145 NFI=1,NF                                               C21 178
      CALL PLOTIT(NUR2,NFINOR,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN, C21 179
     1SCALE,A,C)                                                    C21 180
145   CONTINUE                                                      C21 181
      DO 150 NFI=1,NF                                               C21 182
      CALL PLOTIT(NFINOR,NUR2,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN, C21 183
     1SCALE,A,C)                                                    C21 184
150   CONTINUE                                                      C21 185
C***  CANARD                                                        C21 186
155   IF (J5.EQ.0) GO TO 180                                        C21 187
      DO 165 NCA=1,NCAN                                             C21 188
      DO 160 I=1,2                                                  C21 189
      CALL PLOTIT(2,NC,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN,SCALE,A C21 190
     1,C)                                                           C21 191
160   CONTINUE                                                      C21 192
165   CONTINUE                                                      C21 193
      DO 175 NCA=1,NCAN                                             C21 194
      DO 170 I=1,2                                                  C21 195
      CALL PLOTIT(NC,2,ITEST,ITEST1,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN,SCALE,A C21 196
     1,C)                                                           C21 197
170   CONTINUE                                                      C21 198
175   CONTINUE                                                      C21 199
180   CONTINUE                                                      C21 200
      RETURN                                                        C21 201
      END                                                           C21 202
                                                                    C21 203
                                                                    C21 204
                                                                    C21 205
                                                                    C21 206

      SUBROUTINE PLOTIT(ML,NPT,ITEST,ITEST,ITEST1,ITEST2,IHORZ,IVERT,HMIN,VMIN PLT 1
     1,SCALE,A,C)                                                   PLT 2
C***  READS LINES OF POINTS DEFINING A SURFACE FROM TAPE.           PLT 3
C***     MANIPULATES IN SPECIFIED MANNER, AND PLOTS                 PLT 4
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE2,KFILE3,KFILE4         PLT 5
      DIMENSION VECRT(30,3,2),VECLF(30,3,2),ALINE(31,3),RLINE(31,2),AVEC PLT 6
     1(30,2),RLINE(33,2),PLINE(31,2),A(2,3),C(3)                    PLT 7
      NVEC=NPT-1                                                    PLT 8
      DO 155 N=1,NL                                                 PLT 9
      IF (N.GT.1) GO TO 5                                           PLT 10
      KODE=3                                                        PLT 11
      K1=1                                                          PLT 12
      K2=2                                                          PLT 13
      GO TO 20                                                      PLT 14
5     KODE=1                                                        PLT 15
      K1=1                                                          PLT 16
      K2=2                                                          PLT 17
```

```
      DO 15 NV=1,NVEC                                              PLT  18
      DO 10 M3=1,3                                                 PLT  19
      VECRT(NV,M3,1)=VECRT(NV,M3,2)                                PLT  20
      VECLF(NV,M3,1)=VECLF(NV,M3,2)                                PLT  21
   10 CONTINUE                                                     PLT  22
   15 CONTINUE                                                     PLT  23
   20 READ (KFILE2)((ALINE(NN,M3),NN=1,NPT),M3=1,3)                PLT  24
      IF (N.NE.NL) GO TO 25                                        PLT  25
      KODE=2                                                       PLT  26
      K1=1                                                         PLT  27
      K2=1                                                         PLT  28
      GO TO 35                                                     PLT  29
   25 READ (KFILE2)((VECRT(NN,M3,2),NN=1,NVEC),N3=1,3)             PLT  30
      DO 30 NN=1,NVEC                                              PLT  31
      VECLF(NN,1,2)=VECRT(NN,1,2)                                  PLT  32
      VECLF(NN,2,2)=-VECRT(NN,2,2)                                 PLT  33
      VECLF(NN,3,2)=VECRT(NN,3,2)                                  PLT  34
   30 CONTINUE                                                     PLT  35
   35 DO 150 NN2=1,2                                               PLT  36
C***    LOOP FOR RIGHT AND LEFT SIDE OF AIRCRAFT                   PLT  37
      IF (NN2.EQ.1) GO TO 45                                       PLT  38
   40 ALINE(NN,2)=-ALINE(NN,2)                                     PLT  39
   45 IF (ITEST1.EQ.1) GO TO 70                                    PLT  40
      IF (ITEST2.EQ.1) GO TO 55                                    PLT  41
C***    NO ROTATION OR VISIBILITY TEST                            PLT  42
      DO 50 NN=1,NPT                                               PLT  43
      XLINE(NN,1)=ALINE(NN,IHORZ)                                  PLT  44
      XLINE(NN,2)=ALINE(NN,IVERT)                                  PLT  45
   50 GO TO 65                                                     PLT  46
C***    ROTATE BUT NO VISIBILITY TEST                             PLT  47
   55 CALL PROT(NPT,A,ALINE,RLINE)                                 PLT  48
      DO 60 NN=1,NPT                                               PLT  49
      DO 60 N2=1,2                                                 PLT  50
      XLINE(NN,N2)=RLINE(NN,N2)                                    PLT  51
   60 CONTINUE                                                     PLT  52
C***    SCALE AND PLOT                                            PLT  53
   65 XLINE(NPT+1,1)=VMIN                                          PLT  54
      XLINE(NPT+1,2)=VMIN                                          PLT  55
      XLINE(NPT+2,1)=SCALE                                         PLT  56
      XLINE(NPT+2,2)=SCALE                                         PLT  57
      CALL LINE(XLINE(1,1),XLINE(1,2),NPT,1,0,0,0)                 PLT  58
      GO TO 150                                                    PLT  59
   70 IF (ITEST2.EQ.1) GO TO 95                                    PLT  60
C***    CHECK VISIBILITY BUT NO ROTATION                          PLT  61
      DO 75 NN=1,NPT                                               PLT  62
      RLINE(NN,1)=ALINE(NN,IHORZ)                                  PLT  63
      RLINE(NN,2)=ALINE(NN,IVERT)                                  PLT  64
   75 CONTINUE                                                     PLT  65
      DO 90 NN=1,NVEC                                              PLT  66
      DO 85 N2=1,2                                                 PLT  67
      IF (NN2.EQ.2) GO TO 80                                       PLT  68
      RVEC(NN,N2)=VECRT(NN,ITEST,N2)                               PLT  69
      GO TO 85                                                     PLT  70
   80 RVEC(NN,N2)=VECLF(NN,ITEST,N2)                               PLT  71
   85 CONTINUE                                                     PLT  72
   90 CONTINUE                                                     PLT  73
      GO TO 115                                                    PLT  74
C***    ROTATE AND CHECK VISIBILITY                               PLT  75
   95 CALL PROT(NPT,A,ALINE,RLINE)                                 PLT  76
      IF (NN2.EQ.2) GO TO 105                                      PLT  77

      DO 100 N2=K1,K2                                              PLT  78
      CALL VECROT(NVEC,C,VECRT(1,1,N2),RVEC(1,N2))                 PLT  79
  100 CONTINUE                                                     PLT  80
      GO TO 115                                                    PLT  81
  105 DO 110 N2=K1,K2                                              PLT  82
      CALL VECROT(NVEC,C,VECLF(1,1,N2),RVEC(1,N2))                 PLT  83
  110 CONTINUE                                                     PLT  84
C***    FIND VISIBLE LINES                                        PLT  85
  115 IF (ITEST.NE.2) GO TO 125                                    PLT  86
      DO 120 N2=K1,K2                                              PLT  87
      DO 120 M=1,NVEC                                              PLT  88
  120 RVEC(M,N2)=-RVEC(M,N2)                                       PLT  89
  125 CALL VISTST(KODE,NPT,NSET,NNUM,RLINE,RVEC,PLINE)             PLT  90
      IF (ITEST.NE.2) GO TO 135                                    PLT  91
      DO 130 N2=K1,K2                                              PLT  92
      DO 130 M=1,NVEC                                              PLT  93
  130 RVEC(M,N2)=-RVEC(M,N2)                                       PLT  94
  135 CONTINUE                                                     PLT  95
      IF (NSET.EQ.0) GO TO 150                                     PLT  96
C***    SCALE AND PLOT                                            PLT  97
      NIT=0                                                        PLT  98
      DO 145 N1=1,NSET                                             PLT  99
      NN=NNUM(N1)                                                  PLT 100
      DO 140 NN1=1,NN                                              PLT 101
      NIT=NIT+1                                                    PLT 102
      XLINE(NN1,1)=PLINE(NIT,1)                                    PLT 103
      XLINE(NN1,2)=PLINE(NIT,2)                                    PLT 104
  140 CONTINUE                                                     PLT 105
      XLINE(NN+1,1)=VMIN                                           PLT 106
      XLINE(NN+1,2)=VMIN                                           PLT 107
      XLINE(NN+2,1)=SCALE                                          PLT 108
      XLINE(NN+2,2)=SCALE                                          PLT 109
      CALL LINE(XLINE(1,1),XLINE(1,2),NN,1,0,0,0)                  PLT 110
  145 CONTINUE                                                     PLT 111
  150 CONTINUE                                                     PLT 112
  155 CONTINUE                                                     PLT 113
      RETURN                                                       PLT 114
      END                                                          PLT 115

      SUBROUTINE VISTST(KODE,NPT,NSET,NNUM,RLINE,RVEC,PLINE)       VIS   1
C***    TESTS A LINE OF POINTS FOR VISIBILITY                     VIS   2
      DIMENSION NNUM(4),RLINE(31,2),RVEC(30,2),PLINE(31,2)         VIS   3
      NVEC=NPT-1                                                   VIS   4
      NPLT=0                                                       VIS   5
      NSET=0                                                       VIS   6
      ICOUNT=0                                                     VIS   7
      GO TO (5,10,15),KODE                                         VIS   8
    5 N1=1                                                         VIS   9
      N2=2                                                         VIS  10
      GO TO 20                                                     VIS  11
   10 N1=1                                                         VIS  12
      N2=1                                                         VIS  13
      GO TO 20                                                     VIS  14
   15 N1=2                                                         VIS  15
      N2=2                                                         VIS  16
   20 DO 65 N=1,NPT                                                VIS  17
      IF (N.EQ.1) GO TO 30                                         VIS  18
      IF (N.EQ.NPT) GO TO 40                                       VIS  19
```

351

```
C***        CONTROL ROUTINE FOR PERSPECTIVE AND STEREO                    C22    2
      COMMON ABC(20),J0,J1,J2,J3,J4,J5,J6,MNAF,NNAFOR,NFUS,MRADX(41),NFOR C22    3
     1X(4),MP,NPDDOR,NF,NFINOR,MCAN,MCANOR,J2TEST,NW,NC,ABCDE(20),HORZ,V   C22    4
     1ERT,TEST1,PHI,THETA,PSI,XF,YF,ZF,DIST,FRAG,PLOTSZ,TYPE,KODE,XMIN,X   C22    5
     1MAX,YMIN,YMAX,ZMIN,ZMAX,XMID,YMID,ZMID,BIGD,ISP                      C22    6
      COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4               C22    7
      DIMENSION XINIT(2),YINIT(2),ZINIT(2)                                C22    8
      DATA NAN/2H24/                                                      C22    9
      XINIT(1)=PHI                                                        C22   10
      XINIT(2)=XF                                                         C22   11
      YINIT(1)=THETA                                                      C22   12
      YINIT(2)=YF                                                         C22   13
      ZINIT(1)=PSI                                                        C22   14
      ZINIT(2)=ZF                                                         C22   15
      CALL STERPT(XINIT,YINIT,ZINIT,0,1,0,3,PLOTSZ,DIST,FRAG)             C22   16
C***        LOOP FOR RIGHT AND LEFT FRAMES                                C22   17
      DO 95 IC=1,ISP                                                      C22   18
      REWIND KFILE2                                                      C22   19
      NCI=-IC                                                            C22   20
C***        BEGIN PLOTTING LINES                                          C22   21
C***        WING                                                         C22   22
      IF (J1.EQ.0) GO TO 20                                              C22   23
      DO 10 I=1,2                                                        C22   24
   10 CALL PLTIT3(NNAF,NW,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,NCI)   C22   25
      DO 15 I=1,2                                                        C22   26
   15 CALL PLTIT3(NW,NNAF,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,NCI)   C22   27
C***        FUSELAGE                                                     C22   28
   20 IF (J2.EQ.0) GO TO 35                                              C22   29
      DO 25 NFU=1,NFUS                                                   C22   30
      NANG1=NRADX(NFU)                                                   C22   31
      NFUSOR=NFORX(NFU)                                                  C22   32
      CALL PLTIT3(NANG1,NFUSOR,NFUSOR,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,N C22   33
     1CI)                                                                C22   34
   25 CONTINUE                                                           C22   35
      DO 30 NFU=1,NFUS                                                   C22   36
      NANG1=NRADX(NFU)                                                   C22   37
      NFUSOR=NFORX(NFU)                                                  C22   38
      CALL PLTIT3(NFUSOR,NANG1,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,N  C22   39
     1CI)                                                                C22   40
   30 CONTINUE                                                           C22   41
C***        NACELLES                                                     C22   42
   35 IF (J3.EQ.0) GO TO 50                                              C22   43
      NANG1=NAN2+1                                                       C22   44
      DO 40 NP=1,NP                                                      C22   45
      CALL PLTIT3(NANG1,NPDDOR,NANG1,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,N C22   46
     1CI)                                                                C22   47
   40 CONTINUE                                                           C22   48
      DO 45 NP=1,NP                                                      C22   49
      CALL PLTIT3(NPDDOR,NANG1,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,N  C22   50
     1CI)                                                                C22   51
C***        FINS                                                         C22   52
   45 CONTINUE                                                           C22   53
   50 IF (J4.EQ.0) GO TO 65                                              C22   54
      DO 55 NF=1,NF                                                      C22   55
   55 CALL PLTIT3(2,NFINOR,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,NCI)  C22   56
      DO 60 NF=1,NF                                                      C22   57
      CALL PLTIT3(NFINOR,2,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,NCI)  C22   58
   60 CALL PLTIT3(NFINOR,2,PHI,THETA,PSI,XF,YF,ZF,PLOTSZ,DIST,FRAG,NCI)  C22   59
C***        CANARD                                                       C22   60
```

```
      DO 25 NN=N1,N2                                                    VIS   20
      IF (RVEC(N-1,NN),GT.0..OR.(RVEC(N,NN).GT.0.)) GO TO 60            VIS   21
   25 CONTINUE                                                          VIS   22
      GO TO 50                                                          VIS   23
   30 DO 35 NN=N1,N2                                                    VIS   24
      IF (RVEC(N,NN).GT.0.) GO TO 60                                    VIS   25
   35 CONTINUE                                                          VIS   26
      GO TO 50                                                          VIS   27
   40 DO 45 NN=N1,N2                                                    VIS   28
      IF (RVEC(RVEC,NN).GT.0.) GO TO 60                                 VIS   29
   45 CONTINUE                                                          VIS   30
C***        POINT NOT VISIBLE                                            VIS   31
   50 IF (ICOUNT.LE.1) GO TO 55                                         VIS   32
      NSET=NSET+1                                                       VIS   33
      NNUM(NSET)=ICOUNT                                                 VIS   34
   55 ICOUNT=0                                                          VIS   35
      GO TO 65                                                          VIS   36
C***        POINT IS VISIBLE                                             VIS   37
   60 NPLT=NPLT+1                                                       VIS   38
      ICOUNT=ICOUNT+1                                                   VIS   39
      PLINE(NPLT,1)=ALINE(N,1)                                          VIS   40
      PLINE(NPLT,2)=ALINE(N,2)                                          VIS   41
   65 IF (ICOUNT.LE.1) GO TO 70                                         VIS   42
      NSET=NSET+1                                                       VIS   43
      NNUM(NSET)=ICOUNT                                                 VIS   44
   70 RETURN                                                            VIS   45
      END                                                               VIS   46
                                                                        VIS   47
```

```
C***  SUBROUTINE PTROT(NPT,A,ALINE,RLINE)                                PTR    1
C***        ROTATES AND PROJECTS A SET OF 3-D POINTS                     PTR    2
      DIMENSION A(2,3),ALINE(31,3),RLINE(31,2)                           PTR    3
      DO 10 N=1,NPT                                                      PTR    4
      RLINE(N,1)=0.                                                      PTR    5
      RLINE(N,2)=0.                                                      PTR    6
      DO 5 I=1,2                                                         PTR    7
      DO 5 J=1,3                                                         PTR    8
    5 RLINE(N,I)=RLINE(N,I)+A(I,J)*ALINE(N,J)                            PTR    9
   10 CONTINUE                                                          PTR   10
      RETURN                                                            PTR   11
      END                                                               PTR   12
```

```
C***  SUBROUTINE VECROT(RVEC,C,FVEC,RVEC)                                VEC    1
C***        TRANSFORMS VECTORS                                          VEC    2
      DIMENSION C(3),FVEC(30,3),RVEC(30)                                 VEC    3
      DO 10 N=1,NVEC                                                     VEC    4
      SUM=0.                                                            VEC    5
      DO 5 NN=1,3                                                       VEC    6
    5 SUM=SUM+C(NN)*FVEC(N,NN)                                          VEC    7
      RVEC(N)=SUM                                                       VEC    8
   10 RETURN                                                            VEC    9
      END                                                              VEC   10
```

```
      SUBROUTINE CBC22                                                   C22    1
```

352

```
C***  PROGRAMER - GEORGE C. SALLEY
      SUBROUTINE STERPT(X,Y,Z,N,R,NC,IP,PAG,PLA,NPR)
      DIMENSION VP(3),TRAN(3),SANG(3),CANG(3),ADJ(3),PT(4),XLP(2),ZLP(2)
      DIMENSION X(1),Y(1),Z(1)
      DIMENSION PLH(4),PLV(4),PL2(2)
      DIMENSION PIK(4),PIV(4),PIZ(2)
      DIMENSION ILP(4),IPL(4)
      DATA PI,PI2,PI32,PI42/3.1415926,1.5707963,4.7123889,6.2831952/
      DATA PAR/1.125/
      DATA NPG/0/
      DATA NPT/1/
      DATA FRAME/9.80/
      DATA TURN/11.01/
      NO=1
      KK=K
      II=IP
      IF (NC) 80,5,110
    5 NP=NONK+1
      NR=NNP+K
      PLIM=PAG/2.
      SF=XPR
      VPL=PLA
      DO 10 I=1,4
      PLX(I)=0.
      PLY(I)=0.
      IPL(I)=0.
   10 ILP(I)=0.
      DO 15 I=1,2
   15 PIZ(I)=0.
      VPX=X(NP)
      VPY=Y(NP)
      VPZ=Z(NP)
      FPX=X(NR)
      FPY=Y(NR)
      FPZ=Z(NR)
      VX=VPX-FPX
      VY=VPY-FPY
      VZ=VPZ-FPZ
      VP(2)=SQRT((VX**2)+(VY**2))
      VP(3)=SQRT((VZ**2)+(VP(2)**2))
      TRAN(1)=VPX-(VPL*(VX/VP(2)))
      TRAN(2)=VPY-(VPL*(VY/VP(2)))
      TRAN(3)=VPZ-(VPL*(VZ/VP(3)))
      VANG=ATAN(PAR/VP(3))
      IF (VX) 55,20,35
   20 PANG=PI2
      GO TO 75
   25 PANG=PI2
      GO TO 75
   30 PANG=PI32
   35 IF (VY) 50,40,45
   40 PANG=0.
      GO TO 75
   45 PANG=ATAN(VY/VX)
      GO TO 75
   50 PANG=PI+2.-ATAN((ABS(VY))/VX)
      GO TO 75
   55 IF (VY) 70,60,65
   60 PANG=PI
   65 PANG=PI-ATAN(VY/(ABS(VX)))
      GO TO 75
   70 PANG=PI+ATAN((ABS(VY))/(ABS(VX)))
   75 PANG=PI32-PANG
      VANG=PANG-VANG
      RANG=UANG+(2.*VANG)
      SANG(1)=SIN(UANG)
      SANG(2)=SIN(RANG)
      CANG(1)=COS(UANG)
      CANG(2)=COS(RANG)
      SANG(3)=V2/VP(3)
      CANG(3)=VP(2)/VP(3)
      VP(3)=VPL
      XLP(1)=0.
      ZLP(1)=0.
      XLP(2)=FRAME
      ZLP(2)=0.
      ADJ(1)=PLIM
      ADJ(2)=PLIM
      ADJ(3)=ADJ(2)+FRAME
      IF (N) 300,300,110
   80 M=IABS(NC)
      L=M
      IF (NPG+NC) 115,85,115
   85 IF (2*NC) 300,95,90
   90 NPG=2
   95 NPG=1
      GO TO 100
  100 CONTINUE
      DO 105 I=1,L
      CALL CALPLT(TURN,0.,-3)
  105 CONTINUE
      CALL CALPLT(XLP(M),ZLP(M),3)
  110 M=1
      L=2
  115 DO 295 I=M,L
      IF (NPG) 300,120,150
  120 IF (NC) 125,145,145
  125 IF (2*NC) 150,130,150
  130 IF (NP+NC) 300,140,135
  135 NPT=2
```

```
   65 IF (J5.EQ.0) GO TO 90
      DO 75 NCA=1,NCAN
      DO 70 I=1,2
   70 CALL PLTIT3(Z,NC,PHI,THETA,PSI,XF,YF,ZF,PLOTS1,DIST,FMAG,NC1)
   75 CONTINUE
      DO 85 NCA=1,NCAN
      DO 80 I=1,2
   80 CALL PLTIT3(NC,Z,PHI,THETA,PSI,XF,YF,ZF,PLOTS1,DIST,FMAG,NC1)
   85 CONTINUE
   90 CONTINUE
   95 CONTINUE
      RETURN
      END
```

353

```
      GO TO 145
:40   NPT=1
145   CONTINUE
150   CALL CALPLT(XLP(I),ZLP(I),3)
      DO 290 J=1,N
      PT(1)=((X(NO)-TRAN(1))*CANG(I))-((Y(NO)-TRAN(2))*SANG(I))
      PT(4)=((X(NO)-TRAN(1))*SANG(I))+((Y(NO)-TRAN(2))*CANG(I))
      PT(2)=((PT(4)*CANG(3))-((Z(NO)-TRAN(3))*SANG(3)))
      PT(3)=((PT(4)*SANG(3))+((Z(NO)-TRAN(3))*CANG(3)))
155   IF (PT(2)) 155,180,180
      IF (ILP(I)) 300,160,175
160   IF (II-3) 165,170,300
165   VX=PLX(I)-PT(1)
      VY=PLY(I)-PT(2)
      VZ=PLZ(I)-PT(3)
      VPL(I)=SQRT((VX**2)+(VY**2))
      VPL(2)=SQRT((VZ**2)+(VP(I)**2))
      VPL=PLV(I)/(VV/VPL(I))
      PT(1)=PLX(I)-((VX/VPL(I))*VPL)
      PT(I)=PT(4)
      PLV(I)=PT(2)
      PT(2)=0.
      PT(4)=PLZ(I)-((VZ/VPL(2))*VPL)
      PT(3)=PT(3)
      PT(3)=PT(4)
      ILP(I)=1
170   ILP(I)=1
175   PLX(I)=PT(1)
      PLY(I)=PT(2)
      PLZ(I)=PT(3)
      GO TO 270
180   IF (ILP(I)) 300,200,185
185   IF (II-3) 190,195,300
190   II=3
      IPL(I)=1
      PLX(I)=PLX(I)
      PLY(I)=PT(2)
      PLZ(I)=PT(3)
      VX=PT(1)-PLX(I)
      VY=PT(2)-PLY(I)
      VZ=PT(3)-PLZ(I)
      VPL(I)=SQRT((VX**2)+(VY**2))
      VPL(2)=SQRT((VZ**2)+(VP(I)**2))
      VPL=PT(2)/(VV/VPL(I))
      PT(2)=0.
      PT(1)=PT(1)-((VX/VPL(I))*VPL)
      PT(3)=PT(3)-((VZ/VPL(2))*VPL)
195   ILP(I)=0
200   PLY(I)=PT(1)
      PLY(I)=PT(2)
      PLZ(I)=PT(3)
205   XP=(PT(3)-(PT(2)*(-PT(1))/(PT(2)-VP(3))))/((PT(2)-VP(3)))*SF
      VPL=SORT((NP**2)+(PT(2)**2)+(PT(2)-VP(3)))*SF
      IF (VPL-PLIM) 235,235,210
210   IF (ILP(I-2)) 300,215,230
215   IF (II-3) 220,225,300
220   PLX(I+2)=XP
```

```
      PLY(I+2)=ZP
      XP=PLIM*(XP/VPL)
      ZP=PLIM*(ZP/VPL)
      ILP(I+2)=1
      GO TO 265
225   ILP(I+2)=1
230   PLX(I+2)=XP
      PLY(I+2)=ZP
      GO TO 270
235   IF (ILP(I+2)) 300,255,240
240   IF (II-3) 245,250,300
245   II=3
      IPL(I+2)=1
      PLX(I+2)=XP
      PLY(I+2)=ZP
      VPL=SQRT((PLX(I+2)**2)+(PLY(I+2)**2))
      XP=PLIM*(PLX(I+2)/VPL)
      ZP=PLIM*(PLY(I+2)/VPL)
      ILP(I+2)=0
250   IF (ISORT((PLX(I+2)**2)+((PLY(I+2)**2))-PLIM) 260,260,240
255   IF ((SQRT((PLX(I+2)**2)+((PLY(I+2)**2))-PLIM) 260,260,240
260   PLX(I+2)=XP
      PLY(I+2)=ZP
265   XP=XP*ADJ(I+1)
      VPT=ZP*ADJ(I)
      CALL CALPLT(XPT,VPT,II)
270   II=2
275   IF (IPL(I+2)) 300,280,275
      KP=PIX(I+2)
      ZP=PLY(I+2)
      GO TO 255
280   IPL(I)=0
285   PIX(I)=PLX(I)
      PYT(2)=PLY(I)
      PYT(3)=PLZ(I)
290   NO=NO+KK
      NO=1
295   II=1P
      KLP(I)=XPT
      ZLP(I)=VPT
300   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLTIT3(NL,NPT,PHI,THETA,PSI,XF,VF,ZF,PLOTSZ,DIST,FRAG,M
     1CI)
C***  READS LINES OF POINTS DEFINING A SURFACE FROM TAPE
      COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4
      DIMENSION ALINE(33,3)
      ALINE(NPT+1,1)=PHI
      ALINE(NPT+1,2)=XF
      ALINE(NPT+1,2)=THETA
      ALINE(NPT+2,2)=VF
      ALINE(NPT+1,3)=PSI
      ALINE(NPT+2,3)=ZF
```

```
      DO 25 M=1,ML
      READ (KFILE22)((ALINE(MM,M3),MM=1,NPT),M3=1,3)
      IF (N.EQ.ML) GO TO 5
C***  SKIP VECTORS
      READ (KFILE2)VEC
C***  LOOP FOR RIGHT AND LEFT SIDE OF AIRCRAFT
    5 DO 20 NN2=1,2
      IF (NN2.EQ.1) GO TO 15
      DO 10 NN=1,NPT
   10 ALINE(NN,2)=-ALINE(NN,2)
   15 CALL STERP(ALINE(1,1),ALINE(1,2),ALINE(1,3),NPT,1,NC1,3,PLOTS2,DI
     1ST,PMAG)
   20 CONTINUE
   25 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALPLT(A,B,I)
C
C***  IMPORTANT???
C
C***  THIS SUBROUTINE USES THE PLOTTER SOFTWARE AVAILABLE AT N. C. STATE
C***  UNIVERSITY (SUBROUTINES PLOT AND ORIGIN) TO REPRODUCE THE ACTIONS
C***  OF THE CALCOMP SOFTWARE SUBROUTINE CALPLT.
C
C***  IF THE USER DOES NOT HAVE THE ORIGINAL CALCOMP ROUTINE CALPLT
C***  AVAILABLE TO HIM, HE MUST WRITE A DUMMY SUBROUTINE, USING THE
C***  PLOTTER SOFTWARE AVAILABLE AT HIS INSTALLATION, WHICH WILL
C***  REPRODUCE THE ACTION OF CALPLT.
      DIMENSION A(1),B(1)
      IF (I.LT.0) GO TO 5
      J=I-2
      J=J-2
      J=IABS(I)
      CALL PLOT(A,B,J)
    5 CALL ORIGIN(A,B,1,0)
      RETURN
      END
```

```
      XI=1
      IF ((XR-J).GT.0.1)J=J+1
    5 DO 5 K=1,J
      DO(K)=D(K)
      DO(J-1)=STOP
      CALL SYMBOL(A,B,C,DO,E)
      RETURN
      END
```

```
      SUBROUTINE LINE(A,B,I,J,K,L,S)
C
C***  IMPORTANT???
C
C***  THIS SUBROUTINE USES THE PLOTTER SOFTWARE AVAILABLE AT N. C. STATE
C***  UNIVERSITY (SUBROUTINE PLOT) TO REPRODUCE THE ACTIONS OF THE
C***  CALCOMP SOFTWARE SUBROUTINE LINE.
C
C***  IF THE USER DOES NOT HAVE THE ORIGINAL CALCOMP ROUTINE LINE
C***  AVAILABLE TO HIM, HE MUST WRITE A DUMMY SUBROUTINE, USING THE
C***  PLOTTER SOFTWARE AVAILABLE AT HIS INSTALLATION, WHICH WILL
C***  REPRODUCE THE ACTION OF LINE.
      DIMENSION A(1),B(1),X(31),Y(31)
      XMIN=A(I+1)
      XSCALE=A(I+2)
      YMIN=B(I+1)
      YSCALE=B(I+2)
      DO 5 II=1,I
      X(II)=(A(II)-XMIN)/XSCALE
    5 Y(II)=(B(II)-YMIN)/YSCALE
      CALL PLOT(X,Y,I)
      RETURN
      END
```

```
      SUBROUTINE NOTATE(A,B,C,D,E,I)
C
C***  IMPORTANT???
C
C***  THIS SUBROUTINE USES THE PLOTTER SOFTWARE AVAILABLE AT N. C. STATE
C***  UNIVERSITY (SUBROUTINE SYMBOL) TO REPRODUCE THE ACTIONS OF THE
C***  CALCOMP SOFTWARE SUBROUTINE NOTATE.
C
C***  IF THE USER DOES NOT HAVE THE ORIGINAL CALCOMP ROUTINE NOTATE
C***  AVAILABLE TO HIM, HE MUST WRITE A DUMMY SUBROUTINE, USING THE
C***  PLOTTER SOFTWARE AVAILABLE AT HIS INSTALLATION, WHICH WILL
C***  REPRODUCE THE ACTION OF NOTATE.
      DIMENSION D(1),DO(21)
      DATA STOP/4H    /
      J=I/4
```

PROGRAM D2290   PLOTS OF AIRCRAFT CONFIGURATION

CONFIGURATION DESCRIPTION

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- COMPLETE AIRPLANE

Figure D-3. Example of the unplotted portion of the sample output for the PLOT program.

```
0.0        0.36670    0.75000    1.15420    1.57290    1.79790    1.87500    1.89170    1.89580    1.89580
1.89580    1.89580    1.89580    1.89170    1.87500    1.84580    1.73960    1.28130    0.05210    0.40830
0.0
2.31250    2.32290    2.33750    2.36460    2.47920    2.82920    3.26250    3.66670    4.00620    4.32000
4.62500    4.92000    5.22920    5.58750    5.98960    6.46250    6.93120    6.93750    6.93750    6.93750
6.93790
0.0        0.36250    0.73960    1.14170    1.54580    1.78130    1.86040    1.87500    1.87500    1.87500
1.87500    1.87500    1.87500    1.87080    1.85420    1.83330    1.70830    1.25030    0.03330    0.40420
0.0
2.33330    2.33750    2.35830    2.37500    2.48540    2.82500    3.25420    3.65420    3.99170    4.30420
4.60420    4.89580    5.20420    5.55420    5.95420    6.42710    6.86870    6.87500    6.87500    6.87500
6.87500
0.0        0.34790    0.71680    1.09580    1.48960    1.75000    1.84790    1.87500    1.87500    1.87500
1.87500    1.87500    1.87500    1.86670    1.84580    1.81250    1.63750    1.20030    0.80210    0.38540
0.0
2.39580    2.39580    2.40630    2.43120    2.52500    2.82920    3.22920    3.62290    3.96040    4.27290
4.57290    4.86460    5.17290    5.52290    5.92000    6.37500    6.74790    6.75000    6.75000    6.75000
6.75006
0.0        0.33330    0.68130    1.09210    1.44790    1.71880    1.85420    1.87500    1.87500    1.87500
1.87500    1.87500    1.87500    1.87080    1.85420    1.79170    1.56670    1.15210    0.76670    0.36870
0.0
2.45830    2.45830    2.46880    2.48330    2.55620    2.82920    3.21250    3.59380    3.92920    4.24170
4.54170    4.83330    5.14170    5.49170    5.88960    6.32500    6.62500    6.62500    6.62500    6.62500
6.62500
0.0        0.31880    0.65630    1.01460    1.40420    1.68750    1.81250    1.86670    1.87500    1.87500
1.87500    1.87500    1.87500    1.86870    1.84380    1.78130    1.51040    1.11040    0.73540    0.35420
0.0
2.54170    2.54170    2.54580    2.55830    2.61670    2.85030    3.22920    3.59380    3.92920    4.24170
4.54170    4.83330    5.14170    5.49170    5.88960    6.31250    6.54170    6.54170    6.54170    6.54170
6.54170
0.0        0.30000    0.61460    0.94580    1.28130    1.56250    1.74170    1.79170    1.79170    1.79170
1.79170    1.79170    1.79170    1.77080    1.65630    1.33330    1.00030    0.67920    0.33330
0.0
2.62500    2.62500    2.63540    2.65210    2.73960    2.93750    3.23750    3.59380    3.91670    4.21460
4.50000    4.76130    5.07290    5.41250    5.78750    6.14580    6.26670    6.31670    6.34380    6.36870
6.37500
0.0        0.27920    0.57080    0.86460    1.18750    1.44880    1.66670    1.72000    1.72000    1.72920
1.72920    1.72920    1.72920    1.72500    1.60750    1.54500    1.23750    0.92920    0.62920    0.30830
0.0
2.70830    2.71040    2.72080    2.75000    2.81670    2.99120    3.24370    3.57290    3.84380    4.17000
4.44790    4.71880    4.96250    5.32290    5.67710    5.98540    6.00120    6.12290    6.15420    6.10120
6.18790
0.0        0.25000    0.52080    0.76330    1.00330    1.35420    1.55210    1.62500    1.64580    1.64580
1.64580    1.64580    1.64580    1.63750    1.59380    1.41250    1.11440    0.83330    0.57080    0.27920
0.0
2.77080    2.77080    2.77710    2.81250    2.86670    3.00000    3.23330    3.52920    3.81870    4.08960
4.35420    4.61250    4.87920    5.18540    5.51460    5.75030    5.42710    5.85030    5.88960    5.91460
5.93750
0.0        0.22080    0.45830    0.70000    0.97920    1.23960    1.41670    1.50830    1.53750    1.55000
1.55210    1.55000    1.54580    1.53130    1.48330    1.26250    0.99580    0.75000    0.51040    0.24580
0.0
2.85420    2.85420    2.85830    2.86870    2.90030    3.01670    3.22500    3.48540    3.75000    4.00000
4.25000    4.49170    4.74370    5.02920    5.32920    5.56470    5.59790    5.62500    5.64170
5.64580
0.0        0.20830    0.42710    0.65830    0.92710    1.17290    1.31670    1.38540    1.41250    1.41670
1.41670    1.41670    1.41670    1.41670    1.35030    1.15030    0.92500    0.69790    0.47080    0.22920
0.0
2.97920    2.97920    2.98120    2.98540    3.00030    3.11040    3.32710    3.58120    3.82290    4.05210
4.28130    4.50210    4.73330    5.00210    5.26670    5.43120    5.50210    5.53750    5.55030    5.57290
5.58330
0.0        0.18750    0.38750    0.59580    0.84380    1.07290    1.19500    1.23540    1.24170    1.25000
1.25000    1.25000    1.25000    1.23960    1.16250    1.01130    0.86790    0.64170    0.43330    0.21250
0.0
3.12500    3.12500    3.12920    3.13540    3.15210    3.24370    3.44580    3.68750    3.90630    4.11250
4.31250    4.50830    4.70830    4.93750    5.15830    5.33330    5.43330    5.46880    5.48750    5.49580
5.50000
0.0        0.17500    0.35000    0.53750    0.75000    0.92710    1.02500    1.06250    1.07290    1.08330
1.08330    1.08330    1.08330    1.07500    1.02500    0.92000    0.77000    0.58330    0.39580    0.19170
0.0
3.29170    3.29170    3.29790    3.30620    3.33330    3.43960    3.61070    3.82710    4.01250    4.19370
4.36460    4.53130    4.70830    4.90630    5.10830    5.27710    5.38540    5.41670    5.43940    5.43750
5.43750
0.0        0.15000    0.30420    0.46670    0.63330    0.78130    0.83540    0.87500    0.88540    0.89170
0.89580    0.89580    0.89580    0.88750    0.85830    0.78130    0.69630    0.51040    0.34580    0.16670
0.0
3.45830    3.46460    3.47500    3.49580    3.53540    3.62710    3.78750    3.96250    4.11870    4.26460
4.40630    4.54790    4.69170    4.85420    5.02920    5.18540    5.27290    5.32290    5.33960    5.34790
5.35420
0.0        0.13330    0.26460    0.38540    0.50000    0.59380    0.65030    0.69170    0.70420    0.70030
0.70830    0.70830    0.70830    0.69790    0.65630    0.60000    0.51670    0.41670    0.29790    0.14580
0.0
3.64580    3.65630    3.67920    3.72500    3.79170    3.88540    3.99790    4.12710    4.25000    4.36460
4.47920    4.58750    4.70420    4.83120    4.95620    5.16440    5.23330    5.27920    5.30620
5.31250
0.0        0.10830    0.20830    0.29170    0.35420    0.40650    0.41870    0.46250    0.47500    0.48960
0.48960    0.48960    0.48960    0.48330    0.46880    0.43330    0.38750    0.31870    0.23330    0.11670
0.0
3.83330    3.83960    3.89170    3.95000    4.11460    4.20210    4.28540    4.36460    4.44580
4.52080    4.59580    4.67500    4.78460    4.86040    4.95420    5.03130    5.09580    5.15000    5.20000
5.20030
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
0.0
4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170
4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170    4.79170
4.79170
20.23000   0.0        5.31300    4.16670    24.45999   0.0        9.97000    2.29170
0.0        2.90000    5.00000    10.00000   20.00000   30.00000   40.00000   60.00000   80.00000   100.00000
0.0        1.96100    2.66600    3.51200    4.30300    4.50100    4.35200    3.42300    1.96700    0.09500
20.23000   0.25000    4.77000    4.16670    29.91800   5.03330    4.81170    2.75000
0.0        2.50000    5.00000    10.00000   20.00000   30.00000   40.00000   60.00000   80.00000   100.00000
0.0        1.96100    2.66600    3.51200    4.30300    4.50100    4.35200    3.42300    1.96700    0.09500
```

PLOT DATA

```
Y Z OUT   -45.00000   10.00000   -30.00000    0.0        0.0        0.0        0.0        0.0     7.50000ORT
X Z OUT    45.00000   10.00000    30.00000    0.0        0.0        0.0        0.0        0.0    10.00000ORT
```

Figure D-3. Continued.
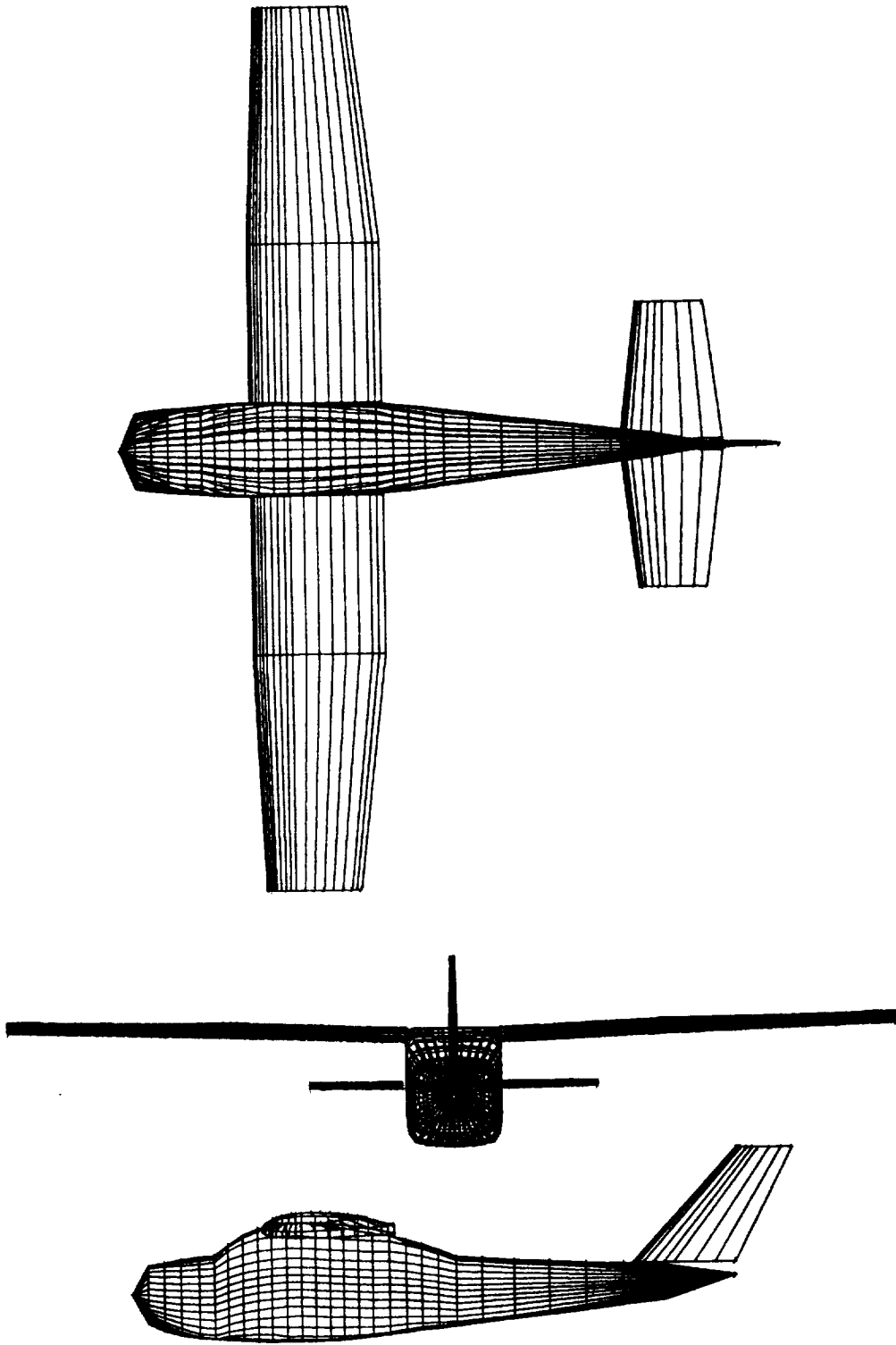
Figure D-4. Plotted 3-view of the Cessna 182.

358
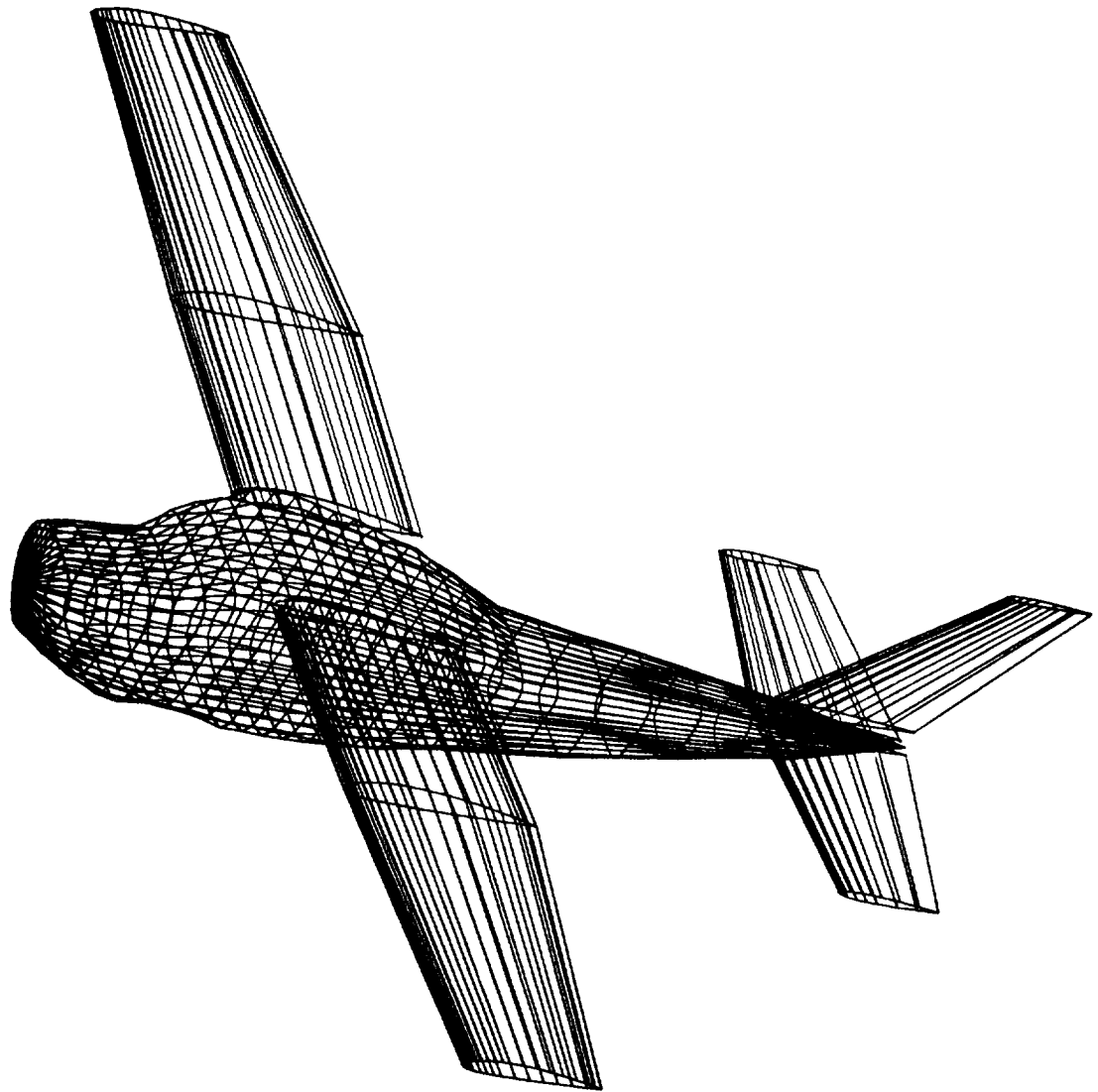
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS — COMPLETE AIRPLANE

X Z   −45. 10. −30.   8.5 ORT

Figure D-5. Orthographic projection of a Cessna 182 rolled −45°,
             pitched 10° and yawed −30° with respect to the
             X-Z plane of symmetry.
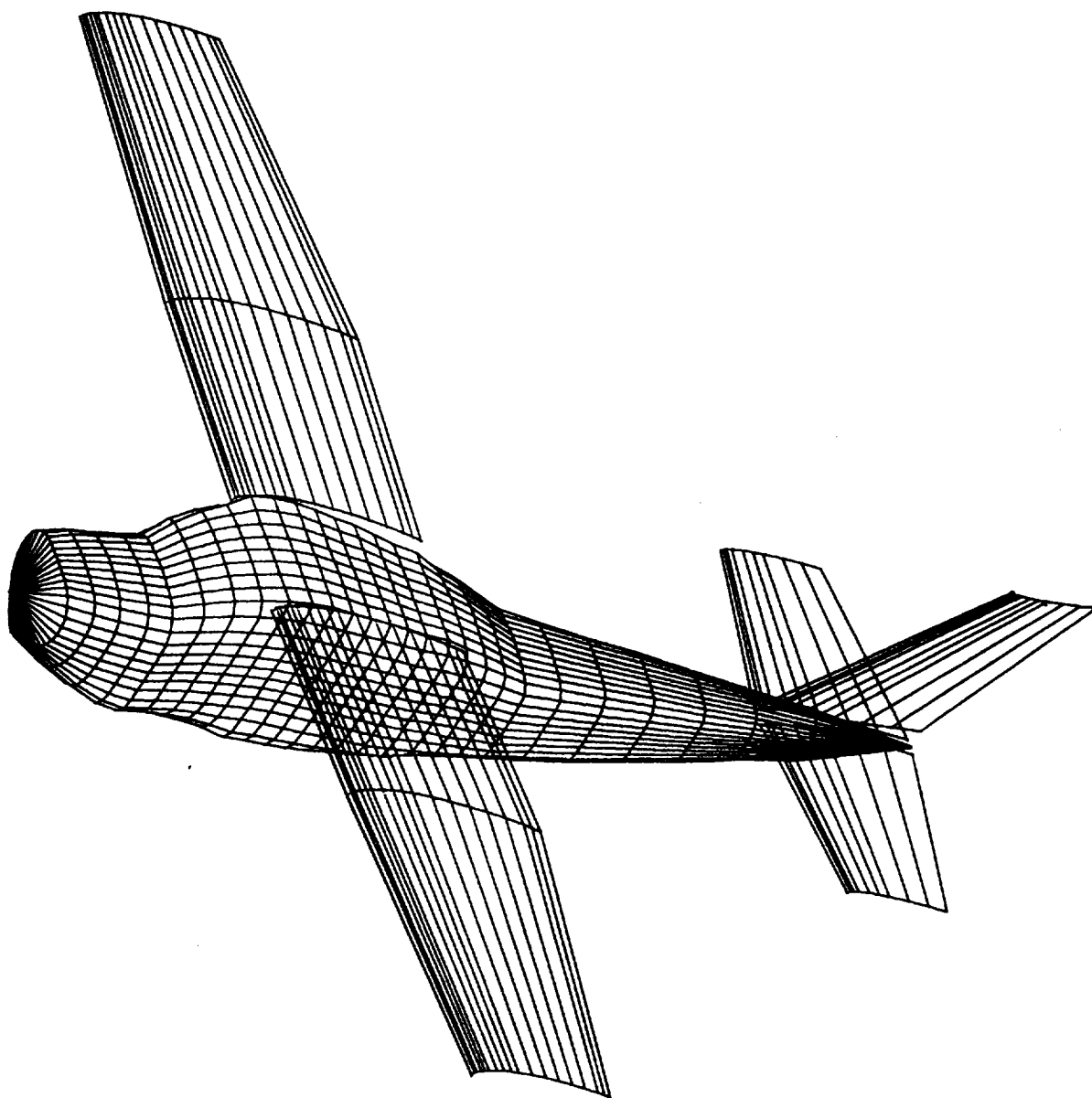
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS — COMPLETE AIRPLANE

X Z OUT 45. 10.   30.   9.25 ORT

Figure D-6. Orthographic projection with hidden lines removed of a
Cessna 182 rolled 45°, pitched 10°, and yawed 30° with
respect to the X-Z plane of symmetry.
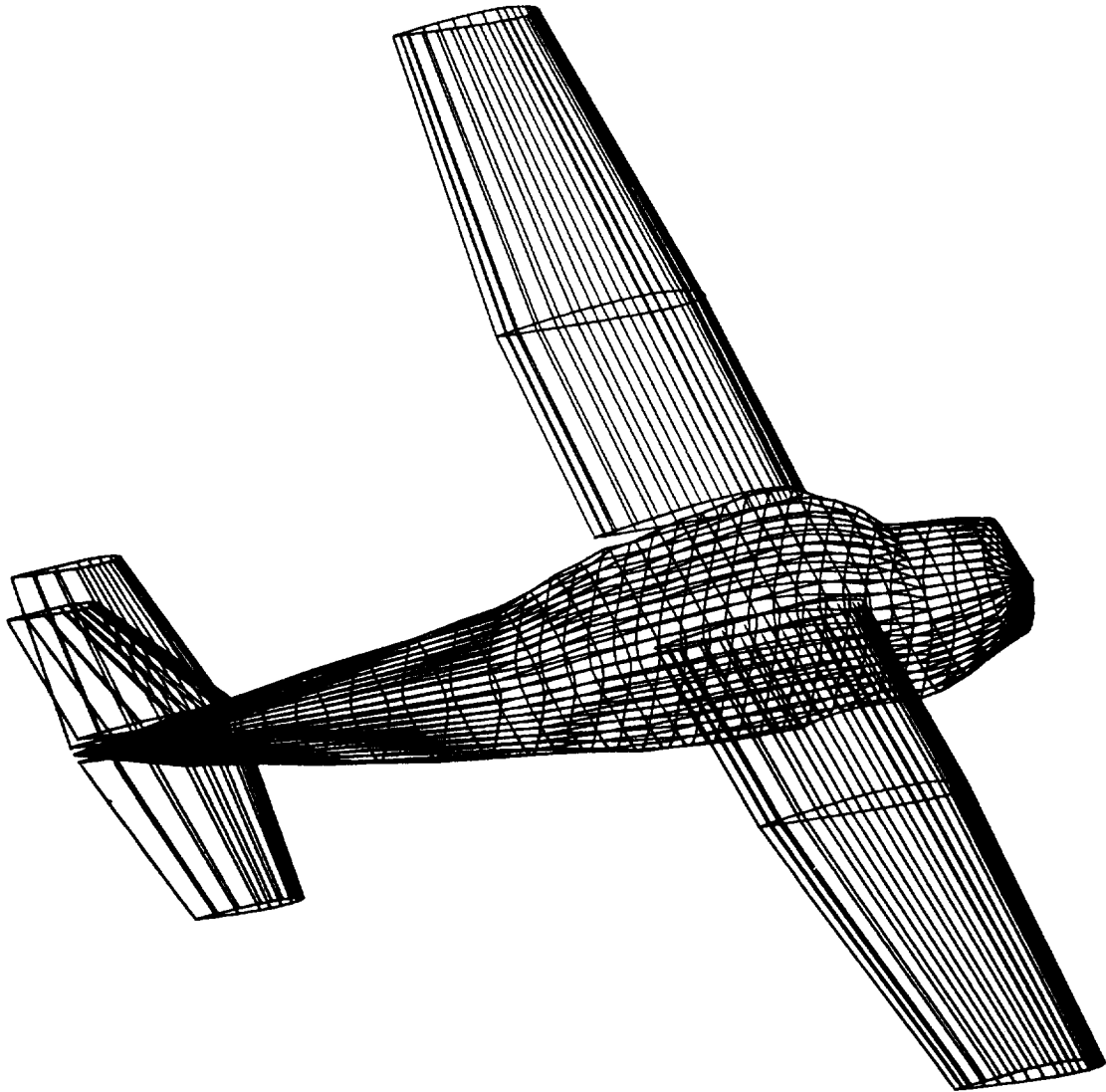
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS — COMPLETE AIRPLANE

X Z   45. 10. 160.   8.5 ORT

Figure D-7. Orthographic projection of a Cessna 182 rolled 45°,
pitched 10°, and yawed 160° with respect to the X-Z
plane of symmetry.
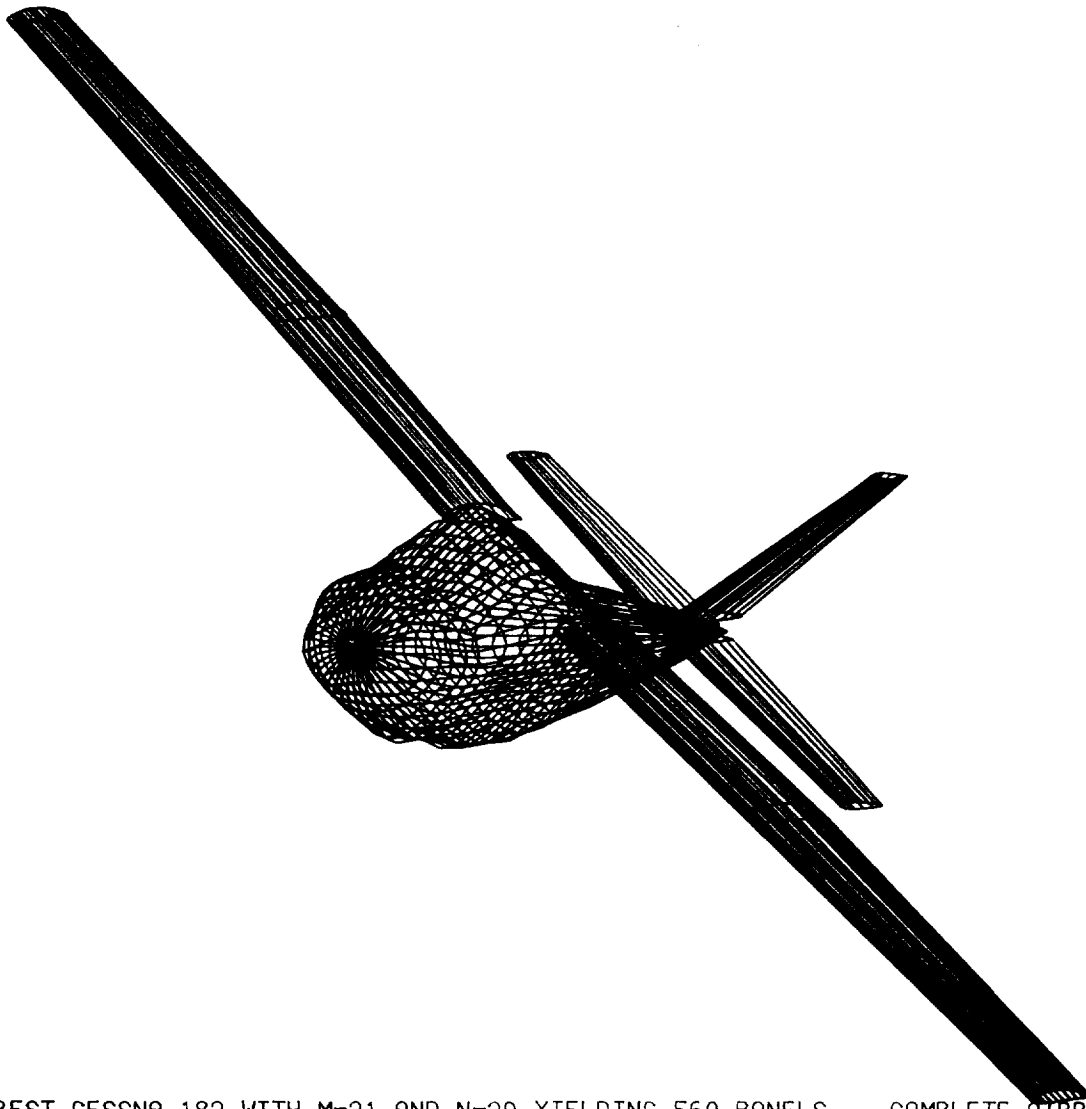
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS –– COMPLETE AIRPLANE

X Z   –45.   0. –70.   8.5 ORT

Figure D-8. Orthographic projection of a Cessna 182 rolled -45°
            and yawed -70° with respect to the X-Z plane of
            symmetry.
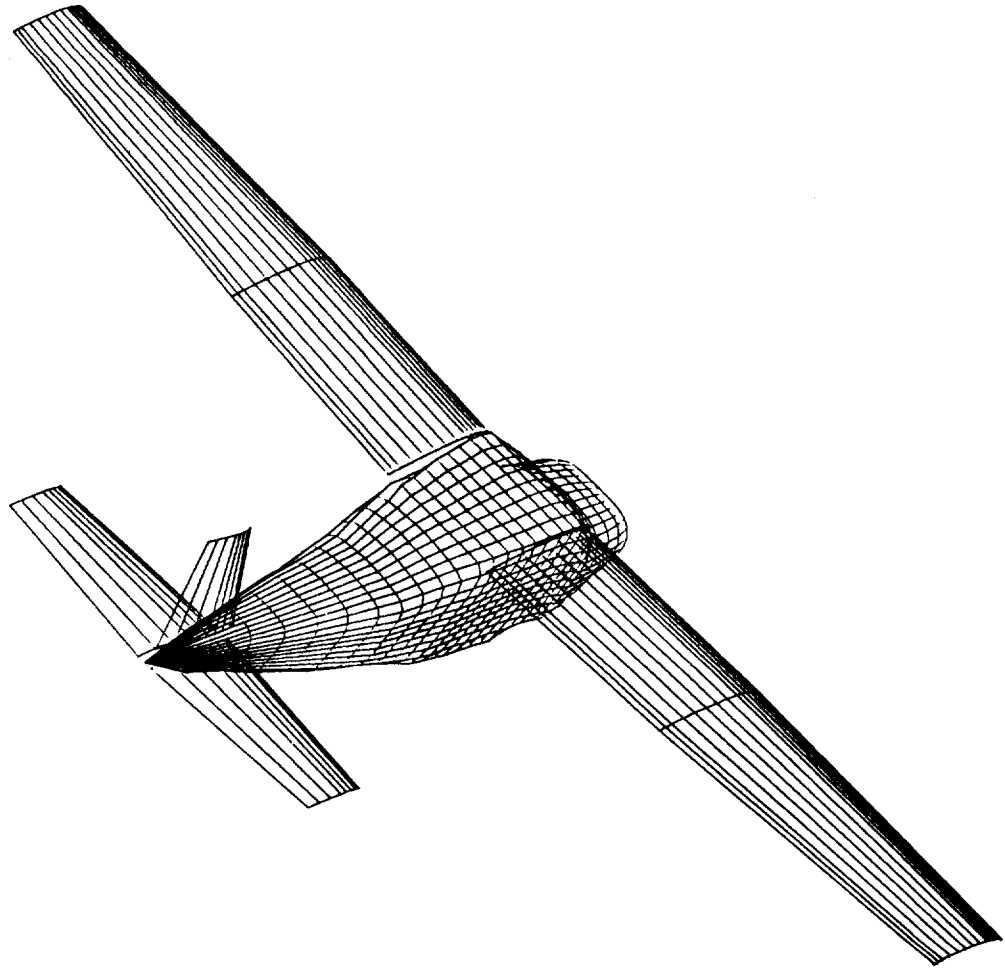
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS   - COMPLETE AIRPLANE

Y Z OUT -45. 10. -30   7.5 ORT

Figure D-9. Orthographic projection of a Cessna 182 rolled -45°,
             pitched 10°, and yawed -30° with respect to the
             Y-Z plane.
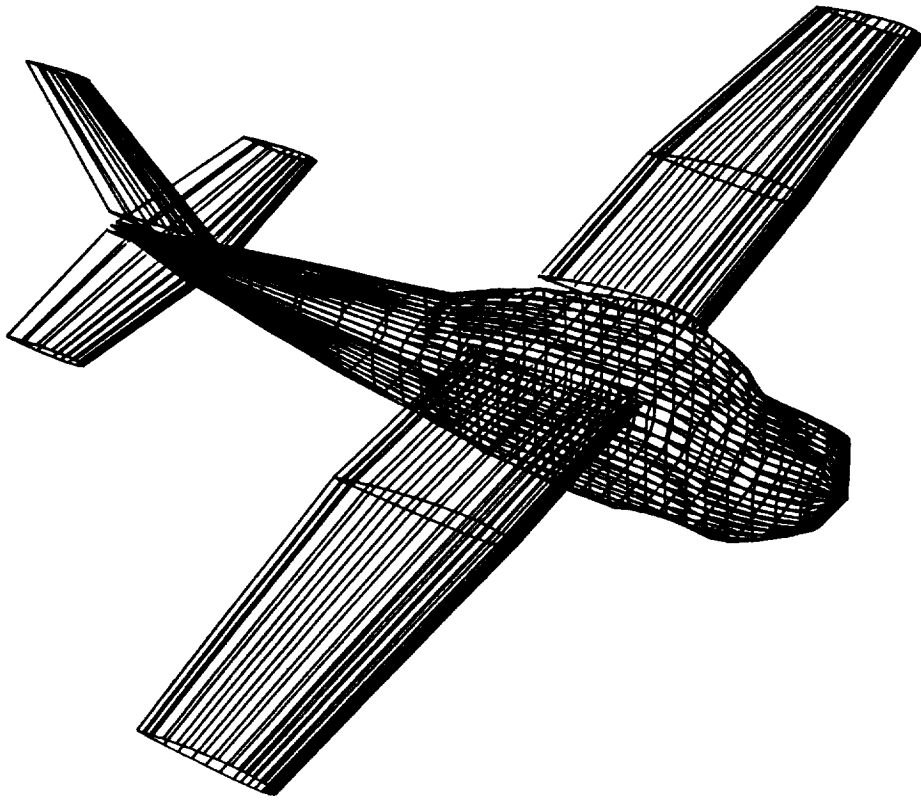
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- COMPLETE AIRPLANE

-20. 50. 50. 7.0 0.0 5.0 14.0 1.0 10. PER

Figure D-10. Perspective view number 1 of the Cessna 182.
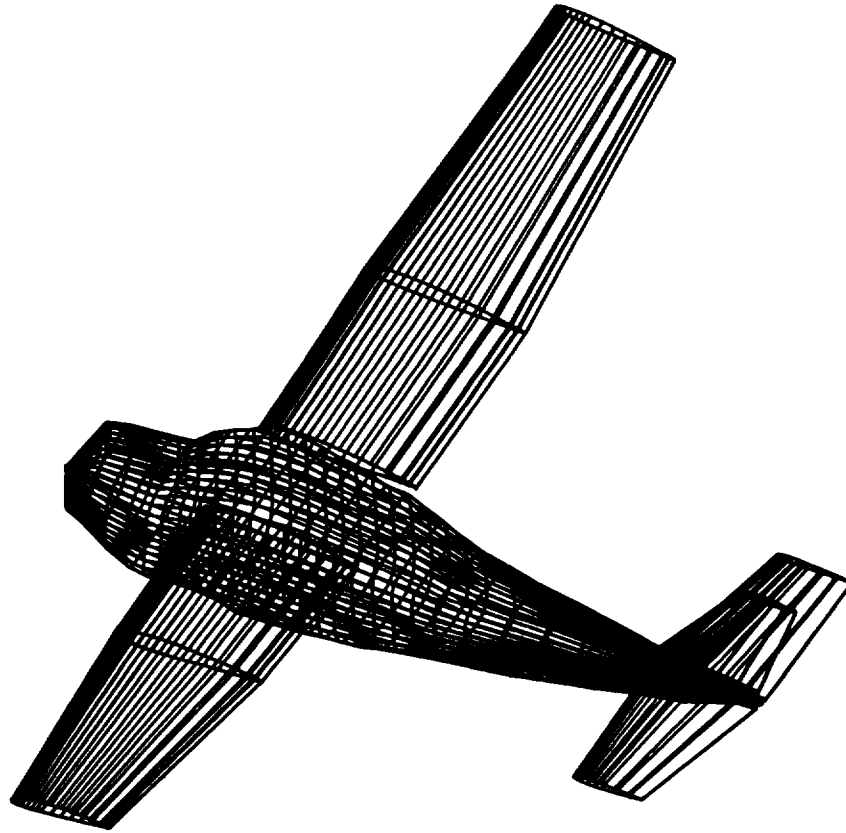
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS — COMPLETE AIRPLANE

-20. -50. -50. 7.0 0.0 5.0 14.0 1.0 10. PER

Figure D-11.   Perspective view number 2 of the Cessna 182
(The reader should note that the viewer is
under the aircraft looking up.)

BEST CESSNA 182 WITH M=21 AND N=29
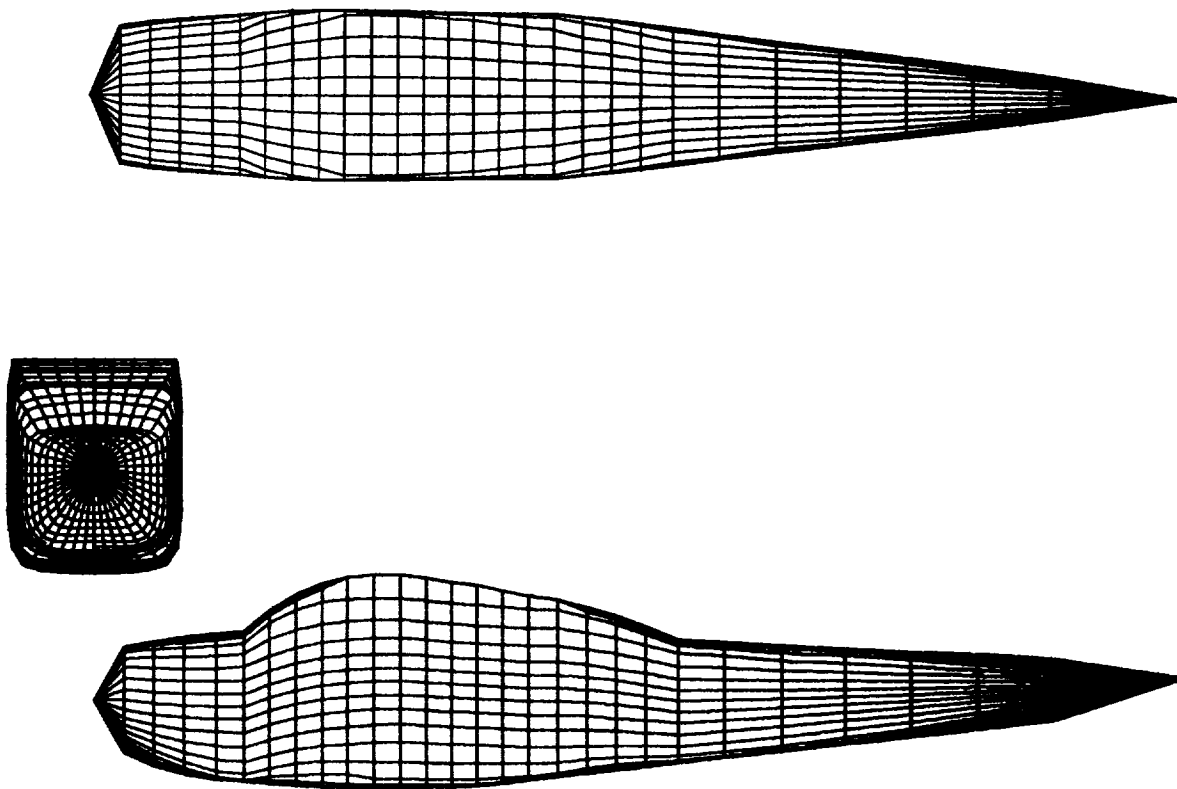
Figure D-12. Plotted 3-view of the Cessna 182 fuselage.
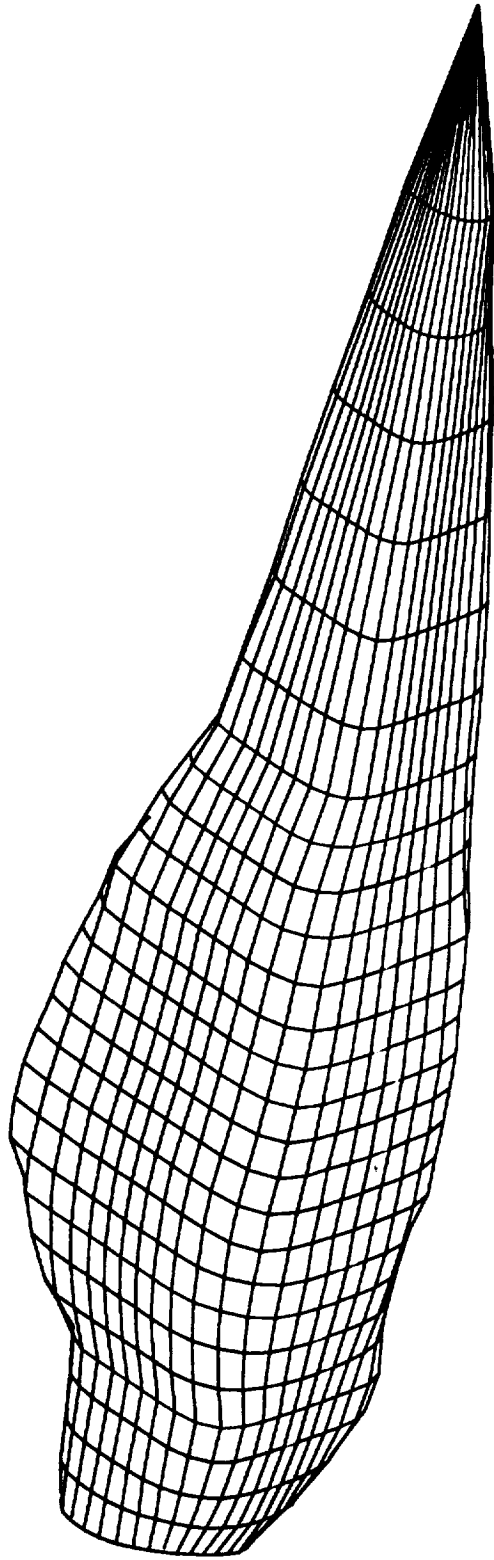
X Z OUT -45. 10.   -30.   10. ORT
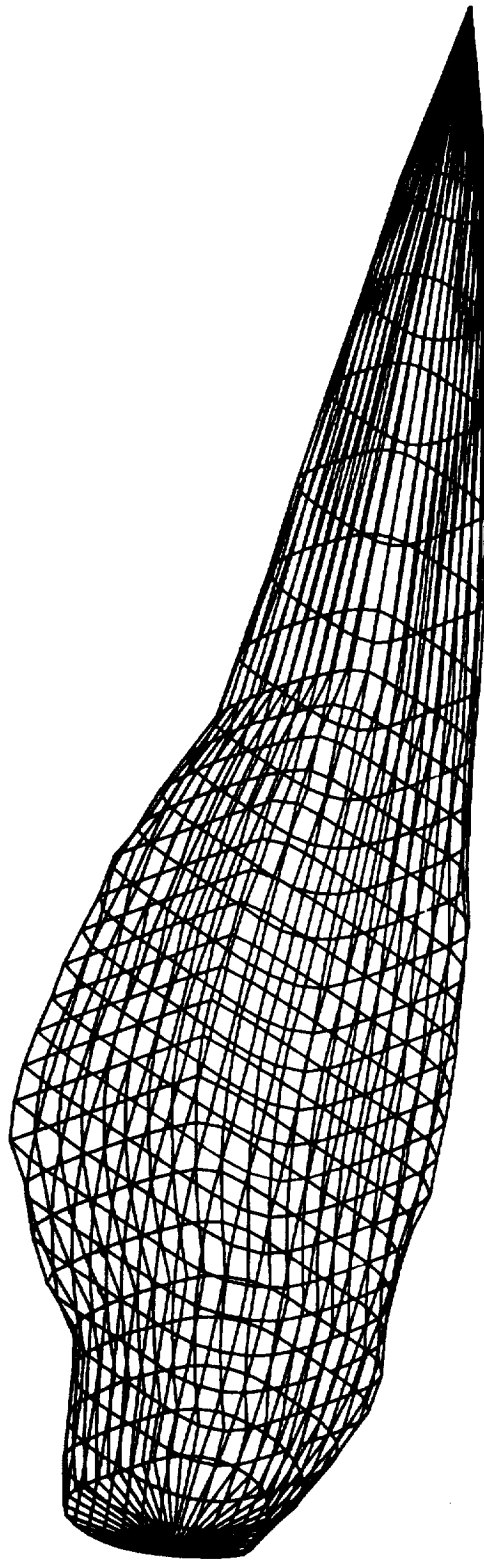
X Z   -45. 10.   -30.   10. ORT

Figure D-13. Two orthographic projections of a Cessna 182 fuselage rolled -45°, pitched 10°, and yawed -30° with respect to the X-Z plane of symmetry. The top view has hidden lines removed.

367
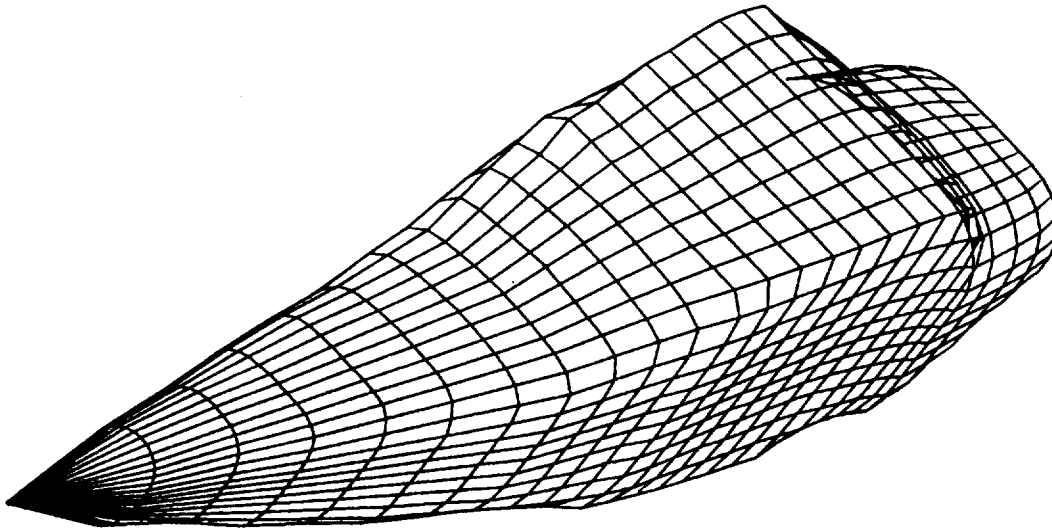
BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY

Y Z OUT -45. 10. -30. 11. ORT

Figure D-14. Orthographic projection with hidden lines removed of a
Cessna 182 fuselage rolled -45°, pitched 10°, and yawed
-30° with respect to the Y-Z plane.

APPENDIX E - CONVERT Program

User Instructions

This program is written in FORTRAN IV and is designed to run in single precision on an IBM 370-165 computer. Given a set of plot input data as described in Appendix D, this program (1) produces a properly indexed data set for the NCSU BODY program, (2) computes the area of each body panel described by the input points and displays each area in an orderly fashion, and (3) displays the ratio of the area of each panel to the area of the panel below it and the panel to its right. It should be noted that while the CONVERT program produces a data set for the aircraft body, it will accept the input of a data set for a complete aircraft configuration and ignore the un- necessary information (wing, tail, nacelles, etc.). Execution requires 92,000 bytes of core storage and approximately 15 seconds to run a case with 560 panels specified. A description of the input data cards is not included here since it is the same as that for the PLOT program in Appendix D. The format specification is also the same. The program listing is given in the next section of this appendix and is followed by the sample output (Figure E-1) corresponding to the Cessna 182 data set given in Figure D-3.

```
C***   THIS PROGRAM CONVERTS A SET OF CRAIDON INPUT DATA INTO A DATA SET        CVT    1
C***   WHICH CAN BE USED WITH THE XYZ POTENTIAL FLOW PROGRAM FOR A BODY.        CVT    2
       DIMENSION DUM(30),DUMMY(30),TITLE(20),NRADK(4),NPFORX(4),ZPFUS(30),Z     CVT    3
      1FUSAR(30),XI(1500),YI(1500),ZI(1500),XX(1500),YY(1500),ZZ(1500),ID(30    CVT    4
      1,50),AQ(50,30),NWRITE(50),RATIO(50,30),RATIOM(50,30)                     CVT    5
       IREAD=1                                                                  CVT    6
       IPUNCH=2                                                                 CVT    7
       IWRITE=3                                                                 CVT    8
C***   ONLY POSITIVE Y-VALUES ARE PERMITTED                                     CVT    9
  5    READ (IREAD,10) TITLE                                                    CVT   10
       WRITE (IWRITE,15) TITLE                                                  CVT   11
  10   FORMAT (20A4)                                                           CVT   12
  15   FORMAT (1X,20A4)                                                        CVT   13
       READ (IREAD,20) J0,J1,J2,J3,J4,J5,J6,NWAF,NWAFOR,NFUS,(NRADX(I),NF       CVT   14
      1ORX(I),I=1,4),NP,NPODOR,NF,NFINOR,NCAM,NCANOR                            CVT   15
  20   FORMAT (2413)                                                           CVT   16
       WRITE (IWRITE,25) J0,J1,J2,J3,J4,J5,J6,NWAF,NWAFOR,NFUS,(NRADX(I),       CVT   17
      1NPFORX(I),I=1,4),NP,NPODOR,NF,NFINOR,NCAM,NCANOR                         CVT   18
  25   FORMAT (1X,2413)                                                        CVT   19
  30   FORMAT (10F7.4)                                                         CVT   20
  35   FORMAT (1X,10F7.4)                                                      CVT   21
C***   FIND OUT IF REFERENCE AREA WAS INPUT                                     CVT   22
       IF (J0.NE.1) GO TO 40                                                    CVT   23
       READ (IREAD,30) DUM(1)                                                   CVT   24
C***   SEE IF WING DATA IS INPUT                                                CVT   25
  40   IF (J1.EQ.0.OR.J1.EQ.2) GO TO 65                                         CVT   26
       READ (IREAD,30) (DUM(I),I=1,NWAFOR)                                      CVT   27
  45   READ (IREAD,30) (DUM(I),I=1,4)                                           CVT   28
C***   SEE IF CAMBERED DATA IS TO BE READ                                       CVT   29
       IF (J1.NE.1) GO TO 55                                                    CVT   30
  50   READ (IREAD,30) (DUM(I),I=1,NWAFOR)                                      CVT   32
C***   READ WING AIRFOIL ORDINATES                                             CVT   34
  55   DO 60 I=1,NWAF                                                           CVT   35
  60   READ (IREAD,30) (DUM(I),I=1,NWAFOR)                                      CVT   36
C***   NOW REALLY READ FUSELAGE INPUT                                          CVT   37
       IF (J2.EQ.0.OR.J2.EQ.2) GO TO 230                                       CVT   38
       XMAX, XMIN, ZMAX, AND ZMIN ARE FOUND IN ORDER TO PUT THE ORGIN FOR      CVT   39
C***   XYZ NEAR THE CENTER OF THE BODY.                                        CVT   40
  65   XMAX=-1000.                                                              CVT   41
       XMIN=-1000.                                                              CVT   42
       ZMAX=-1000.                                                              CVT   43
       ZMIN=-1000.                                                              CVT   44
       K=0                                                                     CVT   45
       L=0                                                                     CVT   46
       DO 105 I=1,NFUS                                                          CVT   47
C***   READ FUSELAGE ORDINATES                                                CVT   48
       JJ=NPFORX(I)                                                            CVT   49
       READ (IREAD,30) (DUM(I),I=1,JJ)                                          CVT   51
       WRITE (IWRITE,35) (DUM(I),I=1,JJ)                                        CVT   52
       JK=NRADX(I)                                                              CVT   53
       DO 70 I=1,JJ                                                             CVT   54
       IF (XMAX.LT.DUM(I))XMAX=DUM(I)                                           CVT   55
       IF (XMIN.GT.DUM(I))XMIN=DUM(I)                                           CVT   56
  70   DO 70 J=1,JK                                                             CVT   57
       K=K+1                                                                   CVT   58
C***   SEE IF FUSELAGE IS ARBITRARY                                            CVT   59
  70   XI(K)=DUM(I)                                                             CVT   60
       IF (J2.EQ.1) GO TO 80                                                    CVT   61
C***   COMPLETE SYMMETRY W.R.T. XY PLANE IS NOT HANDLED IN THIS PROGRAM         CVT   61
C***   AS IT IS IN THE CRAIDON PROGRAM (J6 NOT=1)                               CVT   62
C***   CHECK TO SEE IF FUSELAGE IS CAMBERED (J6=0)                              CVT   63
       IF (J6.EQ.-1) GO TO 75                                                   CVT   64
C***   J6=0 CORRESPONDS TO A CAMBERED FUSELAGE THUS THE Z LOCATIONS OF          CVT   65
C***   THE CENTERS ARE READ                                                    CVT   66
       READ (IREAD,30) (ZFUS(I),I=1,JJ)                                         CVT   67
       WRITE (IWRITE,35) (ZFUS(I),I=1,JJ)                                       CVT   68
  75   READ (IREAD,30) (ZFUSAR(I),I=1,JJ)                                       CVT   69
       WRITE (IWRITE,35) (ZFUSAR(I),I=1,JJ)                                     CVT   70
       GO TO 90                                                                 CVT   71
  80   DO 85 II=1,JJ                                                            CVT   72
       READ (IREAD,30) (DUM(I),I=1,JK)                                          CVT   73
       WRITE (IWRITE,35) (DUM(I),I=1,JK)                                        CVT   74
       READ (IREAD,30) (DUMMY(I),I=1,JK)                                        CVT   75
       WRITE (IWRITE,35) (DUMMY(I),I=1,JK)                                      CVT   76
       DO 85 I=1,JK                                                             CVT   77
       L=L+1                                                                   CVT   78
       YI(L)=DUM(I)                                                             CVT   79
       ZI(L)=DUMMY(I)                                                           CVT   80
       IF (ZMAX.LT.ZI(L))ZMAX=ZI(L)                                            CVT   81
  85   IF (ZMIN.GT.ZI(L))ZMIN=ZI(L)                                            CVT   82
C***   SEE IF CIRCULAR FUSELAGE WAS SPECIFIED                                  CVT   83
       GO TO 105                                                                CVT   84
  90   NTHETA=NRADX(I)-1                                                        CVT   85
       DTHETA=(180./NTHETA)/57.2958                                            CVT   86
       THETA=-90./57.2958                                                      CVT   87
       RSQ=ZFUSAR(I)/3.1415927                                                 CVT   88
       R=SQRT(RSQ)                                                             CVT   89
       DO 100 I=1,JK                                                            CVT   90
       L=L+1                                                                   CVT   91
       YI(L)=R*COS(THETA)                                                      CVT   92
       ZI(L)=R*SIN(THETA)                                                      CVT   93
       IF (J6.NE.0) GO TO 95                                                    CVT   94
  95   ZI(L)=ZI(L)+ZFUS(J)                                                     CVT   95
       IF (ZMAX.LT.ZI(L))ZMAX=ZI(L)                                            CVT   96
       IF (ZMIN.GT.ZI(L))ZMIN=ZI(L)                                            CVT   97
 100   THETA=THETA+DTHETA                                                      CVT   98
 105   CONTINUE                                                                CVT   99
C***   NOW LETS SET UP XP,YP, AND ZP ARRAYS FOR PROGRAM XYZ                    CVT  100
       NS=1                                                                    CVT  101
       XORG=(XMIN+XMAX)/2.                                                     CVT  102
       ZORG=(ZMIN+ZMAX)/2.                                                     CVT  103
       DO 110 I=1,50                                                           CVT  104
       DO 110 J=1,30                                                           CVT  105
 110   ID(J,I)=0                                                               CVT  106
       WRITE (IWRITE,15) TITLE                                                 CVT  107
       WRITE (IPUNCH,10) TITLE                                                 CVT  108
       L=NPFORX(1)+NPFORX(2)+NPFORX(3)+NPFORX(4)                               CVT  109
       NQE=(L-1)*(NRADX(I)-1)                                                   CVT  110
       WRITE (IWRITE,115) NQE                                                  CVT  111
 115   FORMAT (//,1X,5HNQE=,I4,//)                                             CVT  112
       WRITE (IPUNCH,120) NQE                                                  CVT  113
 120   FORMAT (1X,I5)                                                          CVT  114
       WRITE (IWRITE,125)                                                      CVT  115
 125   FORMAT (///,7X,2HXP,10X,2HYP,10X,2HZP,6X,2HN1,2X,2HN5,///)              CVT  116
       DO 130 I=1,K                                                            CVT  117
       XP=XI(I)-XORG                                                           CVT  118
       YP=YI(I)                                                                CVT  119
```

```
      ZP=Z(I)-ZORG
      XX(I)=XP
      YY(I)=YP
      ZZ(I)=ZP
  130 CONTINUE
      LL=0
      DO 145 I=1,LM
      DO 145 J=I,JK
      LL=LL+1
C***  PRINT DATA FOR VISCOUS FLOW PROGRAM
      WRITE (IWRITE,135) XX(LL),YY(LL),ZZ(LL),I,J,NS
  135 FORMAT (1X,3F12.7,3I4)
C***  PUNCH DATA FOR VISCOUS FLOW PROGRAM
      WRITE (IPUNCH,140) XX(LL),YY(LL),ZZ(LL),I,J,NS
  140 FORMAT (3F12.7,3I2X,I2))
  145 ID(J,I)=LL
      DO 150 N=1,LM
      DO 150 M=1,JK
      P1=ID(M,N)
      P2=ID(M+1,N)
      P3=ID(M,N+1)
      P4=ID(M+1,N+1)
      IT=ID(M,N)*ID(M+1,N)*ID(M,N+1)*ID(M+1,N+1)
      IF (IT.EQ.0) GO TO 150
      X1=XX(P3)-XX(P1)
      X2=XX(P4)-XX(P2)
      Y1=YY(P3)-YY(P1)
      Y2=YY(P4)-YY(P2)
      Z1=ZZ(P3)-ZZ(P1)
      Z2=ZZ(P4)-ZZ(P2)
      XN=Y2*Z1-Y1*Z2
      YN=X1*Z2-X2*Z1
      ZN=X2*Y1-X1*Y2
      R=SQ2(XN,YN,ZN)
  150 AQ(N,M)=0.5*R
      MDD=LM-1
      MDO=JK-1
      DO 170 N=1,NDO
      DO 170 M=1,MDO
      IF (M.EQ.1) GO TO 155
      RATIO(N,M)=AQ(N,M)/AQ(N-1,M)
      GO TO 160
  155 RATIO(N,M)=1.0
  160 IF (M.EQ.1) GO TO 165
      RATIO(N,M)=AQ(N,M)/AQ(N,M-1)
      GO TO 170
  165 RATIO(N,M)=1.0
  170 CONTINUE
      WRITE (IWRITE,225)
      WRITE (IWRITE,220)
      LINES=5
      MCOUNT=10
      NSTART=1
  175 NSTOP=MIN0(MCOUNT,NDO)
      DO 180 M=NSTART,NSTOP
  180 NWRITE(I)=I
C***  OUTPUT AREAS AND AREA RATIOS FOR QUADRILATERALS
      DO 185 I=1,MDO
      WRITE (IWRITE,200) (I,NWRITE(J),J=NSTART,NSTOP)
      WRITE (IWRITE,205) (AQ(J,I),J=NSTART,NSTOP)
      WRITE (IWRITE,210) (RATION(J,I),J=NSTART,NSTOP)
  185 WRITE (IWRITE,215) (RATION(J,I),J=NSTART,NSTOP)
      WRITE (IWRITE,220)
      IF (NSTOP.EQ.NDO) GO TO 195
      NSTART=NSTART+10
      MCOUNT=MCOUNT+10
      LINES=LINES+MDO+5
      IF (LINES.GT.50) GO TO 190
      GO TO 175
  190 WRITE (IWRITE,225)
      LINES=5
      GO TO 175
  195 CONTINUE
  200 FORMAT (/,1X,10(2X,I2,1H ,I2,5X))
  205 FORMAT (1X,10(E10.3,2X))
  210 FORMAT (1X,10(1X,E9.2,2X))
  215 FORMAT (1X,10(1X,E9.2,2X),/)
  220 FORMAT (1X,115(1H-))
  225 FORMAT (1H1,1X,10(1X,5HIM,N),6X),/,1X,10(2X,4HAREA,6X),/,1X,10(2X,
     16HRATION,4X),/,1X,10(2X,6HRATION,4X))
C***  SEE IF PODS WERE SPECIFIED
  230 IF (J3.EQ.0.OR.J3.EQ.2) GO TO 240
      DO 235 I=1,NP
      READ (IREAD,30) (DUM(I),I=1,3)
      READ (IREAD,30) (DUM(I),I=1,NPODOR)
  235 READ (IREAD,30) (DUM(I),I=1,NPODOR)
C***  SEE IF FINS WERE SPECIFIED
  240 IF (J4.EQ.0.OR.J4.EQ.2) GO TO 250
      DO 245 I=1,NF
      READ (IREAD,30) (DUM(I),I=1,8)
      READ (IREAD,30) (DUM(I),I=1,NFINOR)
  245 READ (IREAD,30) (DUM(I),I=1,NFINOR)
C***  SEE IF CANARDS WERE SPECIFIED
  250 IF (J5.EQ.0.OR.J5.EQ.2) GO TO 260
      NCAN1=IABS(NCAN)
      DO 255 I=1,NCAN1
      READ (IREAD,30) (DUM(I),I=1,8)
      READ (IREAD,30) (DUM(I),I=1,NCANOR)
      READ (IREAD,30) (DUM(I),I=1,NCANOR)
      IF (NCAN.LT.0) READ (1,200)(DUM(I),I=1,NCANOR)
  255 CONTINUE
  260 READ (IREAD,265, END=275) KODE
  265 FORMAT (71X,I1)
      IF (KODE.NE.1) GO TO 260
      WRITE (IPUNCH,270)
  270 FORMAT (80X)
      GO TO 5
  275 WRITE (IPUNCH,270)
      CALL EXIT
      END

      FUNCTION SQ2(X,Y,Z)
C***  COMPUTE SQUARE ROOT OF R**2
      R=ABS(X)+ABS(Y)+ABS(Z)+.00000000000001
      RS=X**2+Y**2+Z**2
      R=R+RS/R
      R=0.25*R+RS/R
      R=R+RS/R
      SQ2=.25*R+RS/R
      RETURN
      END
```

371

Figure E-1. CONVERT program sample output of the Cessna 182
with 560 panels describing the half-body.

Figure E-1. Continued.

Figure E-1. Continued.

Figure E-1. Continued.

APPENDIX F- NCSU BODY Program

User Instructions

The program is written in FORTRAN IV and is designed to run in single precision on an IBM 370-165 computer with an execution time of 10-12 minutes for a half-body with 560 panels (1 minute for a half-body with 100 panels). The 560 panel case required 250,000 bytes of core storage. The program calculates an approximate solution of the three-dimensional viscous flow over an arbitrary body and estimates the body lift and drag coefficients. The program was obtained by making major modifications to the XYZ potential flow program (Ref. 94) supplied by the Naval Ship Research and Development Center, Bethesda, Maryland. The data input and program logic modifications were employed to specialize the program for light aircraft fuselages; therefore, this modified program no longer has the design capability of the original XYZ program. Figure F-1 illustrates how the body ordinates should be input to provide the correct body orientation with respect to the flow direction.

The program input data specification is based on the description given in Reference 94 except where changes were made. The input consists of an identification card, two parameter cards, and several body point cards.[†] A description of the program input is given below.

    Card 1 - Identification - Card 1 contains any information to identify the problem in columns 1 through 80.

    Card 2 - Flow Control Variables - Card 2 contains 3 variables which determine the flow Reynolds Number, the reference area upon which the co-efficients are based, and an output control parameter. This is the only card which must be inserted into a data set produced by the CONVERT program in Appendix E.

Parameter	Column	Description
VINF	1-10	Reference free stream velocity in ft./sec. if the body input points are specified in feet (in general it will be units/second where units are the units in which the body input points are specified).
VO	11-20	Kinematic viscosity of the fluid in which the body is moving in $ft.^2/sec.$ if the velocity is specified in ft./sec.

[†]All of these cards except one (card 2) is supplied by the CONVERT program given in Appendix E.

376

Parameter	Column	Description
ROE	21-30	The density of the fluid in which the body is moving in slugs/ft.^{3}
REFA	31-40	The reference area upon which the aerodynamic coefficients will be based in ft.^{2}
IWRITE	45	Control variable which denotes the amount of output the user desires. IWRITE=0 yields maximum output. IWRITE=1 deletes information given for each input point. IWRITE=2 deletes streamline and boundary layer information as well as input point information. (See Sample Output).

Card 3 - Control Integer - Card 3 contains a control integer which must be right justified.

Parameter	Column	Description
NQE	1-4	Number of quadrilateral panels to be specified by the point cards. The value of NQE should generally be less than 600 (see page 244).

Cards 4, 5, . . . - Point Cards - Each point card contains the following information for one point on the body surface. If one section is used for the fuselage there should be P points specified where P is equal to the maximum MI times the maximum NI.

Parameter	Column	Description
XI	1-12	X-coordinate of the input point.
YI	13-24	Y-coordinate of the input point.
ZI	25-36	Z-coordinate of the input point.
NI	39-40	N body station index (see Figure F-2). For plotting purposes NI≤30.
MI	43-44	M body station index (see Figure F-2). For plotting purposes MI≤30.
NS	45-48	Section identification number. The CONVERT program (Appendix E) supplies a one-section data set with a section number of 1.

<u>Last Card - End of Data Set</u> - The last card is a blank card to signify the last card of a particular data set.

It should be noted that the program can be run for more than one data set; the user must simply put the complete data sets he desires to analyze in consecutive order. For more information about this program the user should see pages 243 - 255 which describe the modifications made to the original program as well as Reference 94 which describes the original program.

Specification of the cards above represent a complete set of data for a particular body. The format specification for this data is given in Figure F-3. A sample data set of a prolate spheroid is shown in Figure F-4. Portions of the sample output for the light aircraft data set shown in Figure D-3 and plotted in Appendix D are given in Figure F-5. The light aircraft data set was not used in Figure F-4 as the sample data set because of the excessive length of the light aircraft data set (in excess of 600 cards).
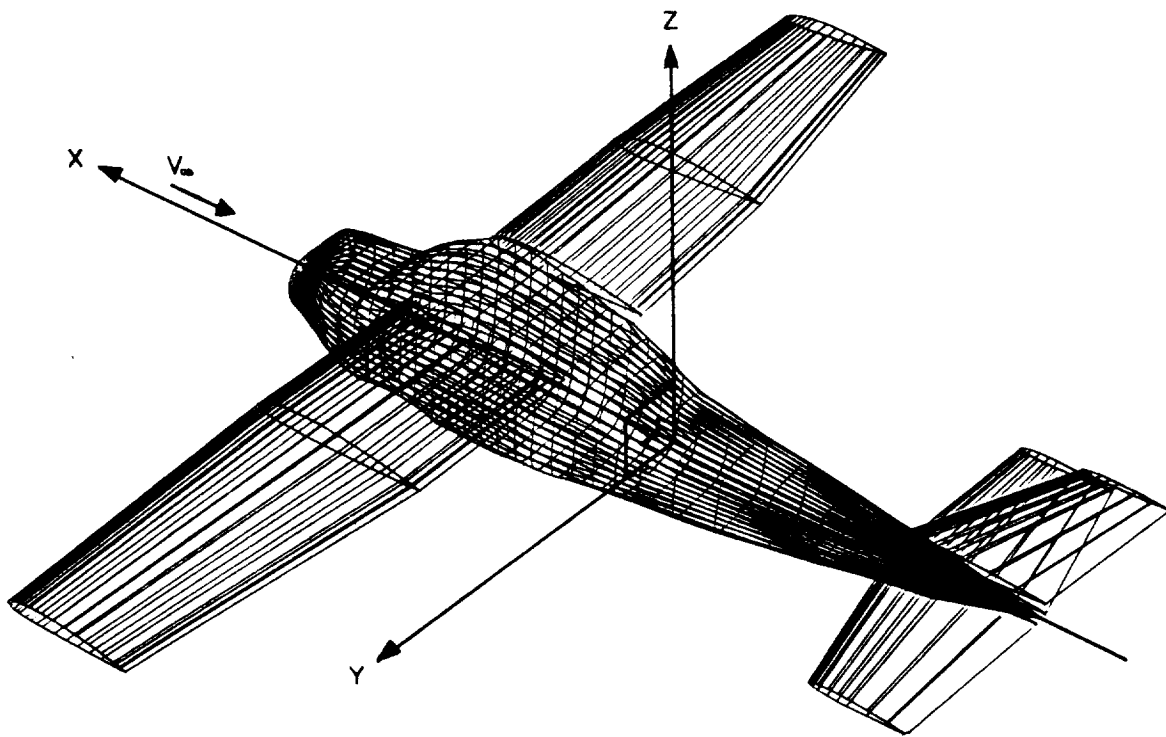
Figure F-1. Orientation of body with respect to body reference axes for the NCSU BODY program.
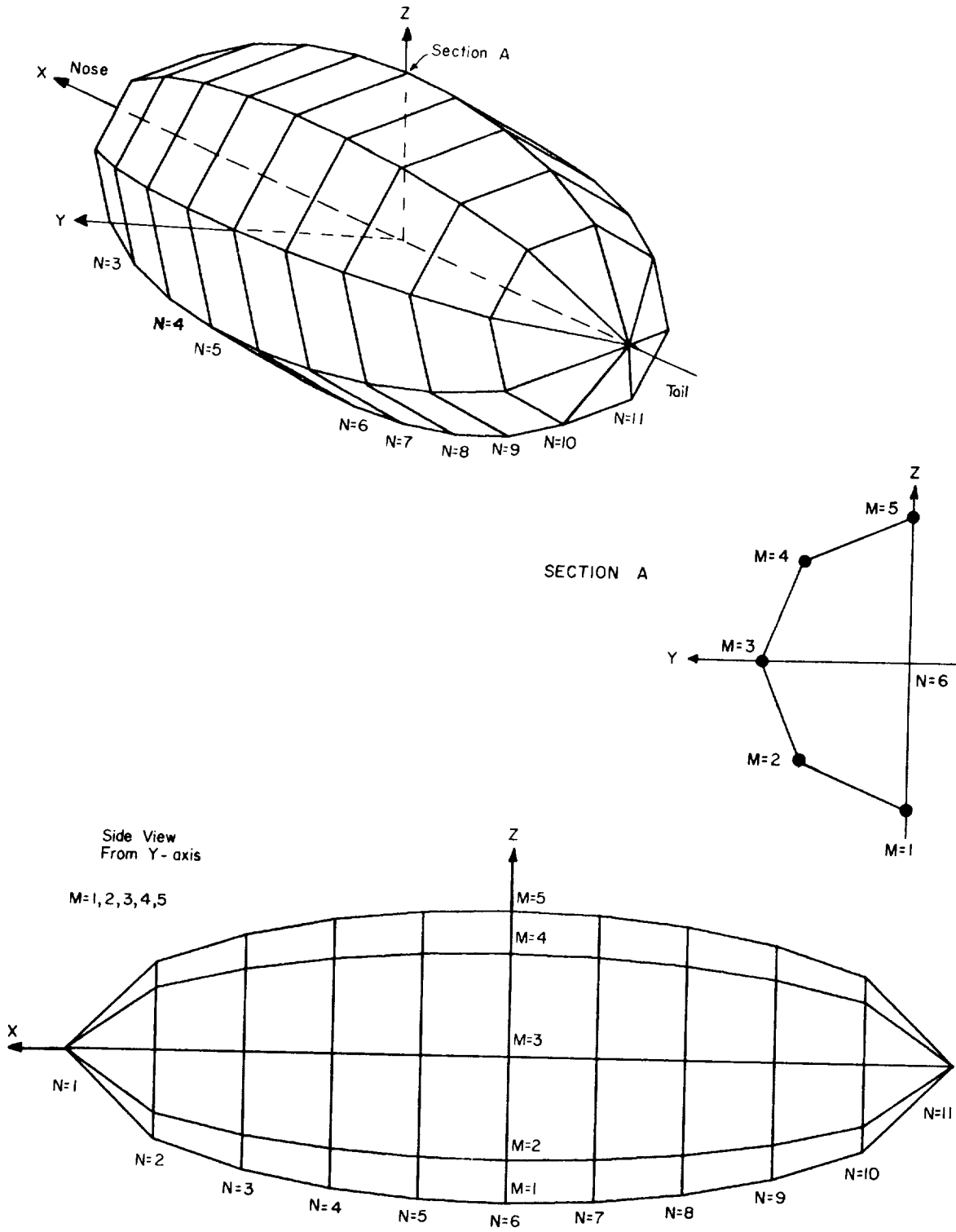
Figure F-2.   Schematic of indexing scheme used for a 3-1 ellipsoid with
60 panels describing the half-body.

379

Figure F-3. Format specification for the NCSU BODY program.

380

Figure F-4. NCSU BODY program sample data set for a 3-1 ellipsoid with 100 panels describing the half-body.

```
C***  THIS BASIC PROGRAM WAS OBTAINED FROM THE NAVAL SHIP RESEARCH AND
C***  DEVELOPMENT CENTER AND WAS THEN EXTENSIVELY MODIFIED FOR USE AT
C***  N.C. STATE UNIVERSITY.  WE WOULD LIKE TO THANK CHARLES W DAWSON FOR
C***  PROVIDING THE PROGRAM AND FOR HIS HELP IN MAKING SOME OF THE
C***  MODIFICATIONS.
C***
C***  REFERENCE:  THE XYZ POTENTIAL FLOW PROGRAM
C***              CHARLES W. DAWSON & JANET S. DEAN
C***              REPORT 3892
C***              NAVAL SHIP RESEARCH & DEVELOPMENT CENTER
C***              BETHESDA, MARYLAND  20034
C***
C***  POTENTIAL FLOW PROGRAM SECTION 1 OF 5 SECTIONS
      DIMENSION XK(9,3),G(9,6),F(9),CZ(10),ZP(9),XP(9),YP(9),ZP(9),X(6
     190),Y(650),Z(650),ZP(650),XR(4),RK(4)
      COMMON TITLE(20),KCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AQP(650),REFA,REBDY,NQUAD,IWRITE,NP,KNEW
      COMMON /BL/VDV(75),SS(75),VINF,VO,RDE,DELS(150),CFI(150),THT(150),
     1STOTAL,XKK
      COMMON /IKOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
      EQUIVALENCE (XNEW(1),DELS(1)),(XNEW(1),CF(1)),(ZNEW(1),THT(1)),(C
     1Z(1),F(1)),(F(1),FF(1,1))
      INTEGER P,P1,P2,P3,P4,PC,P5,P6,P7,P8,P9
      JREAD=1
      JWRITE=3
      KFILE1=7
      KFILE2=8
      KFILE3=9
      KFILE4=10
      KFILE5=11
    5 READ (JREAD,15, END=440) (TITLE(I),I=1,20)
      WRITE (JWRITE,10)
      KFILE=KFILE1
      REWIND KFILE1
      REWIND KFILE2
      REWIND KFILE3
      REWIND KFILE4
      REWIND KFILE5
   10 FORMAT (1H1,//,22X,34H POTENTIAL FLOW PROGRAM SECTION 1,//)
   15 FORMAT (20A4)
      KNEW=1
      KWRITE=0
      GHOLD=0
      GHOLD=1.0E-44
      J=0
      WRITE (JWRITE,20) (TITLE(I),I=1,20)
   20 FORMAT (1X,20A4)
      SA=-0
   25 FORMAT (I4)
   30 FORMAT (I1)THOMD. OF QUADS,  =,I4/17H NO. OF SECTIONS=,I4/31H MAX. N
C***  1D. OF ITERATIONS X FLOW ,I3)
C***  DEFINE AND READ PERTINENT GEOMETRIC & FLOW CONTROL PARAMETERS
      READ (JREAD,35) VINF,VO,RDE,REFA,IWRITE
   35 FORMAT (4F10.6,I5) RDE
      NSE=1
      NIX=150
      ISM=1
```

```
      EPS=0.0001
      ISP=0
      WRITE (JWRITE,30) NSE,NSE,NIX
      WRITE (JWRITE,40) VINF,VO,RDE,REFA,IWRITE
   40 FORMAT (//,1X,7HVVINF = ,F10.6,5H,5HVVO = ,F10.6,5H,6HRDE = ,F10.6,5
     1X,7HREFA = ,F10.6,5H,9H IWRITE = ,I2//)
   45 FORMAT (23H0CONVERGENCE CRITERIA,/F6.5)
   50 FORMAT (4H0  N,7X,2HX1,12X,2HX2,12X,2HX3,12X,2HX4,12X,2HXP,12X,2HX
     1N,12X,1HA,13X,3HCZ4/4H  N,7X,2HY1,12X,2HY2,12X,2HY3,12X,2HY4,12X,
     12HYP,12X,2HVN,12X,2HFL,12X,3HCZ5/4H  N,7X,2HZ1,12X,2HZ2,12X,2HZ3,
     112X,2HZ4,12X,2HZP,12X,2HEN,12X,4HCZ1 ,10X,3HCZ6/)
   55 FORMAT (4H1  N,7X,2HX1,12X,2HX2,12X,2HX3,12X,2HX4,12X,2HXP,12X,2HX
     1N,12X,1HA,13X,3HCZ4/4H  N,7X,2HY1,12X,2HY2,12X,2HY3,12X,2HY4,12X,
     12HYP,12X,2HVN,12X,2HFL,12X,3HCZ5/4H  N,7X,2HZ1,12X,2HZ2,12X,2HZ3,
     112X,2HZ4,12X,2HZP,12X,2HZN,12X,4HCZ1 ,10X,3HCZ6/)
   60 FORMAT (1H , //,1H0VM PLANES OF SYMMETRY)
      WRITE (JWRITE,60) ISM
      WRITE (JWRITE,45) EPS
      R=0
      XNIM=0.0
      XNLM=0.0
      NN=0
      P=1
      Q=1.0
      DO 65 I=1,31
      DO 65 J=1,31
   65 ID(I,J)=0
      J=0
C***  READ INPUT POINTS--AN ODD NUMBER MUST BE USED FOR BOTH NI AND NI
   70 READ (JREAD,70) X1,Y1,Z1,NI,NI,NS,NE,VN
   70 FORMAT (3F12.9,4I4,F12.9)
      MAX1=NI
      MIN1=NI
      MAX=NI
      MIN=NI
      NSS=NS
      PC=1
      GO TO 95
   75 READ (JREAD,70) X1,Y1,Z1,NI,NI,NS,NE,VN
      GO TO 90
   80 NI=NI-1
      IF (NI.LE.MAXGO) GO TO 85
      NI=1
      NIN=NI-1
      IF (NI.EQ.6)NS=0
   85 IF (NI.EQ.6) GO TO 105
      X1=XNEW(PC-1)
      Y1=YNEW(PC-1)
      Z1=ZNEW(PC-1)
   90 PC=PC-1
   95 IF (NS.NE.NSS) GO TO 105
      IF (NE.EQ.0) GO TO 100
      NI=NI
      NI=1N
C***  STORE INPUT POINTS IN POINT ARRAY
  100 XCPC1=X1
      YCPC1V1
      ZCPC2=1
      ID(NI,NI)=PC
```

```
      XMAX=AMAX0(XMAX,MI)                                              PFI 121
      XMIN=AMIN0(XMIN,MI)                                              PFI 122
      MMIN=AMIN0(MMIN,MI)                                              PFI 123
      IF (XMIN.GE.XI)XMIN=XI                                           PFI 124
      IF (XMAX.LE.XI)XMAX=XI                                           PFI 125
      GO TO (75,80),KMEM                                               PFI 126
  105 WRITE (JWRITE,110) NSS                                           PFI 127
      LINE=0                                                           PFI 128
  110 FORMAT (/,10H0 SECTION ,I4)                                      PFI 129
      IF (IWRITE.GT.0) GO TO 115                                       PFI 130
      WRITE (JWRITE,50)                                                PFI 131
      STOTAL=XMAX-XMIN                                                 PFI 132
  115 IF (RMEM.EQ.3) GO TO 120                                         PFI 133
      REROBY=VIMP*STOTAL/VO                                            PFI 134
  120 MMAXDD=MMAX                                                      PFI 135
      MMAXDD=MMAX                                                      PFI 136
      IDO=(MMAXDD-3)*(MMAXDD-1)/12                                     PFI 137
      N1=MMIN                                                          PFI 138
      MM2=MMAX-MMIN                                                    PFI 139
      MM2=MMAX-MMIN                                                    PFI 140
      IF (MOD(MM2,2).EQ.0.AND.MOD(MM2,2).EQ.0) GO TO 125              PFI 141
      ISP=1                                                            PFI 142
  125 MM2=MM2/2                                                        PFI 143
      MM2=MM2/2                                                        PFI 144
C***  DO LOOPS WHICH SWEEP POINT ARRAYS & SET UP PANEL GEOMETRY        PFI 145
      N1=MMIN                                                          PFI 146
      DO 295 NM=1,MM2                                                  PFI 147
      NQ=1                                                             PFI 148
      IT=ID(M1,N1)*ID(M1+1,N1)*ID(M1+2,N1)*ID(M1,N1+1)*ID(M1+1,N1+1)*ID( PFI 149
     1M1+1,N1+2)*ID(M1,N1+2)*ID(M1+1,N1+2)*ID(M1+2,N1+2)              PFI 150
      IF (IT.EQ.0) GO TO 295                                           PFI 151
      M2=M1+1                                                          PFI 152
      N2=N1+1                                                          PFI 153
      DO 290 M=M1,M2                                                   PFI 154
      N2=N1+1                                                          PFI 155
      DO 290 N=N1,N2                                                   PFI 156
      GO TO (130,135,140,145),NQ                                       PFI 157
  130 P1=ID(M,N)                                                       PFI 158
      P2=ID(M+1,N)                                                     PFI 159
      P3=ID(M+1,N+1)                                                   PFI 160
      P4=ID(M,N+1)                                                     PFI 161
      P5=ID(M+2,N)                                                     PFI 162
      P6=ID(M+2,N+1)                                                   PFI 163
      P7=ID(M+1,N+2)                                                   PFI 164
      P8=ID(M,N+2)                                                     PFI 165
      P9=ID(M+2,N+2)                                                   PFI 166
      IF ((IX(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).NE.Z(P2).AND.IX(P PFI 167
     1).NE.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NE.Z(P4))) GO TO 150       PFI 168
      P9=ID(M+2,N+2)                                                   PFI 169
      GO TO 150                                                        PFI 170
      P1=ID(M+1,N+1)                                                   PFI 171
      P2=ID(M+1,N)                                                     PFI 172
      P3=ID(M,N)                                                       PFI 173
      P4=ID(M+1,N+1)                                                   PFI 174
      P5=ID(M+1,N+1)                                                   PFI 175
      P6=ID(M+2,N+1)                                                   PFI 176
      P7=ID(M,N+1)                                                     PFI 177
      P8=ID(M+2,N)                                                     PFI 178
      P8=ID(M+2,N+1)                                                   PFI 179
      P9=P1                                                            PFI 180
      IF ((IX(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).OR.Z(P2).AND.IX(P PFI 181
     1).NE.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NE.Z(P4))) GO TO 150       PFI 182
      P9=ID(M+2,N-1)                                                   PFI 183
      GO TO 150                                                        PFI 184
  140 P1=ID(M+1,N)                                                     PFI 185
      P2=ID(M+1,N+1)                                                   PFI 186
      P3=ID(M,N+1)                                                     PFI 187
      P4=ID(M,N)                                                       PFI 188
      P5=ID(M+1,N+2)                                                   PFI 189
      P6=ID(M,N+2)                                                     PFI 190
      P7=ID(M-1,N+1)                                                   PFI 191
      P8=ID(M-1,N)                                                     PFI 192
      P9=P1                                                            PFI 193
      IF ((IX(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).OR.Z(P2).AND.IX(P PFI 194
     1).NE.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NE.Z(P4))) GO TO 150       PFI 195
      P9=ID(M-1,N+2)                                                   PFI 196
      GO TO 150                                                        PFI 197
  145 P1=ID(M,N+1)                                                     PFI 198
      P2=ID(M+1,N+1)                                                   PFI 199
      P3=ID(M+1,N)                                                     PFI 200
      P4=ID(M,N)                                                       PFI 201
      P5=ID(M-1,N+1)                                                   PFI 202
      P6=ID(M-1,N)                                                     PFI 203
      P7=ID(M+1,N-1)                                                   PFI 204
      P8=ID(M,N-1)                                                     PFI 205
      P9=P1                                                            PFI 206
      IF ((IX(P1).NE.X(P2).OR.Y(P1).NE.Y(P2).OR.Z(P1).OR.Z(P2).AND.IX(P PFI 207
     1).NE.X(P4).OR.Y(P1).NE.Y(P4).OR.Z(P1).NE.Z(P4))) GO TO 150       PFI 208
      P9=ID(M-1,N-1)                                                   PFI 209
  150 P1(1)=P1                                                         PFI 210
      IP(2)=P2                                                         PFI 211
      IP(3)=P3                                                         PFI 212
      IP(4)=P4                                                         PFI 213
      IP(5)=P5                                                         PFI 214
      IP(6)=P6                                                         PFI 215
      IP(7)=P7                                                         PFI 216
      IP(8)=P8                                                         PFI 217
      IP(9)=P9                                                         PFI 218
C***  COMPUTE NORMAL VECTOR TO PANEL                                   PFI 219
      X1=X(P3)-X(P1)                                                   PFI 220
      X2=X(P4)-X(P2)                                                   PFI 221
      Y1=Y(P3)-Y(P1)                                                   PFI 222
      Y2=Y(P4)-Y(P2)                                                   PFI 223
      Z1=Z(P3)-Z(P1)                                                   PFI 224
      Z2=Z(P4)-Z(P2)                                                   PFI 225
      XN=Y1*Z2-Y2*Z1                                                   PFI 226
      YN=Z1*X2-Z2*X1                                                   PFI 227
      ZN=X1*Y2-X2*Y1                                                   PFI 228
      R=SQ2(XN,YN,ZN)                                                  PFI 229
      XN=XN/R                                                          PFI 230
      YN=YN/R                                                          PFI 231
      ZN=ZN/R                                                          PFI 232
      AO=.5*R                                                          PFI 233
C***  COMPUTE PANEL CENTROID                                           PFI 234
      X1=X(P3)-X(P2)                                                   PFI 235
      Y1=Y(P3)-Y(P2)                                                   PFI 236
      Z1=Z(P3)-Z(P2)                                                   PFI 237
      X5=X1*Z2-Z1*X2                                                   PFI 238
      Y2=Z1*X2-X1*Z2                                                   PFI 239
      Z5=X1*Y2-X2*Y1                                                   PFI 240
```

383

```
        A1=SQ2(X5,Y5,Z5)
        A2=M-A1
        IT=1
        XC=(X(P2)+X(P4)+(A1*X(P3)+A2*X(P1))/R)/3.
        YC=(Y(P2)+Y(P4)+(A1*Y(P3)+A2*Y(P1))/R)/3.
        ZC=(Z(P2)+Z(P4)+(A1*Z(P3)+A2*Z(P1))/R)/3.
155     X4=YM*Z1-Y1*ZM
        Y4=ZM*X1-Z1*XM
        Z4=XM*Y1-X1*YM
        A=1./SQ2(X4,Y4,Z4)
        X4=X4*A
        Y4=Y4*A
        Z4=Z4*A
        X3=ZM*Y4-Z4*YM
        Y3=XM*Z4-X4*ZM
        Z3=YM*X4-Y4*XM
C***    COMPUTE POINTS IN QUAD SYSTEM
        DO 160 I=1,9
        L=IP(I)
        XP(I)=X3*(X(L)-XC)+Y3*(Y(L)-YC)+Z3*(Z(L)-ZC)
        YP(I)=X4*(X(L)-XC)+Y4*(Y(L)-YC)+Z4*(Z(L)-ZC)
160     ZP(I)=XN*(X(L)-XC)+YN*(Y(L)-YC)+ZN*(Z(L)-ZC)
C***    COMPUTE MATRIX COEFFICIENTS TO FIND LOCAL BODY SURFACE EQUATION
        DO 165 I=2,9
        G(I,1)=1.
        G(I,2)=XP(I)
        G(I,3)=YP(I)
        G(I,4)=XP(I)**2
        G(I,5)=YP(I)**2
165     G(I,6)=XP(I)*YP(I)
        DO 170 I=1,6
        G(5,I)=G(5,I)+G(6,I)
        G(6,I)=G(7,I)+G(8,I)
        F(I)=F(9)
        F(5)=F(7)+F(8)
170     F(6)=F(7)+F(8)
C***    SOLVE MATRIX EQ.  G*C=F  FOR C
        CALL MATINS(G,9,6,FF,6,1,DETERM,IDM,INDEX)
        IF (IDM.EQ.1) GO TO (180,185),IT
        WRITE (JWRITE,175)
175     FORMAT (33H ERROR IN INPUT - SINGULAR MATRIX)
        GO TO 185
180     IT=2
C***    COMPUTE NEW NORMAL VECTORS
        XN=XM-CZ(2)*X3-CZ(3)*X4
        YN=YM-CZ(2)*Y3-CZ(3)*Y4
        ZN=ZM-CZ(2)*Z3-CZ(3)*Z4
        A=1./SQ2(XN,YN,ZN)
        XN=XN*A
        YN=YN*A
        ZN=ZN*A
C***    STORE DATA IN ARRAY TO BE WRITTEN ON TAPE
185     B(J+1)=XP(1)
        B(J+2)=YP(1)
        B(J+3)=XP(2)
        B(J+4)=YP(2)
        B(J+5)=XP(3)
        B(J+6)=XP(4)
        B(J+7)=YP(4)
        B(J+8)=X3
        B(J+9)=Y3
        B(J+10)=Z3
        B(J+11)=X4
        B(J+12)=Y4
        B(J+13)=Z4
        XCP(IK+1)=XC
        YCP(IK+1)=YC
        ZCP(IK+1)=ZC
        XNP(IK+1)=XN
        YNP(IK+1)=YN
        ZNP(IK+1)=ZN
        AQP(IK+1)=AQ
C***    COMPUTE QUADRUPOLE MOMENTS
        XI1=XP(1)-XP(2)
        XI2=XP(1)-XP(4)
        XI3=XP(3)-XP(4)
        XI4=XP(3)-XP(2)
        XI5=XP(2)+XP(4)
        YI1=YP(1)-YP(4)
        YI2=YP(1)-YP(2)
        YI3=YP(3)-YP(2)
        YI4=YP(3)-YP(4)
        YI5=YP(2)+YP(4)
        R1=A1/24.
        R2=A2/24.
        R3=AQ/12.
        AXX=(XI1**2+XI2**2)*R1+(XI3**2+XI4**2)*R2+XI5**2*R3
        AXY=(XI1*YI1+XI2*YI2)*R1+(XI3*YI3+XI4*YI4)*R2+XI5*YI5*R3
        AYY=(YI1**2+YI2**2)*R1+(YI3**2+YI4**2)*R2+YI5**2*R3
C***    COMPUTE BODY SOLID ANGLE
        X1=XC*X3-YC*Y3+ZC*Z3
        Y1=XC*X4+YC*Y4+ZC*Z4
        Z1=XC*XN+YC*YN+ZC*ZN
        RO=1./SQ2(X1,Y1,Z1)
        RCU=RO**3
        RSV=RCU*RO**2*RO
        SA=SA*ZI*(AQ*RCU-((AXX*(YI**2+ZI**2)-2.*AXY*XI*YI+
     1  (XI**2)*(1.5-15.*XI*YI*AXY)*RSV)
        B(J+14)=AXX
        B(J+15)=AXY
        B(J+16)=AYY
C***    BODY ERROR TESTS
        D1=SQ2((XP(3)-XP(1)),(YP(3)-YP(1)),0.)
        D2=SQ2((XP(4)-XP(2)),(YP(4)-YP(2)),0.)
        FL=-SQ4MAX(D1,D2)
        CZ23=ABS(CZ(2))+ABS(CZ(3))
        IF (ABS(CZ(1)).LT.FL*.001) GO TO 190
        IF (ABS(CZ(1)).LT.FL*.3) GO TO 200
190     IF (IWRITE.GT.0) GO TO 200
195     IF (IWRITE.GT.0) GO TO 200
        WRITE (JWRITE,195) CZ23
        FORMAT (29H QUESTIONABLE POINT -POOR FIT,6E14.3)
        LINE=LINE+1
200     IF (XP(4).LT.XP(1)) GO TO 205
        IF ((YP(4)-YP(1))*(YP(1)-YP(2)).GE.0.) GO TO 220
205     IF (IWRITE.GT.0) GO TO 215
        WRITE (JWRITE,210) (XP(I),YP(I),I=1,4)
```

```
        MI=1
        NI=1
        NS=1000
        K=(MMAXQD-3)*(MMAXQD-1)                                         PFI 481
        ME=0                                                            PFI 482
        NOE=NOE*2*(MMAXQD-1)                                            PFI 483
        REWIND KFILE1                                                   PFI 484
        DO 390 I=1,100                                                  PFI 485
        READ (KFILE1)Q,(B(IJK),IJK=1,192)                              PFI 486
        WRITE (KFILE2)Q,(B(IJK),IJK=1,192)                             PFI 487
  390   CONTINUE                                                        PFI 488
        KFILE=KFILE2                                                    PFI 489
        J=JHOLD                                                         PFI 490
        IF (J.EQ.0) GO TO 400                                           PFI 491
        DO 395 I=1,J                                                    PFI 492
  395   B(I)=BHOLD(I)                                                   PFI 493
  400   Q=QHOLD                                                         PFI 494
        JP=JHOLD/16                                                     PFI 495
        QADD=QHOLD+.5                                                   PFI 496
        P=QADD-JP                                                       PFI 497
        ISTART=(MMAXQD-3)*MMAXQD                                        PFI 498
        XHOLD1=X(ISTART+1)                                              PFI 499
        XHOLD2=X(IST+1)                                                 PFI 500
C***    Y REFERENCE LINE MUST BE 0.0                                    PFI 501
        ZAV=0.0                                                         PFI 502
        DO 405 I=1,MMAXQD                                               PFI 503
        XNEM(I)=X(ISTART+I)                                             PFI 504
        YNEM(I)=Y(ISTART+I)                                            PFI 505
        ZNEM(I)=Z(ISTART+I)                                            PFI 506
  405   ZAV=ZAV+ZNEM(I)                                                PFI 507
        ZAV=ZAV/MMAXQD                                                  PFI 508
        RAV=0.0                                                         PFI 509
        DO 410 I=1,MMAXQD                                               PFI 510
        RNEW=SQRT((YNEM(I))**2+(ZNEM(I)-ZAV)**2)                       PFI 511
  410   RAV=RAV+RNEW                                                   PFI 512
        RAV=RAV/MMAXQD                                                  PFI 513
        FAC=(AVXCG-XHOLD1)/(XHOLD2-XHOLD1)                             PFI 514
        AVSLOP=0.0                                                      PFI 515
        DO 415 I=1,MMAXQD                                               PFI 516
        YCG=YNEM(I)-YNEM(I)-Y(IST+I))*FAC                             PFI 517
        ZCG=ZNEM(I)-(ZNEM(I)-Z(IST+I))*FAC                           PFI 518
        R2=SQRT((YCG)**2+(ZCG-ZAV)**2)                                PFI 519
        R1=SQRT((YCG)**2+(ZCG-ZAV)**2)+2.0*AVDELS                     PFI 520
        SLOPE=(R2-R1)/(XHOLD2-XHOLD1)                                 PFI 521
  415   AVSLOP=AVSLOP+SLOPE                                           PFI 522
        AVSLOP=AVSLOP/MMAXQD                                          PFI 523
        AREAT=-.0*AREAAV*2.0*MMAXQD                                   PFI 524
        XINF=XHOLD1-AREAT/(3.14159**RAV)                             PFI 525
C***    CHECK TO MAKE SURE BODY SLOPE IS DECREASING                   PFI 526
        IF (AVSLOP.GE.0.)AVSLOP=-ABS(RAV/(XINF-XHOLD1))              PFI 527
        WRITE (JWRITE,420) AVSLOP,AREAAV,AVXCG,XINF                   PFI 528
  420   FORMAT (1H1/50X,24HBEGIN MAKE BODY GEOMETRY///,17H AVERAGE SLOPE = PFI 529
       1 ,F8.5,3X,22H AVERAGE PANEL AREA = ,F8.5,3X,22H AVERAGE X-CENTROID PFI 530
       2 = ,F10.5,3X,22H END OF BODY AT X = ,F10.5)                   PFI 531
        DELTAZ=.01                                                    PFI 532
        KK=1                                                          PFI 533
        XX1=XHOLD1                                                    PFI 534
        ZZ2=0.0                                                       PFI 535
```

```
  425   RR1=RAV                                                       PFI 541
        ZZ2=ZZ2*DELTAZ                                                PFI 542
        XX2=ZZ2*(XINF-XHOLD1)+XHOLD1                                 PFI 543
        RR2=RAV*EXP(AVSLOP*ZZ2)*(1.0-ZZ2)                            PFI 544
        DTHETA=3.14159/(MMAXQD-1)                                     PFI 545
        AREA2=(RR2*RR1)*SIN(DTHETA/2.0)*SQRT((RR2-RR1)**2+(XX2-XX1)**2) PFI 546
        IF (AREA2.LT.AREAAV) GO TO 425                                PFI 547
        XK(KK)=XX2                                                    PFI 548
        XK(KK)=RR2                                                    PFI 549
        XX1=XX2                                                       PFI 550
        RR1=RR2                                                       PFI 551
        KK=KK+1                                                       PFI 552
        IF (KK.GE.4) GO TO 430                                        PFI 553
        GO TO 425                                                     PFI 554
  430   XK(4)=XINF                                                    PFI 555
        RK(4)=0.0                                                     PFI 556
        X1=XK(1)                                                      PFI 557
        X2=XK(2)                                                      PFI 558
        X3=XK(3)                                                      PFI 559
        X4=XK(4)                                                      PFI 560
        R1=RK(1)                                                      PFI 561
        R2=RK(2)                                                      PFI 562
        R3=RK(3)                                                      PFI 563
        R4=RK(4)                                                      PFI 564
        NSTRT1=MMAXQD                                                 PFI 565
        NSTRT2=2*MMAXQD                                               PFI 566
        NSTRT3=3*MMAXQD                                               PFI 567
        NSTRT4=4*MMAXQD                                               PFI 568
        VFAC=(-Y(1))/(XINF-XHOLD1)                                    PFI 569
        ZFAC=(ZAV-Z(1))/(XINF-XHOLD1)                                PFI 570
        DO 435 I=1,MMAXQD                                             PFI 571
        COSTHT=COS(THETA)                                            PFI 572
        SINTHT=SIN(THETA)                                            PFI 573
        XNEW(NSTRT1+I)=X1                                            PFI 574
        XNEW(NSTRT2+I)=X2                                            PFI 575
        XNEW(NSTRT3+I)=X3                                            PFI 576
        XNEW(NSTRT4+I)=X4                                            PFI 577
        YNEW(NSTRT1+I)=R1*COSTHT-(XNEW(NSTRT1+I)-XHOLD1)*VFAC        PFI 578
        YNEW(NSTRT2+I)=R2*COSTHT-(XNEW(NSTRT2+I)-XHOLD1)*VFAC        PFI 579
        YNEW(NSTRT3+I)=R3*COSTHT-(XNEW(NSTRT3+I)-XHOLD1)*VFAC        PFI 580
        YNEW(NSTRT4+I)=R4*COSTHT-(XNEW(NSTRT4+I)-XHOLD1)*VFAC        PFI 581
        ZNEW(NSTRT1+I)=R1*SINTHT+ZAV-(XNEW(NSTRT1+I)-XHOLD1)*ZFAC    PFI 582
        ZNEW(NSTRT2+I)=R2*SINTHT+ZAV-(XNEW(NSTRT2+I)-XHOLD1)*ZFAC    PFI 583
        ZNEW(NSTRT3+I)=R3*SINTHT+ZAV-(XNEW(NSTRT3+I)-XHOLD1)*ZFAC    PFI 584
        ZNEW(NSTRT4+I)=R4*SINTHT+ZAV-(XNEW(NSTRT4+I)-XHOLD1)*ZFAC    PFI 585
  435   THETA=THETA+DTHETA                                           PFI 586
        GO TO 310                                                    PFI 588
  440   CALL EXIT                                                    PFI 589
        END                                                          PFI 590
                                                                     PFI 591
                                                                     PFI 592

        FUNCTION SQ2(X,Y,Z)                                          SQ2  1
C***    FUNCTION TO COMPUTE SQUARE ROOT OF R**2                      SQ2  2
        R=ABS(X)+ABS(Y)+ABS(Z)+.0000000000001                       SQ2  3
        RS=X**2+Y**2+Z**2                                           SQ2  4
        R=R+RS/R                                                     SQ2  5
```

```
      R=.25*R*RS/R
      R=R*RS/R
      SQ2=.25*R*RS/R
      RETURN
      END

      SUBROUTINE MATINS(A,NR,N1,B,NC,M1,DETERM,ID,INDEX)
C     MATRIX INVERSION(PIVOT METHOD) WITH SIMULATNEOUS EQUATION SOLUTION
C     INPUT A(NR,NR),B(NR,NC),& INDEX(NR,3)--NR,NC ARE ARRAY DIMENSIONS.
C     DEFINITIONS: N1-THE ORDER OF A, M1-THE NUMBER OF COLUMN VECTORS IN
C     B(MAY BE 0), DETERM-CONTAINS DETERMINANT ON EXIT, INDEX-WORK ARRAY
C     ID-(ID=2--SINGULAR MATRIX,ID=1--SUCESSFUL INVERSION), A-THE INPUT
C     MATRIX WHICH IS REPLACED BY A INVERSE, B-COLUMN VECTORS WHICH IS
C     REPLACED BY CORRESPONDING SOLUTION VECTORS. IF MATRIX IS TO BE
C     SCALED DELETE CARD 29 & PRESET DETERM BEFORE ENTERING THE ROUTINE.
      DIMENSION A(9,6),B(9,1),INDEX(9,3)
      EQUIVALENCE (IROW,JROW),(ICOLUM,JCOLUM),(AMAX,T,SWAP)
C     INITIALIZATION
      N=N1
      M=M1
      DETERM=1.0
      DO 5 J=1,N
    5 INDEX(J,3)=0
      DO 105 I=1,N
C     SEARCH FOR PIVOT ELEMENT
      AMAX=0.0
      DO 30 J=1,N
      IF (INDEX(J,3)-1) 10,30,10
   10 DO 25 K=1,N
      IF (INDEX(K,3)-1) 15,25,140
   15 IF (AMAX-ABS(A(J,K))) 20,25,25
   20 IROW=J
      ICOLUM=K
      AMAX=ABS(A(J,K))
   25 CONTINUE
   30 CONTINUE
      INDEX(ICOLUM,3)=INDEX(ICOLUM,3)+1
      INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
      IF (IROW-ICOLUM) 35,55,35
   35 DETERM=-DETERM
      DO 40 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
   40 A(ICOLUM,L)=SWAP
      IF (M) 55,55,45
   45 DO 50 L=1,M
      SWAP=B(IROW,L)
      B(IROW,L)=B(ICOLUM,L)
   50 B(ICOLUM,L)=SWAP
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
   55 PIVOT=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT
      A(ICOLUM,ICOLUM)=1.0
      DO 65 L=1,N
   65 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
      IF (M) 80,80,70

   70 DO 75 L=1,M
   75 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C     REDUCE NON-PIVOT ROWS
   80 DO 105 L1=1,N
      IF (L1-ICOLUM) 85,105,85
   85 T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 90 L=1,N
   90 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      IF (M) 105,105,95
   95 DO 100 L=1,M
  100 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  105 CONTINUE
C     INTERCHANGE COLUMNS
      DO 120 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 110,120,110
  110 JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 115 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
  115 CONTINUE
  120 CONTINUE
  125 CONTINUE
  130 CONTINUE
      ID=1
  135 RETURN
  140 ID=2
      GO TO 135
      END

      SUBROUTINE PFP2(TSW,KM)
C     POTENTIAL FLOW PROGRAM SECTION 2
      DIMENSION A(193),VI(1000),CII9001,VX(8),VY(8),VZ(8)
      EQUIVALENCE (Y3,Y?)
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP1
     1650),AGP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW
      COMMON /INOUT/READ,IWRITE,KFT,S,KFILE1,KFILE2,KFILE...,KFILE
    5 FORMAT (///,23HOPTENTIAL FLOW PROGRAM SECTION 2)
      WRITE (JWRITE,5)
      BLK=1.0
      IDW=0
   10 FORMAT (1H0,20A4)
      WRITE (JWRITE,10) (TITLE(I),I=1,20)
C     READ FIRST BLOCK OF B ARRAY
      READ (KFILE1)(B(I),I=1,193)
      K=1
      J=1
      P=1
      JC=1
      JV=2
      KMM=59(NP*4)/5)+1
C     START LOOP OVER QUADRILATERALS
```

388

```
      IF (B(11)-P) 345,20,345
   20 J=2
   25 X1=B(J)
      Y1=B(J+1)
      X2=B(J+2)
      Y2=B(J+3)
      X3=B(J+4)
      Y3=B(J+5)
      X4=B(J+6)
      Y4=B(J+7)
      XM=XMP(K)
      YM=YMP(K)
      ZM=ZMP(K)
      XC=XCP(K)
      YC=YCP(K)
      ZC=ZCP(K)
      HI=HQP(K)
      X1=B(J+8)
      YK=B(J+9)
      ZK=B(J+10)
      XY=B(J+11)
      ZY=B(J+12)
C***  COMPUTE LENGTH OF SIDES OF QUADRILATERAL
      IF (X2-X3) 35,30,35
   30 GO TO 40
   35 CH23=(Y2-Y3)/(X2-X3)
      C123=0.
      C143=0.
   40 IF (X3-X4) 50,45,50
   45 C143=1.
      GO TO 55
   50 CH34=(Y4-Y3)/(X4-X3)
      C134=0.
   55 IF (X4-X1) 65,60,65
   60 C161=1.
      GO TO 70
   65 CH41=(Y1-Y4)/(X1-X4)
   70 IF (X1-X2) 80,75,80
   75 C112=1.
      GO TO 85
   80 CH12=(Y2-Y1)/(X2-X1)
   85 C112=0.
C***  COMPUTE QUADRUPOLE MOMENTS
      CIX=B(J+13)
      CIY=B(J+14)
      CIV=B(J+15)
      CVI2=0.0
      CX12=0.0
      CY23=0.0
      CX23=0.0
      CY34=0.0
      CX34=0.0
      CY41=0.0
```

```
C***  COMPUTE SINE AND COSINE OF SLOPE ANGLE FOR EACH SIDE
      C141=0.0
      IF (D12) 90,95,90
   90 CY12=(Y2-Y1)/D12
      CX12=(X1-X2)/D12
   95 IF (D23) 100,105,100
  100 CY23=(Y2-Y3)/D23
      CX23=(X2-X3)/D23
  105 IF (D34) 110,115,110
  110 CY34=(Y4-Y3)/D34
      CX34=(X3-X4)/D34
  115 IF (D41) 120,125,120
  120 CY41=(Y1-Y4)/D41
      CX41=(X4-X1)/D41
C***  COMPUTE MAXIMUM LENGTH OF QUADRILATERAL
  125 ST=AMAX(X1-X3)
      ST2=SQ2F(X2-X4,Y2-Y4,0.,0.)
      ST=AMAX(ST,ST2,D12,D23,D34,D41)
C***  START LOOP OVER CENTROID POINTS
      KQ=1
      L=1
  130 IS=1
      XCQ=XCP(I)
      YCQ=YCP(I)
      ZCQ=ZCP(I)
C***  COMPUTE DISTANCE BETWEEN QUADRILATERAL AND CENTROID
  135 RPQ=SQ2F(XC,XCQ,YC,YCQ,ZC,ZCQ)
      IF (RPQ-ST*+) 140,140,200
  140 X=(XCQ-XC)*XK+(YCQ-YC)*YK+(ZCQ-ZC)*ZX
      Y=(XCQ-XC)*XV+(YCQ-YC)*YV+(ZCQ-ZC)*ZY
      Z=(XCQ-XC)*XM+(YCQ-YC)*YM+(ZCQ-ZC)*ZM
      IF (RPQ-ST*2.0) 145,145,190
C***  COMPUTE VELOCITY COEFFICIENTS BY EXACT METHOD
  145 R1=SQ2F(X1-X,Y1-Y,Z-0.)
      R2=SQ2F(X2-X,Y2-Y,Z-0.)
      R3=SQ2F(X3-X,Y3-Y,Z-0.)
      R4=SQ2F(X4-X,Y4-Y,Z-0.)
      IF ((R1+R2)-LE.D12) GO TO 375
      IF ((R2+R3)-LE.D23) GO TO 375
      IF ((R3+R4)-LE.D34) GO TO 375
      IF ((R4+R1)-LE.D41) GO TO 375
      CLA1=ALOG((R1+R2-D12)/(R1+R2+D12))
      CLA2=ALOG((R2+R3-D23)/(R2+R3+D23))
      CLA3=ALOG((R3+R4-D34)/(R3+R4+D34))
      CLA4=ALOG((R4+R1-D41)/(R4+R1+D41))
      TV1=CX12*CLA1+CX23*CLA2+CX34*CLA3+CX41*CLA4
      TV2=CY12*CLA1+CY23*CLA2+CY34*CLA3+CY41*CLA4
      TV3=0.
  150 ZSQ=Z**2
      IF (ABS(Z/ST)-.010) 195,150,150
      E1=ZSQ+(X-X1)**2
      E2=ZSQ+(X-X2)**2
      E3=ZSQ+(X-X3)**2
      E4=ZSQ+(X-X4)**2
      H1=(Y-Y1)*(X-X1)
      H2=(Y-Y2)*(X-X2)
```

```
      H3=(Y-Y3)*(X-X3)                                            PF2 144
      H4=(Y-Y4)*(X-X4)                                            PF2 145
      IF (CI12) 155,155,160                                       PF2 146
  155 MS1=(CM12*E1-H1)/(Z*R1)                                     PF2 147
      MS2=(CM12*E2-H2)/(Z*R2)                                     PF2 148
      TVZ=ATAN1(MS1)-ATAN1(MS2)                                   PF2 149
  160 IF (CI23) 165,165,170                                       PF2 150
  165 TVZ=TVZ+ATAN1(CM23*E2-H2)/(Z*R2))-ATAN1(CM23*E3-H3)/(Z*R3)) PF2 151
  170 IF (CI34) 175,175,180                                       PF2 152
  175 TVZ=TVZ+ATAN1(CM34*E3-H3)/(Z*R3))-ATAN1(CM34*E4-H4)/(Z*R4)) PF2 153
  180 IF (CI41) 185,185,195                                       PF2 154
  185 TVZ=TVZ+ATAN1(CM41*E4-H4)/(Z*R4))-ATAN1(CM41*E1-H1)/(Z*R1)) PF2 155
      GO TO 195                                                   PF2 156
C*** COMPUTE VELOCITY COEFFICIENTS BY QUADRAPOLE METHOD            PF2 157
  190 RPQ3=RPQ*P3                                                 PF2 158
      RPQ7=(RPQ3**2)*RPQ                                          PF2 159
      MS1=H/RP03                                                  PF2 160
      XSQ=X**2                                                    PF2 161
      YSQ=Y**2                                                    PF2 162
      ZSQ=Z**2                                                    PF2 163
      PS=YSQ+2*SQ-4.*XSQ                                          PF2 164
      QS=XSQ+ZSQ-4.*YSQ                                           PF2 165
      MS2=X*(9.*PS+30.*XSQ1/RPQ7                                  PF2 166
      MS3=3.*Y*PS/RPQ7                                            PF2 167
      MS4=3.*X*QS/RPQ7                                            PF2 168
      TVX=A*MS1-CIXY*MS3-CIXX*MS2-CIYY*MS4                        PF2 169
      MS1=V/RP03                                                  PF2 170
      MS2=Y*(9.*QS+30.*YSQ)/RPQ7                                  PF2 171
      MS4=MS1-CIXY*MS3-CIXY*MS4-CIYY*MS2                          PF2 172
      TVY=A*MS1-(A/RP03-3.+CIXX*PS-5.*CIXY*X*Y+CIYY*QS)/RPQ7)     PF2 173
  195 VX(IS)=TVX*X+TVY*XY+TVZ*XN                                  PF2 174
      VY(IS)=TVX*XY+TVY*Y+TVZ*YN                                  PF2 175
      VZ(IS)=TVX*ZX+TVY*ZY+TVZ*ZN                                 PF2 176
      GO TO 205                                                   PF2 177
C*** COMPUTE VELOCITY COEFFICIENTS BY MONOPOLE METHOD              PF2 178
  200 ARPQ3=A/(RPQ**3)                                            PF2 179
      VX(IS)=(XCQ-XC)*ARPQ3                                       PF2 180
      VY(IS)=(YCQ-YC)*ARPQ3                                       PF2 181
      VZ(IS)=(ZCQ-ZC)*ARPQ3                                       PF2 182
C*** REFLECT CENTROID POINT IN PLANE OF SYMMETRY                   PF2 183
  205 GO TO (210,220,230,235,245,250,255,260),IS                  PF2 184
C*** DO LOOPS SET UP TO FORCE USE OF INDEX REGISTERS               PF2 185
  210 JI=JV                                                       PF2 186
      J2=JC                                                       PF2 187
      VI(JI)=VX(I)                                                PF2 188
      VI(JI+1)=VY(I)                                              PF2 189
      VI(JI+2)=VZ(I)                                              PF2 190
      IF (ISN) 275,275,215                                        PF2 191
  215 IS=2                                                        PF2 192
C*** XZ SYMMETRY                                                   PF2 193
      YCQ=-YCQ                                                    PF2 194
      GO TO 135                                                   PF2 195
  220 IF (ISM-1) 270,270,225                                      PF2 196
C*** XY SYMMETRY                                                   PF2 197
  225 IS=3                                                        PF2 198
      ZCQ=-ZCQ                                                    PF2 199
      GO TO 135                                                   PF2 200
  230 IS=4                                                        PF2 201
      YCQ=-YCQ                                                    PF2 202
      GO TO 135                                                   PF2 203
  235 IF (ISM-2) 265,265,240                                      PF2 204
C*** YZ SYMMETRY                                                   PF2 205
  240 IS=5                                                        PF2 206
      XCQ=-XCQ                                                    PF2 207
      GO TO 135                                                   PF2 208
  245 IS=6                                                        PF2 209
      YCQ=-YCQ                                                    PF2 210
      GO TO 135                                                   PF2 211
  250 IS=7                                                        PF2 212
      ZCQ=-ZCQ                                                    PF2 213
      GO TO 135                                                   PF2 214
  255 IS=8                                                        PF2 215
      YCQ=-YCQ                                                    PF2 216
      GO TO 135                                                   PF2 217
C*** ADD CONTRIBUTIONS OF ALL REFLECTIONS                          PF2 218
  260 VI(JI)=VI(JI)+VX(1)+VX(2)+VX(3)+VX(4)+VX(5)+VX(6)+VX(7)+VX(8) PF2 219
      VI(JI+1)=VI(JI+1)+VY(1)+VY(2)+VY(3)+VY(4)+VY(5)+VY(6)+VY(7)+VY(8) PF2 220
      VI(JI+2)=VI(JI+2)+VZ(1)+VZ(2)+VZ(3)+VZ(4)+VZ(5)+VZ(6)+VZ(7)+VZ(8) PF2 221
  265 VI(JI)=VI(JI)+VX(1)+VX(2)+VX(3)                             PF2 222
      VI(JI+1)=VI(JI+1)+VY(1)+VY(2)+VY(3)                         PF2 223
      VI(JI+2)=VI(JI+2)+VZ(1)+VZ(2)+VZ(3)                         PF2 224
  270 VI(JI)=VI(JI)+VX(1)+VX(2)                                   PF2 225
      VI(JI+1)=VI(JI+1)+VY(1)+VY(2)                               PF2 226
      VI(JI+2)=VI(JI+2)+VZ(1)+VZ(2)                               PF2 227
  275 C1(JJ2)=XNQ*VI(JI)+YNQ*VI(JI+1)+ZNQ*VI(JI+2)                PF2 228
      JV=JV+3                                                     PF2 229
  280 JC=JC+1                                                     PF2 230
C*** WRITE COEFFICIENTS ON TAPE                                    PF2 231
      IF (JV-ICOI1) 310,285,285                                   PF2 232
  285 JV=2                                                        PF2 233
      VI(1)=BLK                                                   PF2 234
      IF (BLK-636.0) 290,295,300                                  PF2 235
  290 WRITE (KFILE3)BLK,V1                                        PF2 236
      GO TO 305                                                   PF2 237
  295 REWIND KFILE3                                               PF2 238
  300 WRITE (KFILE3)BLK,V1                                        PF2 239
  305 BLK=BLK+1.                                                  PF2 240
  310 IF (JC-901) 320,315,315                                     PF2 241
  315 IDW=IDW+1                                                   PF2 242
      WRITE (KFILE4)IDW,C1                                        PF2 243
      JC=1                                                        PF2 244
  320 KQ=KQ+1                                                     PF2 245
      L=L+1                                                       PF2 246
C*** END OF LOOP OVER CENTROIDS                                    PF2 247
      IF (KQ-KM) 130,130,325                                      PF2 248
  325 CI(JC)=0                                                    PF2 249
      IF (KQ-KMM) 280,330,330                                     PF2 250
  330 P=P+1                                                       PF2 251
      K=K+1                                                       PF2 252
      IF (K-KN) 335,335,350                                       PF2 253
      J=J+16                                                      PF2 254
C*** END OF LOOP OVER QUADRILATERALS                               PF2 255
C*** READ NEXT BLOCK OF B ARRAY IF NEEDED                          PF2 256
  335 IF (J-193) 25,340,340                                       PF2 257
  340 READ (KFILE4)(B(I),I=1,193)                                 PF2 258
      J=2                                                         PF2 259
  345 IF (B(1)-P) 345,25,345                                      PF2 260
      WRITE (JWRITE,15) B(1),P                                    PF2 261
      CALL EXIT                                                   PF2 262
  350 IF (BLK-636.0) 355,360,365                                  PF2 263
```

389

```
C***  WRITE REMAINING COEFFICIENTS ON TAPE
  355 WRITE (KFILE3)BLK,V1
      REWIND KFILE3
      GO TO 370
  360 REWIND KFILE5
  365 WRITE (KFILE5)BLK,V1
      REWIND KFILE5
  370 WRITE (KFILE4)IDW,C1
      REWIND KFILE
      GO TO 385
  375 WRITE (JWRITE,380) L,P
  380 FORMAT (3H L=,I5,20X,3H P=,F5.1)
  385 CONTINUE
      RETURN
      END
```

```
      FUNCTION SQ2F(X1,X2,Y1,Y2,Z1,Z2)
C***  FUNCTION TO COMPUTE DISTANCE BETWEEN 2 POINTS
      X=X1-X2
      Y=Y1-Y2
      Z=Z1-Z2
      RS=Z*Z+Y*Y+Z*Z
      R=ABS(X)+ABS(Y)+ABS(Z)+1.0E-20
      R=R+RS/R
      R=R+RS/R
      R=.25*R+RS/R
      R=R+RS/R
      SQ2F=.25*R+RS/R
      RETURN
      END
```

```
      SUBROUTINE PFP5(EPS,MIX)
C***  POTENTIAL FLOW PROGRAM SECTION 3--SOLVES FOR SOURCE DENSITY
      DIMENSION VIP(650),SI5(650),COEF(900)
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AOP(650),REPA,REBCOV,MQUAD,JWRITE,NP,KNEW
      COMMON /IHOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
     15
      COMMON /SI5/SN(650)
      WRITE (JWRITE,5)
    5 FORMAT (///,3H0POTENTIAL FLOW PROGRAM SECTION 3)
   10 FORMAT (1H0,20A4)
C***  SET CONDITIONS FOR FLOW OF -1.0 IN X DIRECTION, ZERO IN Y&Z DIRECT
      K1=1
      K2=NP
C***  COMPUTE INITIAL SOURCE APPROXIMATION & SET PARTIAL SUM VECTOR = 0
      FX=-1
      DO 15 K=1,NP
      VIP(K)=XNP(K)*FX
      SI5(K)=-VIP(K)*.11936
   15 SN(K)=0.
      SN(K+1)=0.
      SN(K+2)=0.
      SN(K+3)=0.
      SN(K+4)=0.
```

```
      WRITE (JWRITE,150) FX
      WRITE (JWRITE,20)
   20 FORMAT (/,12X,37HITERATIVE MATRIX SOLUTION INFORMATION//,27HOITERA
     1TION    SUM OF CHANGES,9X,1HA,10X,2HB1,10X,2HB2)
      IT=1
      IC=5
C***  START ITERATION,READ FIRST BLOCK OF COEFS.,START LOOP OVER QUADS.
   25 READ (KFILE4)IDW,COEF
      J=0
      DO 40 K=1,NP
C***  PICK UP SOURCE DENSITY & START LOOP OVER CENTROID POINTS
      SP=SI(C,K)
      DO 40 KP=1,NP,5
      IF (J-900) 35,30,30
   30 READ (KFILE4)IDW,COEF
      J=0
C***  COMPUTE PARTIAL SUMS FOR NEXT 5 POINTS
   35 SN(KP)=SN(KP)+COEF(J+1)*SP
      SN(KP+1)=SN(KP+1)+COEF(J+2)*SP
      SN(KP+2)=SN(KP+2)+COEF(J+3)*SP
      SN(KP+3)=SN(KP+3)+COEF(J+4)*SP
      SN(KP+4)=SN(KP+4)+COEF(J+5)*SP
      J=J+5
C***  END LOOP OVER CENTROID POINTS & END LOOP OVER QUADS.
   40 CONTINUE
C***  COMPUTE NEW SOURCE
      REWIND KFILE4
      PASS=1.0
      SUM=0.
      DO 45 K=1,K2
      SN(K)=(SN(K)+VIP(K))*(-.5/3.141592265)
      TEST=ABS(SN(K)-SI(C,K))
      SUM=SUM+TEST
      IF (TEST.GT.EPS)PASS=-1.0
   45 CONTINUE
      WRITE (JWRITE,145) IT,SUM
      IF (PASS.EQ.1.0) GO TO 135
      IF (IT.EQ.NIN) GO TO 135
      IT=IT+1
      IC=IC-1
      IF (IC.EQ.0) GO TO 55
      DO 50 K=K1,K2
      SI(C,K)=SN(K)
   50 SN(K)=0.
      GO TO 25
   55 A=0.
      B1=0.
      B2=0.
      C1=0.
      D1=0.
      D2=0.
      DO 85 K=K1,K2
      B59=29S(1,K)-SN(K)-SI(2,K)
      IF (D59.GT.0.) GO TO 60
      A=A+SI(2,K)-SI(1,K)
      GO TO 65
   60 A=A+D59
      DA=DA+D59
   65 BSI=SI(4,K)-SI(3,K)
```

390

```
      SUBROUTINE PFP4(CLPA,CDPA)
C***  POTENTIAL FLOW PROGRAM SECTION 4--COMPUTES VELOCITIES AND PRESSURE
C***  COEFFICIENTS AT THE PANEL CENTROID POINTS
      DIMENSION CV1(1000)
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XNP(650),YNP(650),ZNP(
     1650),AQP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW
      COMMON /SIK/S1K(650)
      COMMON /VXYZ/VX1(650),VY1(650),VZ1(650),VABS(650)
      COMMON /HOLD/XMH(650),ZMH(650),AQPH(650)
      COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE
    5 FORMAT (1H1,33H POTENTIAL FLOW PROGRAM SECTION 4)
      WRITE (JWRITE,5)
      D=-.5/3.14159265
      J=1
      K1=1
      IF (KNEW.EQ.1)KHOLD=NP
      K2=NP
      BBR=1.0
      CB=0.0
      CL=0.0
      AREA=0.0
C***  READ FIRST BLOCK OF COEFFICIENTS
      ITAPE=KFILE3
      READ (KFILE3)BB,CV1
      IF (BB-BBR) 130,10,130
   10 DO 15 I=K1,K2
      VX1(I)=-1.0-S1K(I)*XNP(I)/D
      VY1(I)=-S1K(I)*YNP(I)/D
      VZ1(I)=-S1K(I)*ZNP(I)/D
   15 CONTINUE
C***  SET UP LOOP OVER QUADS..PICK UP SOURCE,SET UP LOOP OVER CENTROIDS
      JC=2
   20 S1J=S1K(J)
      DO 50 JP=K1,K2
C***  COMPUTE PARTIAL SUM FOR 3 COMPONENTS OF 3 VELOCITIES
      VX1(JP)=VX1(JP)+S1J*CV1(JC)
      VY1(JP)=VY1(JP)+S1J*CV1(JC+1)
      VZ1(JP)=VZ1(JP)+S1J*CV1(JC+2)
      JC=JC+3
C***  READ MORE COEFFICIENTS IF NEEDED
      IF (JC-1000) 50,25,25
   25 JC=2
      IF (BBR-635.0) 30,35,40
   30 READ (KFILE3)BB,CV1
      GO TO 45
   35 REWIND KFILE3
   40 READ (KFILE3)BB,CV1
      BBR=BBR+1.
      IF (BBR-BB) 130,50,130
C***  END OF LOOP OVER CENTROIDS THEN END OF LOOP OVER QUADS.
   50 CONTINUE
      J=J+1
   55 IF (J-NP) 20,20,55
      REWIND KFILE3
   60 GO TO 75

      DS2=S(3,K)-S(2,K)
      DS3=DS1-DS2
      DS5=S(2,K)-S(1,K)
      DS6=DS2-DS5
      DS4=DS2-S(1,K)+SN(K)
      DS7=DS3*DS6-DS5*DS6
      DS8=DS6*DS5-DS4*DS3
      IF (DS7.GT.0.1) GO TO 70
      B1=B1+DS1*DS4-DS2*DS6
      D1=D1-DS7
      GO TO 75
   70 B1=B1+DS1*DS4-DS2*DS6
      D1=D1+DS7
   75 IF (DS8.GT.0.1) GO TO 80
      B2=B2-DS1-DS5*DS2*DS3
      D2=D2-DS8
      GO TO 85
   80 B2=B2+DS1*DS5-DS2*DS3
      D2=D2+DS8
   85 CONTINUE
      A=A/DA
      B1=B1/D1
      B2=B2/D2
      IF (IIT.EQ.6) GO TO 115
      AA=ABS(A-A5)
      IF (AA.GT.-.02) GO TO 100
   90 S(5,K)=A+(SN(K)-S(1,K))*S(1,K)
      SN(K)=0.
      WRITE (JWRITE,95)
   95 FORMAT (29X,17HA EXTRAPOLATION )
      GO TO 125
  100 BB1=50.*ABS(B1-BS1)
      BB2=50.*ABS(B2-BS2)
      BBB=ABS(B1)+ABS(B2)
      IF ((BB1.GT.BBB).OR.(BB2.GT.BBB)) GO TO 115
      DO 105 K=K1,K2
      S(5,K)=S(2,K)+B1*(S(1,K)-S(2,K))+B2*(SN(K)-S(2,K))
  105 SN(K)=0.
      WRITE (JWRITE,110)
  110 FORMAT (29X,17HB EXTRAPOLATION )
      GO TO 125
  115 DO 120 K=K1,K2
      S(5,K)=SN(K)
  120 SN(K)=0.
  125 IC=5
      WRITE (JWRITE,130) A,B1,B2
  130 FORMAT (29X,3E12.3)
      A5=A
      BS1=B1
      BS2=B2
      GO TO 25
  135 DO 140 K=K1,K2
  140 S(1,K)=SN(K)
  145 FORMAT (4X,I3,E18.5)
  150 FORMAT (//13H0 X VELOCITY=,F4.1,15H     Y VELOCITY=,4H 0-0.15H
     1 VELOCITY=,4H 0.0)
      RETURN
      END
```

```
      DIMENSION XC1(650),YC1(650),XC2(650),YC2(650),XC3(650),XC4(650),YC   PFS 3
     1 4(650),X3I(650),Y3I(650),Z3I(650),Z4I(650),DMX1(650),RL            PFS 4
     1(751),YL(751),ZL(751),YC3I(690),SF(51),XCR(51),YCR(51),STRL(751),   PFS 5
     1SKIN(650),DSTAR(650),MGTEST(51)                                     PFS 6
      EQUIVALENCE (YC3,YC2),(XL(1),(XMM(1),VL(1),AGPHI1)),(ZL(1),ZNM(1))   PFS 7
     1)                                                                    PFS 8
      COMMON /MOLD/XNM(650),ZNM(650),AGPH(650)                            PFS 9
      COMMON TITLE(20),XCP(650),YCP(650),ZCP(650),XMP(650),ZMP(650),ZMP   PFS 10
     1(650),AGP(650),REFA,REBODY,NQUAD,IWRITE,NP,KNEW                     PFS 11
      COMMON /VXYZ/VX1(650),VY1(650),VZ1(650),VABS(650)                   PFS 12
      COMMON /BL/VDV(751),SSI(751),VINF,VQ,ADE,DELS(1150),CP1(1150),THT(150),   PFS 13
     1STOTAL,KKK                                                          PFS 14
      COMMON /INOUT/JREAD,JWRITE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE        PFS 15
     15                                                                   PFS 16
      WRITE (JWRITE,5)                                                    PFS 17
    5 FORMAT (///,33H0POTENTIAL FLOW PROGRAM SECTION 5)                   PFS 18
      NB=INP+1                                                            PFS 19
      NLIN=NQUAD                                                          PFS 20
      DO 10 I=1,NB                                                        PFS 21
      IFN=I+11                                                            PFS 22
      IS=IFN-11                                                           PFS 23
      IS=IFN-11                                                           PFS 24
C***  READ PANEL GEOMETRY FROM TAPE                                       PFS 25
      READ (KFILE1Q,(XC1(J),YC1(J),XC2(J),YC2(J),XC3(J),XC4(J),YC4(J),X3   PFS 26
     1(J),Y3(J),Z3(J),X4(J),Y4(J),Z4(J),(SKIP,K=1,3),J=IS,IFN)           PFS 27
      NQ=0                                                                PFS 28
      IF (NQ.NE.IS) GO TO 315                                            PFS 29
   10 CONTINUE                                                           PFS 30
      DO 15 I=1,NP                                                        PFS 31
      D1=(XC1(I)**2+YC1(I)**2)*1.-01                                      PFS 32
      D2=(XC2(I)**2+VC2(I)**2)*1.-01                                      PFS 33
      D3=(XC3(I)**2+YC3(I)**2)*1.-01                                      PFS 34
      D4=(XC4(I)**2+YC4(I)**2)*1.-01                                      PFS 35
      DMX(I)=AMAX1(D1,D2,D3,D4)                                           PFS 36
   15 CONTINUE                                                            PFS 37
      MID=70                                                              PFS 38
      MAXJ=75                                                             PFS 39
      MIN,J=1                                                             PFS 40
      V1I=-1.0                                                            PFS 41
      IF (NLIN.EQ.0) GO TO 325                                           PFS 42
   20 FORMAT (1X,38HFOR V INFINITY =-1.0,0.0,0.0, COMPUTE,14,50H STREAM   PFS 43
     1LINES STARTING AT EACH PANEL CENTROID POINT)                       PFS 44
   25 FORMAT (///,1H0,2044)                                              PFS 45
      WRITE (JWRITE,25) (TITLE(I),I=1,20)                                PFS 46
      IF (IWRITE.GT.1) GO TO 30                                          PFS 47
      WRITE (JWRITE,20) NLIN                                             PFS 48
C***  CALCULATE STREAMLINES FOR EACH PANEL CENTROID OMITTING TRIANGULAR   PFS 49
C***  PANELS AT EACH END OF THE BODY. STREAMLINES ARE CALCULATED FROM     PFS 50
C***  THE NOSE OF THE BODY TO 3 POINTS PAST THE PANEL CENTROID.           PFS 51
   30 IUPPER=NLIN+1-2*(MM-1)                                             PFS 52
      ILOWER=2*(MM-1)                                                    PFS 53
      NLTNM1=NLIN-1                                                      PFS 54
      DO 200 II=1,3                                                      PFS 55
      IF (III-2) 35,40,45                                                PFS 56
   35 NSTART=2                                                           PFS 57
      NEND=2*(MM-1)                                                      PFS 58
      NDEL=2                                                             PFS 59
      GO TO 50                                                           PFS 60
   40 NSTART=NEND+1                                                      PFS 61
      NEND=NM-3I*(MM-1)                                                  PFS 62
      NDEL=1
      GO TO 50
```

```
   65 REWIND KFILE5                                                      PF4 58
   70 FORMAT (///,1H0,2044,9H  PAGE =,I15)                               PF4 59
   75 IP=K1+49                                                           PF4 60
      IS=K1                                                              PF4 61
      IPAGE=1                                                            PF4 62
   80 FORMAT (1H0 X FLOW)                                                PF4 63
   85 FORMAT (4H PT.,10X,1HX,9X,1HY,9X,1HZ,13X,2HVX,8X,2HVY,8X,2HVZ,9X,5   PF4 64
     1HABS.V,8X,2HCP,6X,6HSOURCE,5X,8HV NORMAL)                          PF4 65
   90 FORMAT (1X,I3,4X,3F10.5,4X,3F10.5,4X,3F10.5,2F11.5,E12.2)          PF4 66
   95 IF (IP-K2) 100,100,115                                            PF4 67
C***  COMPUTE PRESSURE COEFFICIENT, ABSOLUTE VALUE OF VELOCITY, CL, & CD   PF4 68
  100 WRITE (JWRITE,70) (TITLE(I),I=1,20),IPAGE                          PF4 69
      WRITE (JWRITE,80)                                                  PF4 70
      WRITE (JWRITE,85)                                                  PF4 71
      DO 110 I=IS,IP                                                     PF4 72
      VSQ=VX1(I)**2+VY1(I)**2+VZ1(I)**2                                  PF4 73
      VM=(ABS(VX1(I))+ABS(VY1(I))+ABS(VZ1(I)))*0.79                      PF4 74
      VM=VM+VSQ/VM                                                       PF4 75
      VM=.25*VM+VSQ/VM                                                   PF4 76
      VM=.5*(VM+VSQ/VM)                                                  PF4 77
      VABS(I)=VM                                                         PF4 78
      CP1=1.-VSQ                                                         PF4 79
      VM=VX1(I)+XMP(I)+VY1(I)+YMP(I)+VZ1(I)+ZMP(I)                       PF4 80
      IF (I1.GT.KHOLD) GO TO 110                                         PF4 81
      IF (KNEW.EQ.2) GO TO 105                                           PF4 82
      CL=CL+CP*AGP(I)*ZMP(I)                                             PF4 83
      CD=CD+CP*AGP(I)*XMP(I)                                             PF4 84
      AREA=AREA+AGP(I)                                                   PF4 85
      GO TO 110                                                          PF4 86
  105 CL=CL+CP*AGPH(I)*ZNM(I)                                            PF4 87
      CD=CD+CP*AGPH(I)*XNM(I)                                            PF4 88
      AREA=AREA+AGPH(I)                                                  PF4 89
C***  WRITE CENTROID POINTS,VELOCITIES,CP'S,CL-PRESSURE,& CD-PRESSURE      PF4 90
  110 WRITE (JWRITE,90) I,XCP(I),YCP(I),ZCP(I),VX1(I),VY1(I),VZ1(I),VM,C   PF4 91
     1P,S1(I),VM                                                         PF4 92
      IS=IS+50                                                           PF4 93
      IP=IP+50                                                           PF4 94
      IPAGE=IPAGE+1                                                      PF4 95
      IF (K2-IS) 120,115,95                                             PF4 96
  115 IP=K2                                                              PF4 97
      GO TO 100                                                          PF4 98
  120 CLPA=2.0*CL                                                        PF4 99
      CDPA=2.0*CD                                                        PF4 100
      CLP=CLPA/REFA                                                     PF4 101
      CDP=CDPA/REFA                                                     PF4 102
      WRITE (JWRITE,125) CLP,CDP,REFA,REBODY                            PF4 103
  125 FORMAT (///,18X,35HPRESSURE LIFT AND DRAG COEFFICIENTS,//21X,29I1M   PF4 104
     1P,//25X,14HPRESSURE CD = ,F11.5,/25X,14HPRESSURE CD = ,F11.5,/22X,   PF4 105
     11TREFERENCE AREA = ,F11.5,/21X,18HREYNOLDS NUMBER = ,E11.4,//21X,2   PF4 106
     19(1H*),//)                                                         PF4 107
      GO TO 140                                                          PF4 108
  130 WRITE (JWRITE,135) ITAPE,0B4,08                                   PF4 109
  135 FORMAT (6H1TAPE ,12,17H OUT OF POSITION/14,6F6.1)                 PF4 110
      RETURN                                                            PF4 111
      END                                                               PF4 112
```

```
C***  SUBROUTINE PFP5(NM,NN,AREAAV,AVDELS,AVXCG,CDFA)                      PFS 1
C***  POTENTIAL FLOW PROGRAM SECTION 5--COMPUTES ON-BODY STREAMLINES       PFS 2
```

```
45    NSTART=NEND-1                                                    PFS  63
      MEND=(MM-1)*(MM-1)-1                                             PFS  64
      MDEL=2                                                          PFS  66
50    DO 255 LL=NSTART,MEND,MDEL                                      PFS  67
      MID=0.255                                                       PFS  68
      DIRT=1.                                                         PFS  69
      JI=-1                                                           PFS  70
55    AP=1.                                                           PFS  71
      UX=0.                                                           PFS  72
      UY=0.                                                           PFS  73
      CP=0.                                                           PFS  74
      STAL(MID)=0.                                                    PFS  75
      MQ=LL                                                           PFS  76
      NCP=LL                                                          PFS  77
      LNQ=MQ                                                          PFS  78
      XL(MID)=XCP(LL)                                                 PFS  79
      YL(MID)=YCP(LL)                                                 PFS  80
      ZL(MID)=ZCP(LL)                                                 PFS  81
      JSTOP=MID+3                                                     PFS  82
      J=MID                                                           PFS  83
      JL=J                                                            PFS  84
      XLT=(XL(J)-XCP(NQ))*X3(NQ)+(YL(J)-YCP(NQ))*Y3(NQ)+(ZL(J)-ZCP(NQ))*  PFS  85
     1Z3(NQ)                                                          PFS  86
      YLT=(XL(J)-XCP(NQ))*X4(NQ)+(YL(J)-YCP(NQ))*Y4(NQ)+(ZL(J)-ZCP(NQ))*  PFS  87
     1Z4(NQ)                                                          PFS  88
      XL(J)=XLT*X3(NQ)*YLT*X4(NQ)+XCP(NQ)                             PFS  89
      YL(J)=XLT*Y3(NQ)*YLT*Y4(NQ)+YCP(NQ)                             PFS  90
      ZL(J)=XLT*Z3(NQ)*YLT*Z4(NQ)+ZCP(NQ)                            PFS  91
60    IQT=MOD(INQ,4)+1                                                PFS  92
      GO TO (80,65,70,75),IQT                                         PFS  93
65    NR=NQ+1                                                         PFS  94
      NU=NQ+2                                                         PFS  95
      GO TO 85                                                        PFS  96
70    NR=NQ+1                                                         PFS  97
      NU=NQ-1                                                         PFS  98
      GO TO 85                                                        PFS  99
75    NR=NQ-2                                                         PFS 100
      NU=NQ-1                                                         PFS 101
      GO TO 85                                                        PFS 102
80    NR=NQ-1                                                         PFS 103
      NU=NQ-2                                                         PFS 104
C***  TRANSFORM VELOCITIES TO QUAD. SYSTEM                            PFS 105
85    UXQ=-(VXI(NQ)+VXI(NQ))                                          PFS 106
      UYQ=-(VXI(NQ)+VYI(NQ))                                          PFS 107
      UZQ=-(VXI(NQ)+VZI(NQ))                                          PFS 108
      UXR=-(VXI(NR)+VXI(NR))                                          PFS 109
      UYR=-(VXI(NR)+VYI(NR))                                          PFS 110
      UZR=-(VXI(NR)+VZI(NR))                                          PFS 111
      UXU=-(VXI(NU)+VXI(NU))                                          PFS 112
      UYU=-(VXI(NU)+VYI(NU))                                          PFS 113
      UZU=-(VXI(NU)+VZI(NU))                                          PFS 114
      UD=UXQ*X3(NQ)+UYQ*Y3(NQ)+UZQ*Z3(NQ)                             PFS 115
      VD=UXQ*X4(NQ)+UYQ*Y4(NQ)+UZQ*Z4(NQ)                            PFS 116
      CSR=1./(XMP(NQ)*XMP(NQ)*YMP(NQ)*YMP(NR)*ZMP(NR)*ZMP(NR))       PFS 117
      UT=(UXR*X3(NR)+UYR*Y3(NR)+UZR*Z3(NR))                           PFS 118
      XMR=(X3(NR)*X3(NQ)+Y3(NR)*Y3(NQ)+Z3(NR)*Z3(NQ))*CSR             PFS 119
      XYR=(X4(NR)*X3(NQ)+Y4(NR)*Y3(NQ)+Z4(NR)*Z3(NQ))               PFS 120
      UR=UT*XR+VT*XYR                                                 PFS 121
                                                                      PFS 122


      VXR=(X3(NR)*X4(NQ)+Y3(NR)*Y4(NQ)+Z3(NR)*Z4(NQ))                PFS 123
      VR=(X4(NR)*X4(NQ)+Y4(NR)*Y4(NQ)+Z4(NR)*Z4(NQ))                PFS 124
      VR=UT*XYR+VT*XYR                                               PFS 125
      UU=UXU*X3(NQ)+UYU*Y3(NQ)+UZU*Z3(NQ)                            PFS 126
      CSU=(XMP(NQ)*XMP(NU)+YMP(NU)*YMP(NQ)*ZMP(NQ)*ZMP(NU))         PFS 127
      VU=(UXU*X4(NQ)+UYU*Y4(NQ)+UZU*Z4(NQ))/CSU                     PFS 128
C***  FIND RELATIVE COORDINATES OF NEIGHBORING QUADS.                PFS 129
      XD=XCP(NR)-XCP(NQ)                                             PFS 130
      YD=YCP(NR)-YCP(NQ)                                             PFS 131
      ZD=ZCP(NR)-ZCP(NQ)                                             PFS 132
      XT=XD*X3(NR)+YD*Y3(NR)+ZD*Z3(NR)                              PFS 133
      YT=XD*X4(NR)+YD*Y4(NR)+ZD*Z4(NR)                              PFS 134
      ZT=XD*XMP(NR)+YD*YMP(NR)+ZD*ZMP(NR)                           PFS 135
      YT=-4*SQRT((YT**2+ZT**2)+YT*CSR*YT)*CSR*.16666667             PFS 136
      XR=XT*XR*XR+YT*XYR                                             PFS 137
      VR=XT*XYR+YT*XYR                                               PFS 138
      XD=XCP(NU)-XCP(NQ)                                             PFS 139
      YD=YCP(NU)-YCP(NQ)                                             PFS 140
      ZD=ZCP(NU)-ZCP(NQ)                                             PFS 141
      XU=XD*X3(NQ)+YD*Y3(NQ)+ZD*Z3(NQ)                              PFS 142
      ZT=XD*X4(NQ)+YD*Y4(NQ)+ZD*Z4(NQ)                              PFS 143
      ZT=XD*XMP(NQ)+YD*YMP(NQ)+ZD*ZMP(NQ)                           PFS 144
      VU=(4.*SQRT((YT**2+ZT**2)+YT/CSU*YT)*.16666667                PFS 145
      DEN=1./(XR*VU-XU*VR)                                           PFS 146
      U1=((UR-UQ)*VU-(UU-UQ)*VR)*DEN                                PFS 147
      U2=-((UR-UQ)*XU-(UU-UQ)*XR)*DEN                               PFS 148
      V1=((VR-VQ)*VU-(VU-VQ)*VR)*DEN                                PFS 149
      V2=-((VR-VQ)*XU-(VU-VQ)*XR)*DEN                               PFS 150
C***  FIND VELOCITY AT STREAMLINE POINT                             PFS 151
      USL=UQ+U1*XLT+U2*YLT                                           PFS 152
      VSL=VQ+V1*XLT+V2*YLT                                           PFS 153
      UXP=USL*X3(NQ)+VSL*X4(NQ)                                     PFS 154
      UYP=USL*Y3(NQ)+VSL*Y4(NQ)                                     PFS 155
      UZP=USL*Z3(NQ)+VSL*Z4(NQ)                                     PFS 156
      VSQ=USL**2+VSL**2                                             PFS 157
      DEN=VSQ*SQRT(VSQ)                                             PFS 158
C***  FIND LOCAL STREAM FUNCTION                                    PFS 159
      CXY=(U1*VQ**2-V2*UQ**2)/VSQ                                   PFS 160
      CYY=U2-UQ*VQ*(U1-V2)/VSQ                                      PFS 161
      CXX=U2-CYY-V1                                                 PFS 162
      CO=XLT*VQ-VLT*UQ-CXY*XLT*YLT-CYY*YLT**2-CXX*XLT**2           PFS 163
C***  FIND STREAM FUNCTION AT CORNER POINTS                         PFS 164
      XCR(1)=XC1(NQ)                                                PFS 165
      XCR(2)=XC2(NQ)                                                PFS 166
      XCR(3)=XC3(NQ)                                                PFS 167
      XCR(4)=XC4(NQ)                                                PFS 168
      XCR(5)=XCR(1)                                                 PFS 169
      YCR(1)=YC1(NQ)                                                PFS 170
      YCR(2)=YC2(NQ)                                                PFS 171
      YCR(3)=YC3(NQ)                                                PFS 172
      YCR(4)=YC4(NQ)                                                PFS 173
      YCR(5)=YCR(1)                                                 PFS 174
90    SF(N)=CO-VQ*XCR(N)+UQ*YCR(N)+UG*YCR(N)+CXY*XCR(N)*YCR(N)+CYY*YCR(N)**2+CXX*   PFS 175
     1CR(N)**2                                                      PFS 176
      TEST=0                                                        PFS 177
      SF(5)=SF(1)                                                   PFS 178
      DO 105 M=1,4                                                  PFS 179
      IF (SF(M)*SF(M+1).GE.0.) GO TO 105                            PFS 180
      XM=(XCR(N)+XCR(M+1))*0.5                                      PFS 181
                                                                    PFS 182
```

```
C***    YM=(YCR(N)+YCR(N+1))*.5
C       FIND INTERSECTION WITH SIDE OF QUAD.
        SFN=CD-VDX*XM-UQ*YM+CXY*XM*YM+CYY*YM**2+CXX*XM**2
        AC=2.*(SF(N)-2.*SF(M)+SF(M+1))
        BC=3.*SF(N)-4.*SF(M)+SF(M+1)
        IF (AC.EQ.0) GO TO 95
        SR=SQRT(BC**2-4.*AC*SF(N))
        TP=(BC-SR)/(2.*AC)
        IF (TP.LE.1..AND.TP.GE.0.) GO TO 100
        TP=(BC-SR)/(2.*AC)
        GO TO 100
95      IF (BC.EQ.0) GO TO 105
        TP=SF(N)/BC
100     XNTP=(1.-TP)*XCR(N)+TP*XCR(N+1)
        YNTP=(1.-TP)*YCR(N)+TP*YCR(N+1)
        TESTP=((XNTP-XLT)*UQ+(YNTP-YLT)*VQ)*DIRT
        IF (TESTP.LE.TEST) GO TO 105
        TEST=TESTP
        XNT=XNTP
        YNT=YNTP
105     CONTINUE
        IF (TEST.EQ.0) GO TO 160
C***    AVERAGE LAST VELOCITY AND CURVATURE
        UX=(UX+UXP)*AF
        UY=(UY+UYP)*AF
        UZ=(UZ+UZP)*AF
        CP=1.-(UX**2+UY**2+UZ**2)
        UABS(J)=SQRT(1.-CP)
        IF (J.GE.JSTOP) GO TO 165
C***    COMPUTE VELOCITY AT NEXT POINT
        J1=J
        J=J+J1
        USL=UQ+XNT*U1+YNT*U2
        VSL=VQ+XNT*V1+YNT*V2
        UX=USL*X3(NQ)+VSL*X4(NQ)
        UY=USL*Y3(NQ)+VSL*Y4(NQ)
        UZ=USL*Z3(NQ)+VSL*Z4(NQ)
        VSQ=USL**2+VSL**2
        CORD=SQRT((XNT-XLT)**2+(YNT-YLT)**2)*DIRT
        STNL(J)=STNL(JL)+CORD
        CP=1.-VSQ
        UABS(J)=SQRT(1.-CP)
        AF=.5
        LNQ=NQ
        KL(J)=XNT*X3(NQ)+YNT*X4(NQ)+XCP(NQ)
        YL(J)=XNT*Y3(NQ)+YNT*Y4(NQ)+YCP(NQ)
        ZL(J)=XNT*Z3(NQ)+YNT*Z4(NQ)+ZCP(NQ)
        IF (J.LE.MINJ.OR.J.GE.MAXJ) GO TO 245
C***    PROCEDURE FOR FINDING NEXT QUAD. WAS MODIFIED SO THAT, DEPENDING
C***    ON THE DIRECTION, THERE ARE ONLY 5 POSSIBLE QUADS. TO TEST
        KROD=MOD(NQ,2)
        NTEST=5
        IF (DIRT) 120,110,135
110     WRITE (JWRITE,115) DIRT
115     FORMAT (1H1,1X,7HDIRT = ,F10.7,26HPROGRAM TERMINATED IN PFP7)
        CALL EXIT
120     IF (KMOD) 125,130,125
C***    LAST QUAD. IS ODD & WE ARE TESTING AGAINST THE STREAM DIRECTION
125     NQTEST(1)=NQ-2*(NN-1)+1
        NQTEST(2)=NQ+2
```

```
        NQTEST(3)=NQ-2
        NQTEST(4)=NQTEST(1)+2
        NQTEST(5)=NQTEST(1)-2
        IF (NQ.NE.NL(NN)) GO TO 150
        NQTEST(2)=NQTEST(5)
        NTEST=3
        GO TO 150
130     NQTEST(1)=NQ-1
        NQTEST(2)=NQ+2
        NQTEST(3)=NQ-2
        NQTEST(4)=NQ+1
        NQTEST(5)=NQ-1
        IF (NQ.GT.(LOWER) GO TO 150
        NQTEST(2)=NQTEST(2)
        NQTEST(3)=NQTEST(3)
        NTEST=2
        IF (NQ.EQ.2) NTEST=1
135     IF (KMOD) 140,145,140
C***    LAST QUAD IS ODD & WE ARE TESTING IN THE STREAM DIRECTION
140     NQTEST(1)=NQ+1
        NQTEST(2)=NQ+2
        NQTEST(3)=NQ-2
        NQTEST(4)=NQ+3
        NQTEST(5)=NQ-1
        IF (NQ.LT.(UPPER) GO TO 150
        NTEST=2
        NQTEST(1)=NQTEST(3)
        IF (NQ.EQ.NL(NN)) NTEST=1
        NTEST=4
        NQTEST(3)=NQTEST(5)
C***    LAST QUAD IS EVEN & WE ARE TESTING IN THE STREAM DIRECTION
145     NQTEST(1)=NQ+2*(NN-1)-1
        NQTEST(2)=NQ+2
        NQTEST(3)=NQ-2
        NQTEST(4)=NQTEST(1)+2
        NQTEST(5)=NQTEST(1)-2
        IF (NQ.NE.2) GO TO 150
        NTEST=4
        NQTEST(3)=NQTEST(5)
150     M=1
155     I=NQTEST(M)
        NQ=I
        TEST=((XL(J)-XCP(I))**2+(YL(J)-YCP(I))**2+(ZL(J)-ZCP(I))**2-DMX(I))
        IF (TEST.GT.0.) GO TO 160
        DS1=XC1(I)-XC2(I))**2+(YC1(I)-YC2(I))**2+(ZC1(I)-ZC2(I))**2
        DS2=XC2(I)-XC3(I))**2+(YC2(I)-YC3(I))**2+(ZC2(I)-ZC3(I))**2
        DS3=XC3(I)-XC4(I))**2+(YC3(I)-YC4(I))**2+(ZC3(I)-ZC4(I))**2
        DS4=XC4(I)-XC1(I))**2+(YC4(I)-YC1(I))**2+(ZC4(I)-ZC1(I))**2
        XLT=XL(J)-XCP(I)+X3(I)*XL(J)-YCP(I))*Y3(I)+(ZL(J)-ZCP(I))*Z3(I)
        YLT=XL(J)-XCP(I))*X4(I)+(YL(J)-YCP(I))*Y4(I)+(ZL(J)-ZCP(I))*Z4(I)
        XLT=(XL(J)-XCP(I))*XNP(I)+(YL(J)-YCP(I))*YNP(I)+(ZL(J)-ZCP(I))*ZNP
     1(I))
        ZSQ=ZLT**2
        TEST=ZSQ-1.*DMX(I))
        IF (TEST.GT.0.) GO TO 160
        RC1=SQRT(ZSQ*(XLT-XC1(I))**2+(YLT-YC1(I))**2)
        RC2=SQRT(ZSQ*(XLT-XC2(I))**2+(YLT-YC2(I))**2)
        RC3=SQRT(ZSQ*(XLT-XC3(I))**2+(YLT-YC3(I))**2)
        RC4=SQRT(ZSQ*(XLT-XC4(I))**2+(YLT-YC4(I))**2)
```

```
      TEST=(RC1-RC2)**2)-DS1**.3
      IF (TEST.LT.0.) GO TO 60
      TEST=(RC2-RC3)**2)-DS2**.3
      IF (TEST.LT.0.) GO TO 60
      TEST=(RC3-RC4)**2)-DS3**.3
      IF (TEST.LT.0.) GO TO 60
      TEST=(RC4-RC1)**2)-DS4**.3
      IF (TEST.LT.0.) GO TO 60
  160 M=M+1
      IF (M.LE.NTEST) GO TO 155
      IF (IDIR.GT.0.) GO TO 170
C*** CHANGE THE DIRECTION IN WHICH THE STREAMLINE IS TRACED
      DIRT=1.
      JI=1
  170 JMAX=J
      GO TO 55
      SSS=STML(JMIN)
      DO 175 J=JMIN,JMAX
      STML(J)=STML(J)-SSS
  175 IF ((STML(JMIN+3)-STML(JMIN)).LT.0.0(STML(JMIN+1)-STML(JMIN))) GO
     1TO 185
      KEY=1
      JJ=JMIN
  185 KK=JMIN
      JMIN=JMIN+1
  180 XL(KK)=(XL(JJ)+XL(KK))/2.0
      YL(KK)=(YL(JJ)+YL(KK))/2.0
      ZL(KK)=(ZL(JJ)+ZL(KK))/2.0
      UABS(KK)=(UABS(JJ)+UABS(KK))/2.0
      IF (KEY.EQ.1) GO TO 185
      GO TO 190
  185 IF ((STML(JMAX)-STML(JMAX-3)).LT.0.0(STML(JMAX)-STML(JMAX-1))) GO
     1TO 190
      KEY=2
      KK=JMAX-1
      JJ=JMAX
      JMAX=JMAX-2
      GO TO 180
  190 JMM=JMIN+1
      JMI=JMAX-2
      AF=1.
      L=JMM
      IF (IWRITE.GT.1) GO TO 200
      WRITE (JWRITE,195) NCP
  195 FORMAT (///,37H0 LINE PASSING THROUGH QUADRILATERAL ,I4)
  200 DO 220 J=JMM,JMX
      IF ((STML(J+2)-STML(L-1)).LT.0.0(STML(J+1)-STML(L))) GO TO 215
      IF (IWRITE.GT.1) GO TO 210
      WRITE (JWRITE,205) XL(L),YL(L),ZL(L),XL(J+1),YL(J+1),ZL(J+1)
  205 FORMAT (14H POINT DELETED,10X,3F12.5,10X,3F12.5)
  210 STML(L)=(AF*STML(L)+STML(J+1))/(AF+1.)
      XL(L)=(AF*XL(L)+XL(J+1))/(AF+1.)
      YL(L)=(AF*YL(L)+YL(J+1))/(AF+1.)
      ZL(L)=(AF*ZL(L)+ZL(J+1))/(AF+1.)
      UABS(L)=(AF*UABS(L)+UABS(J+1))/(AF+1.)
      AF=AF+1.
      IF (L.LE.MID)MID=MID-1
      GO TO 220
  215 AF=1.
      L=L+1
      K=J+1
      STML(L)=STML(K)
      XL(L)=XL(K)
      YL(L)=YL(K)
      ZL(L)=ZL(K)
      UABS(L)=UABS(K)
  220 CONTINUE
      L=L+1
      STML(L)=STML(JMAX)
      XL(L)=XL(JMAX)
      YL(L)=YL(JMAX)
      ZL(L)=ZL(JMAX)
      UABS(L)=UABS(JMAX)
      JMAX=L
      IF (IWRITE.GT.1) GO TO 230
      WRITE (JWRITE,225)
  225 FORMAT (3H0 I,6X,1HX,9X,1HY,9X,1HZ,13X,2HCP,8X,2HSL,7X,4HUABS)
  230 K=0
      DO 235 I=JMIN,JMAX
      K=K+1
      VOV(K)=UABS(I)
      SS(K)=STML(I)
      IF (IWRITE.GT.1) GO TO 235
C*** WRITE POINTS,VELOCITIES,&CP''S FOR THE STREAMLINE POINTS CALCULATED
      CP=1.0-UABS(I)**2
      WRITE (JWRITE,240) K,XL(I),YL(I),ZL(I),CP,STML(I),UABS(I)
  235 CONTINUE
  240 FORMAT (1X,I3,3F10.5,4X,3F10.5)
C*** CALL ROUTINE TO CALCULATE BOUNDARY LAYER OVER THE STREAMLINE
      CALL BLCOMT(K,IWRITE)
      K2=IMID-JMIN1KKK=1
C*** HOLD THE DISPLACEMENT THICKNESS & THE WALL SHEAR VALUES AT THE
C*** CENTROID POINT FOR WHICH THE STREAMLINE WAS CALCULATED
      SKIN(L)=CFI(K2)
      DSTAR(L)=DELS(K2)
      GO TO 255
  245 WRITE (JWRITE,195) NCP
      WRITE (JWRITE,250)
  250 FORMAT (35H PROBABLE ERROR - LINE IS VERY LONG)
      GO TO 165
  255 CONTINUE
      IF (II-2) 260,280,270
  260 JEND=MENO-1
C*** ESTIMATE DSTAR & SKIN FOR TRIANGLES AT THE NOSE AND TAIL OF BODY
      DO 265 J=1,JEND,2
      DSTAR(J)=DSTAR(J+1)/3.0
  265 SKIN(J)=SKIN(J+1)/3.0
      GO TO 280
  270 JSTART=NSTART+1
      DO 275 J=JSTART,NLIM,2
      DSTAR(J)=DSTAR(J-1)
  275 SKIN(J)=SKIN(J-1)
  280 CONTINUE
C*** END OF LOOP ON PANEL CENTROIDS EXCEPT TRIANGLES
      WRITE (JWRITE,285)
  285 FORMAT (///,1X,56HSUMMARY OF BOUNDARY LAYER INFORMATION FOR QUADRI
     1LATERALS//2X,5HNQUAD,6X,1HX,9X,1HY,9X,1HZ,1HX,5HDSTAR,5X,4HSKIN/)
      CDF=0.0
      AREA=0.0
```

PFS 303
PFS 304
PFS 305
PFS 306
PFS 307
PFS 308
PFS 309
PFS 310
PFS 311
PFS 312
PFS 313
PFS 314
PFS 315
PFS 316
PFS 317
PFS 318
PFS 319
PFS 320
PFS 321
PFS 322
PFS 323
PFS 324
PFS 325
PFS 326
PFS 327
PFS 328
PFS 329
PFS 330
PFS 331
PFS 332
PFS 333
PFS 334
PFS 335
PFS 336
PFS 337
PFS 338
PFS 339
PFS 340
PFS 341
PFS 342
PFS 343
PFS 344
PFS 345
PFS 346
PFS 347
PFS 348
PFS 349
PFS 350
PFS 351
PFS 352
PFS 353
PFS 354
PFS 355
PFS 356
PFS 357
PFS 358
PFS 359
PFS 360
PFS 361
PFS 362
PFS 363
PFS 364
PFS 365
PFS 366
PFS 367
PFS 368
PFS 369
PFS 370
PFS 371
PFS 372
PFS 373
PFS 374
PFS 375
PFS 376
PFS 377
PFS 378
PFS 379
PFS 380
PFS 381
PFS 382
PFS 383
PFS 384
PFS 385
PFS 386
PFS 387
PFS 388
PFS 389
PFS 390
PFS 391
PFS 392
PFS 393
PFS 394
PFS 395
PFS 396
PFS 397
PFS 398
PFS 399
PFS 400
PFS 401
PFS 402
PFS 403
PFS 404
PFS 405
PFS 406
PFS 407
PFS 408
PFS 409
PFS 410
PFS 411
PFS 412
PFS 413
PFS 414
PFS 415
PFS 416
PFS 417
PFS 418
PFS 419
PFS 420
PFS 421
PFS 422

396

```
C***   VELOCITY VERSUS SURFACE DISTANCE FROM THE NOSE OF THE BODY
       CALL SPLINE(NP,VOV,SS,VCOEF)
       NPP=1
       SEND=SS(NP)
       KKK=2
       K=1
 5     S(I)=SS(I)
       V(I)=VOV(I)
       DUDS(I)=VCOEF(3,1)
       DO 5 I=1,N
       NN(SS(I-1)-SS(I))/KKK
       DO 5 J=1,KKK
       K=K+1
       S(K)=S(K-1)+NN
       V(K)=(VCOEF(1,I)+S(K)*VCOEF(2,I))*S(K)*VCOEF(3,I))*S(K)*VCOEF(4,I
      1)
       DUDS(K)=(3.0D0*VCOEF(1,I)*S(K)+2.0D0*VCOEF(2,I))*S(K)+VCOEF(3,I)
       NPP=KKK*(NP-1)+1
 5     CONTINUE
       DO 10 I=1,NPP
 10    DUDS(I)=DUDS(I)/STOTAL
       CALL LAMINR(SS,V,DUDS,STRAM,II,NPP,IWRITE)
       IF (II.EQ.NPP) GO TO 15
       CALL TURB2(S,V,DUDS,SEND,II,NPP)
 15    IF (IWRITE.GT.1) GO TO 25
       WRITE (IWRITE,20)
 20    FORMAT (/,6X,1HS,13X,1HV,10X,4HDUDS,9X,5HMMEAN,8X,6HDELTAS,7X,6HTH
      1ETA,9X,3HTAM,9X,5HCF/)
 25    S(I)=S(I)*STOTAL
       V(I)=V(I)*VINF
       DUDS(I)=DUDS(I)*VINF/STOTAL
       DELS(I)=DELS(I)*STOTAL
 30    DO 35 I=1,NPP,KKK
       WRITE (IWRITE,40) S(I),V(I),DUDS(I),HMEAN(I),DELS(I),THT(I),TAM(I)
      1,CF(I)
 35    CONTINUE
 40    FORMAT (1X,8(F10.5,3X))
       RETURN
       END
```

```
       SUBROUTINE LAMINR(X,UDEL,DUDX,XTRAM,II,NPP,IWRITE)
C***   CALCULATES LAMINAR BOUNDARY LAYER BY MOMENTUM INTEGRAL
       DIMENSION X(150),UDEL(150),DUDX(150)
       COMMON /ROUND/TAM(150),UOEL(150),VCOEF(4,75),NP
      1STOTAL,KKK
       COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CF(150),THT(150),
       11=1
       HX(I)=X(I)-X(II)
       IF (UDEL(II).LT.1.0E-06) GO TO 5
C***   STARTING CONDITIONS ASSUMING FLAT PLATE
       Z=0.0
       THT(I)=0.0
       DELS(I)=0.0
       TAM(I)=5000.0
```

```
C***   WRITE POINTS,DSTAR,& SKIN FOR EACH CENTROID. COMPUTE FRICTION CD
       DO 295 J=1,MLIN
       WRITE (IWRITE,290) J,RCP(J),YCP(J),ZCP(J),DSTAR(J),SKIN(J)
 290   FORMAT (4X,I3,3F10.5,4X,2F10.5)
       CDF=CDF-SKIN(J)*AGP(J)*OVRI(J)/(0.5*VABS(J))
       AREA=AREA+AGP(J)
 295   CONTINUE
       AREAAV=AREA/MLIN
       CDFA=2.0*CDF
       CDF=CDFA/REFA
 300   FORMAT (////,15X,29HFRICTION DRAG COEFFICIENT,//,13X,29(1H*),//17X,1
      1HHFRICTION CD = ,F11.5,//16X,1THREFERENCE AREA = ,F11.5,//13X,18HMEY
      1NOLDS NUMBER = ,E11.4/17X,14HBODY LENGTH = ,F11.5/,13X,2M(1H*),///)
       WRITE (IWRITE,300) CDF,REFA,REBODY,STOTAL
       REWIND KFILE3
       REWIND KFILE4
       REWIND KFILE5
       SUMA=0.0
       AVXCG=0.0
       AVAREA=0.0
       AVDELS=0.0
       NOO=MLIN-2MM+4
C***   CALCULATE AVERAGE DSTAR,AVERAGE PANEL AREA, AND AVERAGE K-ORDINATE
C***   FOR THE THIRD STATION OF INPUT POINTS FROM THE END OF BODY. THESE
C***   QUANTITIES ARE USED TO CALCULATE THE WAKE-BODY COORDINATES.
       DO 305 I=NOO,MLIN,2
       AVXCG=AVXCG+RCP(I-1)*AGP(I-1)
       AVDELS=AVDELS+DSTAR(I-1)*AGP(I-1)
       AVAREA=AVAREA+AGP(I-1)
 305   SUMA=SUMA+AGP(I)
       AVDELS=AVDELS/AVAREA
       AVXCG=AVXCG/AVAREA
       SUMA=SUMA/MM-1)
       AREAAV=SQRT(AREAAV*SUMA)
       DO 310 I=1,MLIN
       2MM(I)=ZNP(I)
 310   AGPHI(I)=AGP(I)
       WRITE (IWRITE,320) IS,MQ
 320   FORMAT (14H TAPE 04 ERROR,2I4)
       REWIND KFILE
       RETURN
       END
```

```
       SUBROUTINE BLCONT(KNP,IWRITE)
C***   BOUNDARY LAYER CONTROL ROUTINE WHICH CALLS LAMINR & TURB2
C***   LAMINAR AND TURBULENT ROUTINES WHICH USE MOMENTUM INTEGRAL METHOD
       DIMENSION V(150),S(150),HMEAM(150),VCOEF(4,75),NP
       COMMON /ROUND/TAM(150),SS(75),VINF,VO,ROE,DELS(150),CF(150),THT(150),
      1STOTAL,KKK
       COMMON /BL/VOV(75),SS(75),VINF,VO,ROE,DELS(150),CF(150),THT(150),
       IF (UDEL(II).LT.1.0E-06) GO TO 5
C***   SPLINE IS CALLED TO CURVE FIT THE STREAMLINE VALUES OF ABSOLUTE
```

```
      SUBROUTINE FFK(AKD,FD,F1D,F2D)                                    FFK   1
C***  ROUTINE FOR NUMERICAL INTERPOLATION OF LAMINAR B.L. FUNCTIONS     FFK   2
      DIMENSION AK(55),F(55),F1(55),F2(55)                             FFK   3
      COMMON /INOUT/JREAD,JWRITE,KFILE,KFILE1,KFILE2,KFILE3,KFILE4,KFILE FFK   4
     1 5                                                                FFK   5
      DATA AK/0.094815,0.094083,0.093166,0.091882,0.090234,0.08        FFK   6
     18223,0.085855,0.083134,0.080068,0.076664,0.072930,0.068877,0.06451 FFK   7
     16,0.059857,0.054912,0.049497,0.044223,0.038506,0.032562,0.028405,0 FFK   8
     1.020054,0.013524,0.006833,0.000000,-0.006951,-0.014021,-0.021170,- FFK   9
     10.028387,-0.035651,-0.042943,-0.050244,-0.057532,-0.064789,-0.0719 FFK  10
     195,-0.079129,-0.086171,-0.093104,-0.099906,-0.106558,-0.113043,-0. FFK  11
     1119341,-0.125434,-0.131304,-0.136935,-0.142309,-0.147411,-0.152224 FFK  12
     1,-0.156735,-0.160927,-0.164789,-0.168307,-0.171470,-0.17426 7,-0.17 FFK  13
     16497/                                                             FFK  14
      DATA F/-0.094815,-0.093915,-0.091177,-0.086542,-0.079963,-0.071400 FFK  15
     1,-0.060818,-0.048195,-0.033512,-0.016759,0.002068,0.022964,0.04591 FFK  16
     19,0.070916,0.097932,0.126938,0.157897,0.190770,0.225508,0.262060,0 FFK  17
     1.300369,0.340371,0.381999,0.425181,0.469861,0.515894,0.563264,0.61 FFK  18
     11853,0.661571,0.712321,0.764006,0.816516,0.867752,0.923601,0.97795 FFK  19
     12,1.032691,1.087700,1.142862,1.198055,1.253157,1.308043,1.362589,1 FFK  20
     1.416665,1.470145,1.522902,1.574802,1.625771,1.675522,1.724079,1.77 FFK  21
     11259,1.816936,1.860975,1.903251,1.943633,1.981994/                FFK  22
      DATA F1/2.250000,2.250688,2.252684,2.255992,2.260349,2.265904,2.27 FFK  23
     12545,2.280231,2.288928,2.298609,2.309246,2.320828,2.333333,2.34675 FFK  24
     11,2.361073,2.376291,2.392406,2.409414,2.427320,2.446130,2.465849,2 FFK  25
     1.486485,2.508057,2.530574,2.554053,2.578518,2.603985,2.630480,2.65 FFK  26
     18031,2.686666,2.716418,2.747321,2.779412,2.812734,2.847327,2.88324 FFK  27
     12,2.920529,2.959246,2.999447,3.041200,3.084570,3.129634,3.176471,3 FFK  28
     1.225146,3.275810,3.328505,3.383357,3.440479,3.499999,3.562056,3.62 FFK  29
     16788,3.694360,3.764940,3.838720,3.915905/                        FFK  30
      DATA F2/0.355556,0.355291,0.354511,0.353236,0.351470,0.349229,0.34 FFK  31
     16528,0.343380,0.339800,0.335801,0.331397,0.326601,0.321428,0.31589 FFK  32
     10,0.310005,0.303782,0.297237,0.290383,0.283234,0.275804,0.268107,0 FFK  33
     1.260156,0.251966,0.243549,0.234921,0.226093,0.217082,0.207899,0.19 FFK  34
     18559,0.189077,0.179464,0.169736,0.159906,0.149987,0.139995,0.12994 FFK  35
     11,0.119841,0.109708,0.099555,0.089397,0.079247,0.069120,0.059028,0 FFK  36
     1.048985,0.039007,0.029105,0.019294,0.009588,0.000000,-0.009455,-0. FFK  37
     1018765,-0.027914,-0.036890,-0.045678,-0.054266/                   FFK  38
      IF (AKD.LE.AK(1)) GO TO 5                                         FFK  39
      FD=F(1)                                                           FFK  40
      F1D=F1(1)                                                         FFK  41
      F2D=F2(1)                                                         FFK  42
      AKD=AK(1)                                                         FFK  43
      RETURN                                                            FFK  44
    5 IF (AKD.GT.AK(55)) GO TO 10                                       FFK  45
      FD=F(55)                                                          FFK  46
      F1D=F1(55)                                                        FFK  47
      F2D=F2(55)                                                        FFK  48
      AKD=AK(55)                                                        FFK  49
      RETURN                                                            FFK  50
   10 DO 15 I=1,54                                                      FFK  51
      IF (AKD.LE.AK(I).AND.AKD.GT.AK(I+1)) GO TO 25                     FFK  52
   15 CONTINUE                                                          FFK  53
      WRITE (JWRITE,20) AKD                                             FFK  54
   20 FORMAT (1X,6HAKD = ,E12.5,22H EXCEEDS ALLOWED RANGE)             FFK  55
      CALL EXIT                                                         FFK  56
   25 FD=F(I)+(F(I+1)-F(I))*(AKD-AK(I))/(AK(I+1)-AK(I))                FFK  57
      F1D=F1(I)+(F1(I+1)-F1(I))*(AKD-AK(I))/(AK(I+1)-AK(I))            FFK  58
```

```
      CF(II)=1000.0                                                     LAM  17
      HMEAN(II)=2.554                                                   LAM  18
      II=2                                                              LAM  19
      GO TO 20                                                          LAM  20
C***  STARTING CONDITIONS ASSUMING STAGNATION POINT                     LAM  21
    5 DUDXO=VCOEF(3,1)*STOTAL                                           LAM  22
      D2UDXO=2.0*VCOEF(2,1)*STOTAL**2                                   LAM  23
      Z=0.0770/DUDXO                                                    LAM  24
      TNT(II)=SQRT(Z*VO/(VINF*STOTAL))                                  LAM  25
      DELS(II)=TNT(II)*2.308                                            LAM  26
      CF(II)=0.664*UDEL(II)/(TNT(II)*VINF*STOTAL/VO)                   LAM  27
      TAH(II)=CF(II)*0.5*RQE*VINF*VINF                                  LAM  28
      HMEAN(II)=2.308                                                   LAM  29
      II=2                                                              LAM  30
      GO TO 15                                                          LAM  31
   10 II=II+1                                                           LAM  32
      H=X(II)-X(II-1)                                                   LAM  33
      GO TO 20                                                          LAM  34
   15 AM1=H*(-0.0652*D2UDXO/(DUDXO**2))                                LAM  35
      GO TO 25                                                          LAM  36
   20 AKD=Z*DUDX(II-1)                                                  LAM  37
      CALL FFK(AKD,FD,F1D,F2D)                                          LAM  38
      AM1=H*FD/UDE(II-1)                                                LAM  39
   25 UDELP=0.5*UDEL(II-1)*UDEL(II))                                    LAM  40
      DUDXP=0.5*(DUDX(II-1)+DUDX(II))                                   LAM  41
      AKD=(Z+AM1/2.0)*DUDXP                                             LAM  42
      CALL FFK(AKD,FD,F1D,F2D)                                          LAM  43
      AM2=H*FD/UDELP                                                    LAM  44
      AKD=(Z+AM2/2.0)*DUDXP                                             LAM  45
      CALL FFK(AKD,FD,F1D,F2D)                                          LAM  46
      AM3=H*FD/UDELP                                                    LAM  47
      AKD=(Z+AM3)*DUDX(II)                                              LAM  48
      CALL FFK(AKD,FD,F1D,F2D)                                          LAM  49
      AM4=H*FD/UDEL(II)                                                 LAM  50
      Z=Z+(AM1+2.0*AM2+2.0*AM3+AM4)/6.0                                LAM  51
      AKD=Z*DUDX(II)                                                    LAM  52
      CALL FFK(AKD,FD,F1D,F2D)                                          LAM  53
      TNT(II)=SQRT(Z*VO/(VINF*STOTAL))                                 LAM  54
      DELS(II)=TNT(II)*F1D                                              LAM  55
      CF(II)=2.0*F2D*UDEL(II)/((TNT(II)*VINF*STOTAL/VO)               LAM  56
      TAH(II)=CF(II)*0.5*RQE*VINF*VINF                                 LAM  57
      HMEAN(II)=DELS(II)/TNT(II)                                       LAM  58
      IF (TAH(II).LT.0.0) GO TO 40                                     LAM  59
      IF (X(II).GE.XTRAN) GO TO 30                                     LAM  60
      IF (II.EQ.NPP) GO TO 10                                          LAM  61
      GO TO 10                                                         LAM  62
   30 IF (JWRITE.GT.1) RETURN                                          LAM  63
      WRITE (JWRITE,35) X(II),IK                                       LAM  64
   35 FORMAT (/7X,18HTRANSITION AT X = ,F10.5,17H FOR STEP NUMBER ,I3) LAM  65
      RETURN                                                           LAM  66
   40 IF (JWRITE.GT.1) RETURN                                          LAM  67
      IK=II*KKK                                                        LAM  68
      WRITE (JWRITE,45) TAH(II),IK                                     LAM  69
   45 FORMAT (/,27H NEGATIVE VALUE OF TAWST = ,E14.7,21H AT OR AFTER STA LAM  70
     1TION ,I3,1HW ON THE STREAMLINE.,38H FOR MORE INFORMATION ON THE LO LAM  71
     1CATION,/,10H OF LAMINAR SEPARATION USE JWRITE = 0 PRINT OPTION. T   LAM  72
     1HE TURBULENT ROUTINE IS CALLED AT THE SEPARATION POINT.)          LAM  73
      RETURN                                                           LAM  74
      END                                                              LAM  75
```

```
      H3=H1*(HM1-2)
      D=Y(N)*(32.0*H1**2-2.0*H2**21.0*H3)/(H1*H2)/(H1*H2*H3)-Y(NM1)*(11.0*HH
     11.0*42.0*H2+21.0*H3)/(H2*H3)/(H2*H3)-Y(NM1-1)*H1*(11.0*(11.0*H2+H3))/(
     1H1*H2)/H2/H3)-Y(NM1-2)*H1*(11.0*U(11.0*H1*21.0*(H2*H3)/(H2*H3)/H3
      D=3.0*D/H1/16.0
      U(N)=(D-A*U(N-1))/P
C***  SOLVE FOR THE SPLINE COEFFICIENTS CORRESPONDING TO AHLBERG'S
C***  M(I) TO M(N) AND STORE THEM IN THE U(I).
      DO 15 J=1,NM1
      I=N-J
   15 U(I)=Q(I)*U(I+1)*U(I)
C***  FORM THE AA(J,I) COEFFICIENTS FOR THE CONVENTIONAL FORM OF
C***  A CUBIC POLYNOMIAL FROM THE U(I)
      UU=U(1)
      XX=X(1)
      YY=Y(1)
      DO 20 I=1,NM1
      UP=U(I+1)
      XP=X(I+1)
      YP=Y(I+1)
      HH=H(I)
      AA(1,I)=(UP-UU)/HH/6.0
      AA(2,I)=0.5*(XP*UU-XX*UP)/HH+(UU-UP)*HH/6.0+(YP-YY)/HH
      AA(3,I)=0.5*(UP*XX**2-UU*XP*XP)/HH*(UU-UP)*HH/6.0+(UP*XX-UU*XP)*HH/6.0+(YY*
     1XP-YP*XX)/HH
      AA(4,I)=(UU*XP*XP*XP-UP*XX*XX*XX)/HH/6.0+(UP*XX-UU*XP)*HH/6.0+(YY*
     1XP-YP*XX)/HH
      XX=XP
      UU=UP
   20 YY=YP
      RETURN
      END
```

                                                        SPL 46
                                                        SPL 47
                                                        SPL 48
                                                        SPL 49
                                                        SPL 50
                                                        SPL 51
                                                        SPL 52
                                                        SPL 53
                                                        SPL 54
                                                        SPL 55
                                                        SPL 56
                                                        SPL 57
                                                        SPL 58
                                                        SPL 59
                                                        SPL 60
                                                        SPL 61
                                                        SPL 62
                                                        SPL 63
                                                        SPL 64
                                                        SPL 65
                                                        SPL 66
                                                        SPL 67
                                                        SPL 68
                                                        SPL 69
                                                        SPL 70
                                                        SPL 71
                                                        SPL 72
                                                        SPL 73
                                                        SPL 74
                                                        SPL 75
                                                        SPL 76

Sample Output

```
                    POTENTIAL FLOW PROGRAM  SECTION 1

 BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY

NO. OF QUADS.  = 560
NO. OF SECTIONS=   1
MAX. NO. OF ITERATIONS X FLOW 150

VINF = 196.599991     VO =   0.C00160     ROE =   0.002378     REFA = 174.000000     IWRITE =  0

1 PLANES OF SYMMETRY

CONVERGENCE CRITERIA , 0.00010

 SECTION    1

  M      X1            X2            X3            X4            XP            XN            A             CZ4
  N      Y1            Y2            Y3            Y4            YP            YN            FL            CZ5
  P      Z1            Z2            Z3            Z4            ZP            ZN            CZ1           CZ6
QUESTIONABLE POINT -POOR FIT     0.214E-01
WARNING LONG THIN QUAD.
  1   0.12200E 02   0.12200E 02   0.11542E 02   0.11542E 02   0.11764E 02   0.75230E 00   0.12710E 00  -0.32742E 00
  1   0.0           0.0           0.10330E 00   0.0           0.61100E-01   0.30580E 00   0.40240E 00  -0.36831E 00
  1  -0.42710E 00  -0.42710E 00  -0.15979E 01  -0.16354E 01  -0.12201E 01  -0.58340E 00   0.14421E 00   0.34764E 00

  1   0.11542E 02   0.11542E 02   0.10875E 02   0.10875E 02   0.11201E 02   0.41400E 00   0.14916E 00  -0.36101E 00
  2   0.0           0.10330E 00   0.22080E 00   0.0           0.10202E 00   0.87456E-01   0.30272E 00  -0.30378E 00
  2  -0.16354E 01  -0.15979E 01  -0.19229E 01  -0.19271E 01  -0.17774E 01  -0.90570E 00   0.54284E-01  -0.21392E 00
QUESTIONABLE POINT -POOR FIT     0.488E-01
WARNING LONG THIN QUAD.
  2   0.12200E 02   0.12200E 02   0.11542E 02   0.11542E 02   0.11764E 02   0.75227E 00   0.12980E 00  -0.49812E 00
  1   0.0           0.0           0.35830E 00   0.10330E 00   0.18053E 00   0.42249E 00   0.47467E 00  -0.25295E 00
  3  -0.42710E 00  -0.42710E 00  -0.15271E 01  -0.15979E 01  -0.11040E 01  -0.50556E 00   0.13047E 00  -0.40180E 00

  2   0.11542E 02   0.11542E 02   0.10875E 02   0.10875E 02   0.11197E 02   0.49571E 00   0.15557E 00  -0.35366E 00
  2   0.10330E 00   0.35830E 00   0.44580E 00   0.22080E 00   0.30250E 00   0.22647E 00   0.39354E 00  -0.35362E 00
  4  -0.15979E 01  -0.15271E 01  -0.18854E 01  -0.19229E 01  -0.17356E 01  -0.86084E 00   0.56341E-01   0.29032E 00
QUESTIONABLE POINT -POOR FIT     0.129E-01
WARNING LONG THIN QUAD.
  3   0.12200E 02   0.12200E 02   0.11542E 02   0.11542E 02   0.11764E 02   0.78614E 00   0.11692E 00  -0.26264E 00
  1   0.0           0.0           0.51460E 00   0.35830E 00   0.29007E 00   0.35407E 00   0.44208E 00  -0.10291E 00
  5  -0.42710E 00  -0.42710E 00  -0.14437E 01  -0.15271E 01  -0.11326E 01  -0.50517E 00   0.10461E 00   0.13621E 00

  3   0.11542E 02   0.11542E 02   0.10875E 02   0.10875E 02   0.11197E 02   0.49261E 00   0.15461E 00  -0.27330E 00
  2   0.35830E 00   0.51460E 00   0.66250E 00   0.44580E 00   0.49900E 00   0.32127E 00   0.40117E 00  -0.29107E 00
  6  -0.15271E 01  -0.14437E 01  -0.18187E 01  -0.18854E 01  -0.16782E 01  -0.80070E 00   0.45556E-01  -0.19683E 00
QUESTIONABLE POINT -POOR FIT     0.746E-02
WARNING LONG THIN QUAD.
  4   0.12200E 02   0.12200E 02   0.11542E 02   0.11542E 02   0.11764E 02   0.79509E 00   0.12000E 00  -0.11760E 00
  1   0.0           0.0           0.67080E 00   0.51460E 00   0.39513E 00   0.40006E 00   0.45639E 00  -0.25219E 00
  7  -0.42710E 00  -0.42710E 00  -0.13500E 01  -0.14437E 01  -0.10736E 01  -0.45584E 00   0.97180E-01  -0.24117E 00

  4   0.11542E 02   0.11542E 02   0.10875E 02   0.10875E 02   0.11193E 02   0.51476E 00   0.16413E 00  -0.28285E 00
  2   0.51460E 00   0.67080E 00   0.87500E 00   0.66250E 00   0.68437E 00   0.41091E 00   0.40733E 00  -0.26450E 00
  8  -0.14437E 01  -0.13500E 01  -0.17104E 01  -0.18187E 01  -0.15881E 01  -0.75249E 00   0.44809E-01   0.56813E-01
QUESTIONABLE POINT -POOR FIT     0.794E-02
WARNING LONG THIN QUAD.
  5   0.12200E 02   0.12200E 02   0.11542E 02   0.11542E 02   0.11764E 02   0.78499E 00   0.11711E 00  -0.27401E 00
  1   0.0           0.0           0.81870E 00   0.67080E 00   0.49650E 00   0.24646E 00   0.66990E 00  -0.21463E 00
  9  -0.42710E 00  -0.42710E 00  -0.12521E 01  -0.13500E 01  -0.10097E 01  -0.56037E 00   0.10259E 00  -0.18030E 00

  5   0.11542E 02   0.11542E 02   0.10875E 02   0.10875E 02   0.11195E 02   0.51362E 00   0.15710E 00  -0.29020E 00
  2   0.67080E 00   0.81870E 00   0.10542E 01   0.87500E 00   0.85045E 00   0.50227E 00   0.40532E 00  -0.27293E 00
 10  -0.13500E 01  -0.12521E 01  -0.15729E 01  -0.17104E 01  -0.14775E 01  -0.69564E 00   0.45421E-01   0.89894E-01
           .                                         .                                         .

           .                                         .                                         .

           .                                         .                                         .

  M      X1            X2            X3            X4            XP            XN            A             CZ4
  N      Y1            Y2            Y3            Y4            YP            YN            FL            CZ5
  P      Z1            Z2            Z3            Z4            ZP            ZN            CZ1           CZ6
WARNING LONG THIN QUAD.
 20  -0.76045E 01  -0.76045E 01  -0.93335E 01  -0.93335E 01  -0.84373E 01  -0.60089E-01   0.22771E 00  -0.83489E 00
 27   0.14500E 00   0.0           0.0           0.11670E 00   0.65912E-01   0.56427E-01   0.86895E 00  -0.17975E-01
559   0.60030E 00   0.61460E 00   0.51040E 00   0.50210E 00   0.56100E 00   0.99659E 00   0.16787E-01  -0.22861E-01
QUESTIONABLE POINT -POOR FIT     0.218E-01
WARNING LONG THIN QUAD.
 20  -0.93335E 01  -0.93335E 01  -0.12200E 02  -0.12200E 02  -0.10292E 02  -0.12820E 00   0.16993E 00  -0.15092E-01
 28   0.11670E 00   0.0           0.0           0.0           0.38900E-01   0.10389E 00   0.14529E 01  -0.14027E 01
560   0.50210E 00   0.51040E 00   0.93000E-01   0.93000E-01   0.36077E 00   0.97470E 00   0.37115E-01   0.83307E-01

PROBABLE ERROR IN INPUT - SOLID ANGLE =      12.555

POTENTIAL FLOW PROGRAM SECTION 2

 BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY
```

Figure F-5. NCSU BODY program sample output for the Cessna 182 with 560 panels describing the half-body using the IWRITE=0 option.

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY

X VELOCITY=1.0   Y VELOCITY= 0.0   Z VELOCITY= 0.0

ITERATIVE MATRIX SOLUTION INFORMATION

ITERATION	SUM OF CHANGES	A	B1	B2
1	0.49002E 01			
2	0.30960E 01			
3	0.24726E 01			
4	0.20169E 01			
5	0.16831E 01			
		0.5490 00	0.4210 00	0.5970 00
6	0.14274E 01			
7	0.12204E 01			
8	0.10719E 01			
9	0.94923E 00			
10	0.85944E 00			
	A EXTRAPOLATION			
	0.520E 00	0.500E 00	0.281E 00	
11	0.36795E-01			
12	0.34513E-01			
13	0.32543E-01			
14	0.30042E-01			
15	0.29303E-01			
	A EXTRAPOLATION			
	0.513E 00	0.499E 00	0.289E 00	
16	0.95409E-03			

POTENTIAL FLOW PROGRAM SECTION 4

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY   PAGE = 1

X FLOW

PT.	X	Y	Z	VX	VY	VZ	ABS.V	CP	SOURCE	V NORM
1	11.76403	0.06118	-1.22013	-0.44399	0.16321	-0.44654	0.67033	0.55907	0.00434	0.210
2	11.20092	0.10202	-1.77742	-0.96956	0.00935	-0.44133	1.06168	-0.12716	0.03079	0.202
3	11.76403	0.18093	-1.18403	-0.44108	0.23781	-0.45003	0.67996	0.55766	0.00813	0.210
4	11.19695	0.30250	-1.73560	-0.92927	0.06342	-0.47520	1.04360	-0.09366	0.04240	0.190
5	11.76403	0.29097	-1.13263	-0.40395	0.23214	-0.46504	0.69820	0.56667	0.00999	0.230
6	11.19651	0.49908	-1.67816	-0.88044	0.14295	-0.48453	1.02106	-0.04441	0.04705	0.165
7	11.76403	0.39513	-1.07360	-0.39398	0.32364	-0.40319	0.65002	0.57768	0.10534	0.215
8	11.19329	0.68437	-1.58809	-0.85930	0.23575	-0.45914	1.00239	-0.00479	0.05457	0.135
9	11.76403	0.49650	-1.00073	-0.40160	0.28190	-0.43246	0.65404	0.57224	0.10568	0.210
10	11.19407	0.85845	-1.47752	-0.85442	0.33213	-0.39107	0.99663	0.00672	0.05465	0.120
551	-8.42007	0.48947	0.36301	-0.97105	-0.05711	-0.10612	0.97890	0.04254	-0.00712	0.606
552	-10.29103	0.27360	0.22703	-0.94773	-0.10324	-0.10779	0.95941	0.07952	-0.00995	0.160
553	-8.43344	0.41137	0.43911	-0.97447	-0.02700	-0.10522	0.98033	0.03857	-0.00501	-0.408
554	-10.29103	0.23540	0.27503	-0.95057	-0.07627	-0.11025	0.96093	0.07662	-0.00039	0.166
555	-8.43734	0.31905	0.49572	-0.97735	-0.00642	-0.09518	0.98190	0.03571	-0.00390	-0.996
556	-10.29103	0.18400	0.31460	-0.95242	-0.05606	-0.12335	0.96201	0.07454	-0.00043	0.164
557	-8.44324	0.19027	0.53526	-0.98005	0.00569	-0.07624	0.98303	0.03365	-0.00151	-0.164
558	-10.29103	0.11667	0.34953	-0.95433	-0.03232	-0.12473	0.96290	0.07269	0.00795	-0.164
559	-8.43732	0.06591	0.56108	-0.98168	0.00467	-0.06024	0.98334	0.03304	0.00053	-0.106
560	-10.29103	0.05090	0.36077	-0.95559	-0.01030	-0.12371	0.96394	0.07194	-0.00637	0.159

PRESSURE LIFT AND DRAG COEFFICIENTS

```
PRESSURE CL =      0.00444
PRESSURE CD =      0.00488
REFERENCE AREA =  174.00000
REYNOLDS NUMBER = 0.3000E 06
```

POTENTIAL FLOW PROGRAM SECTION 5

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY
FOR V INFINITY = -1.0,0.0,0.0, COMPUTE 560 STREAMLINES STARTING AT EACH PANEL CENTROID POINT

LINE PASSING THROUGH QUADRILATERAL   2

I	X	Y	Z	CP	SL	UABS
1	11.54330	0.07060	-1.62204	0.19993	0.0	0.99444
2	11.20092	0.10202	-1.77742	-0.12716	0.37044	1.06168
3	10.47652	0.06400	-1.92773	-0.34795	0.73449	1.16075
4	10.20030	0.08119	-2.11440	-0.27095	1.42011	1.13073
5	9.47936	0.07997	-2.23744	-0.34901	2.16753	1.16101

TRANSITION AT X =   0.05049 FOR STEP NUMBER   7

S	V	BUDS	HMEAN	DELTAS	THETAS	TAW	CFI
0.0	175.09175	93.11162	2.55400	0.0	0.0	5000.00000	1000.00000
0.37044	206.72560	75.76796	2.41216	0.00075	0.00001	0.07393	0.00165
0.73449	220.20360	20.26503	2.46594	0.00096	0.00002	0.06011	0.00151
1.42011	222.30049	-21.96093	1.45000	0.00096	0.00003	0.02000	0.00044
2.16752	220.41102	98.24440	1.64068	0.00317	0.00006	0.13944	0.00303

Figure F-5. Continued.

LINE PASSING THROUGH QUADRILATERAL   4

I	X	Y	Z	CP	SL	UABS
1	11.54038	0.25242	-1.54696	0.20013	0.0	0.84435
2	11.19693	0.30250	-1.73560	-0.09346	0.38588	1.04369
3	10.87431	0.29672	-1.90791	-0.30112	0.75167	1.14067
4	10.20979	0.29440	-2.11429	-0.27854	1.44751	1.13072
5	9.47974	0.29589	-2.23569	-0.36567	2.18758	1.16862

TRANSITION AT X = 0.05928 FOR STEP NUMBER 7

S	V	DUDS	HMEAN	DELTAS	THETAS	TAU	CFT
0.0	175.82973	81.47296	2.55400	0.0	0.0	5000.00000	1000.00000
0.38588	205.58162	68.35936	2.41685	0.00076	0.00001	0.07183	0.00196
0.75167	224.25491	29.85786	2.45881	0.00098	0.00002	0.05803	0.00126
1.44751	222.29994	-14.44576	1.45000	0.00093	0.00003	0.02455	0.00093
2.18758	229.74976	50.28996	1.64666	0.00313	0.00000	0.14129	0.00307

.		.				.	
.		.				.	
.		.				.	

12.21214	169.03409	-24.68887	1.41431	0.04772	0.00138	0.05536	0.00128
12.99302	165.91722	10.80241	1.36530	0.04896	0.00150	0.05662	0.00123
14.03721	182.20955	13.13150	1.78622	0.03492	0.00118	0.08041	0.00175
15.33162	189.83224	3.41562	1.75374	0.03368	0.00110	0.09251	0.00201
16.74883	194.95903	1.55478	1.23158	0.03258	0.00108	0.10044	0.00219
18.23149	193.90907	-0.70904	1.22348	0.03509	0.00117	0.09881	0.00215
18.69722	193.79808	-0.04711	1.22072	0.03574	0.00120	0.09859	0.00215
19.73271	193.64034	-0.29937	1.21549	0.03716	0.00125	0.09811	0.00213
20.57314	193.26279	-0.72820	1.21278	0.03848	0.00130	0.09729	0.00212
21.46687	192.20120	-1.71849	1.71034	0.04033	0.00136	0.09541	0.00208

LINE PASSING THROUGH QUADRILATERAL 559

I	X	Y	Z	CP	SL	UABS
1	11.54265	0.03419	0.78633	0.01125	0.0	0.99436
2	10.87562	0.03990	0.90574	-0.27022	0.67761	1.12704
3	10.20865	0.04642	1.00975	0.11576	1.35266	0.94034
4	9.47910	0.05386	1.08324	0.30982	2.08594	0.83077
5	8.87325	0.06325	1.12966	0.51722	2.69360	0.69462
6	8.31187	0.08404	1.59C76	0.40139	3.42033	0.77370
7	7.70868	0.09246	1.96921	-0.19974	4.13246	1.09533
8	7.12469	0.09784	2.21612	-0.58968	4.76632	1.28083
9	6.56250	0.09979	2.36462	-0.66089	5.34799	1.28876
10	5.95850	0.10092	2.42710	-0.75192	5.95921	1.32345
11	5.37551	0.10111	2.42705	0.57915	6.53821	1.25664
12	4.79149	0.10061	2.34389	-0.33948	7.12812	1.15736
13	4.22990	0.09914	2.23961	-0.24647	7.69970	1.11644
14	3.66650	0.09722	2.17710	-0.24149	8.26616	1.11422
15	3.08350	0.09471	2.05210	-0.13395	8.86242	1.06487
16	2.50052	0.09175	1.92696	-0.08335	9.45869	1.04084
17	1.83348	0.08877	1.84390	-0.15435	10.13089	1.07441
18	1.16648	0.08840	1.67550	-0.15653	10.81081	1.07342
19	0.52030	0.08556	1.48017	0.00985	11.49153	0.99506
20	-0.14472	0.07653	1.23150	0.26054	12.20452	0.85992
21	-0.87460	0.05906	0.94854	0.29028	12.98756	0.84245
22	-1.91642	0.05208	0.88189	0.14268	14.03152	0.92992
23	-3.20849	0.04844	0.80107	0.06617	15.32612	0.96931
24	-4.62459	0.04673	0.74135	0.01688	16.74347	0.99153
25	-6.10451	0.04881	0.65489	0.07755	18.22591	0.98613
26	-7.60452	0.05782	0.61242	0.02848	19.72655	0.98566
27	-8.43732	0.06591	0.56108	0.03304	20.56097	0.98334
28	-9.33394	0.06329	0.50635	0.04475	21.45886	0.97762

NEGATIVE VALUE OF TAWST = -0.61835994E-02 AT OR AFTER STATION   2 ON THE STREAMLINE. FOR MORE INFORMATION ON THE LOCATION
OF LAMINAR SEPARATION USE IWRITE = 0 PRINT OPTION. THE TURBULENT ROUTINE IS CALLED AT THE SEPARATION POINT.

S	V	DUDS	HMEAN	DELTAS	THETAS	TAU	CFT
0.0	195.49095	65.61627	2.55400	0.0	0.0	5000.00000	1000.00000
0.67761	221.57565	15.74668	2.62353	0.00116	0.00002	0.04004	0.00087
1.35266	184.87061	-50.57137	1.64868	0.00291	0.00007	0.09873	0.00215
2.08493	163.32964	-37.20828	1.44820	0.00602	0.00017	0.08674	0.00189
2.69360	136.60429	-26.98991	1.54632	0.01300	0.00034	0.04921	0.00098
3.42033	152.10971	69.25374	1.29407	0.00903	0.00029	0.08480	0.00185
4.13246	215.36378	81.32814	1.25196	0.00402	0.00013	0.20391	0.00444
4.76632	247.07802	22.02628	1.27178	0.00382	0.00012	0.25658	0.00558
5.34799	253.34549	10.52018	1.27847	0.00478	0.00015	0.24877	0.00541
5.95921	260.22705	-2.75351	1.27622	0.00558	0.00018	0.25276	0.00544
6.53820	247.06973	-34.51620	1.29871	0.00789	0.00025	0.20260	0.00441
7.12812	227.54654	-25.96162	1.31625	0.01151	0.00036	0.15305	0.00337
7.69969	219.49673	-2.70016	1.29946	0.01374	0.00043	0.14213	0.00309
8.26616	219.05988	-7.66846	1.28346	0.01465	0.00047	0.14227	0.00310
8.86241	209.33728	-16.94519	1.29281	0.01802	0.00057	0.12204	0.00267
9.45868	204.62071	2.85397	1.27913	0.02012	0.00064	0.11680	0.00254
10.13088	211.25720	9.61164	1.75162	0.01893	0.00062	0.13022	0.00283
10.81880	211.42932	-10.79190	1.25007	0.01999	0.00065	0.12878	0.00280
11.49152	195.64638	-36.30284	1.29572	0.02742	0.00087	0.09726	0.00212
12.20451	169.00110	-25.46275	1.41969	0.04815	0.00139	0.05481	0.00119
12.98756	165.59624	10.76632	1.37049	0.05067	0.00151	0.05583	0.00121
14.03152	182.09996	13.31627	1.28881	0.03724	0.00118	0.07983	0.00174
15.32611	189.77525	3.42514	1.25493	0.03388	0.00111	0.09211	0.00200
16.74341	194.93204	1.62446	1.23261	0.03273	0.00109	0.10035	0.00218
18.22583	193.86407	-0.96472	1.22453	0.03525	0.00118	0.09851	0.00214
19.72647	193.78215	-0.10278	1.21583	0.03720	0.00125	0.09816	0.00214
20.56088	193.32541	-0.88039	1.21277	0.03856	0.00130	0.09722	0.00212
21.45877	192.19989	-1.75161	1.21080	0.04046	0.00137	0.09527	0.00207

Figure F-5. Continued.

SUMMARY OF BOUNDARY LAYER INFORMATION FOR QUADRILATERALS

NQUAD	X	Y	Z	DSTAR	SKIN
1	11.76403	0.06110	-1.22013	0.00026	0.00055
2	11.20092	0.10202	-1.77742	0.00073	0.00165
3	11.76403	0.18053	-1.18403	0.00025	0.00052
4	11.19693	0.30250	-1.73960	0.00076	0.00156
5	11.76403	0.29097	-1.13263	0.00025	0.00052
6	11.19651	0.49908	-1.67816	0.00075	0.00155
7	11.76403	0.39513	-1.07360	0.00026	0.00049
8	11.19323	0.68437	-1.58809	0.00078	0.00147
9	11.76403	0.49650	-1.00973	0.00026	0.00048
10	11.19487	0.85845	-1.47752	0.00078	0.00143
.	.	.	.		
.	.	.	.		
.	.	.	.		
550	-10.29183	0.30070	0.17087	0.03802	0.00211
551	-8.42807	0.48947	0.36301	0.03871	0.00210
552	-10.29183	0.27360	0.72783	0.03871	0.00210
553	-8.43346	0.41137	0.43511	0.03854	0.00211
554	-10.29183	0.23540	0.27503	0.03854	0.00211
555	-8.43734	0.31905	0.49572	0.03847	0.00212
556	-10.29183	0.18400	0.31460	0.03847	0.00212
557	-8.44224	0.19827	0.53526	0.03848	0.00212
558	-10.29183	0.11667	0.34933	0.03848	0.00212
559	-8.43732	0.06591	0.56108	0.03854	0.00212
560	-10.29183	0.03890	0.36877	0.03856	0.00212

FRICTION DRAG COEFFICIENT

```
**************************
     FRICTION CD =      0.00756
    REFERENCE AREA =   174.00000
  REYNOLDS NUMBER =   0.3000E 08
      BODY LENGTH =    24.41699
**************************
```

BEGIN WAKE BODY GEOMETRY

AVERAGE SLOPE = -0.03339   AVERAGE PANEL AREA = 0.17566   AVERAGE X-CENTROID = -8.42821   END OF BODY AT X = -19.24188

SECTION 1000

N	X1	X2	X3	X4	XP	XN	A	CZ4
N	Y1	Y2	Y3	Y4	YP	YN	FL	CZ5
P	Z1	Z2	Z3	Z4	ZP	ZN	CZ1	CZ6

WARNING LONG THIN QUAD.
1	-0.76045E 01	-0.76045E 01	-0.91174E 01	-0.91174E 01	-0.83361E 01	-0.70899E-01	0.18459E 00	0.52359E-02
1	0.0	0.13330E 00	0.10938E 00	0.22200E-05	0.60867E-01	0.78298E-01	0.76098E 00	-0.50199E 00
521	-0.10521E 01	-0.10416E 01	-0.93676E 00	-0.94537E 00	-0.99569E 00	-0.99441E 00	-0.11354E-02	0.80929E-02

WARNING LONG THIN QUAD.
1	-0.91174E 01	-0.91174E 01	-0.10979E 02	-0.10979E 02	-0.10016E 02	-0.53364E-01	0.18532E 00	-0.79198E 00
2	0.22200E-05	0.10938E 00	0.88790E-01	0.18021E-05	0.49722E-01	0.78351E-01	0.93367E 00	0.53019E-02
522	-0.94537E 00	-0.93676E 00	-0.84003E 00	-0.84702E 00	-0.89398E 00	-0.99550E 00	-0.26119E-02	0.22182E-02

WARNING LONG THIN QUAD.
2	-0.76045E 01	-0.76045E 01	-0.91174E 01	-0.91174E 01	-0.83365E 01	-0.74160E-01	0.18422E 00	-0.50261E 00
1	0.13330E 00	0.26460E 00	0.21607E 00	0.10938E 00	0.18129E 00	0.20120E 00	0.76150E 00	0.59465E-02
523	-0.10416E 01	-0.10187E 01	-0.91115E 00	-0.93676E 00	-0.97815E 00	-0.97674E 00	-0.10822E-02	0.48250E-01

WARNING LONG THIN QUAD.
2	-0.91174E 01	-0.91174E 01	-0.10979E 02	-0.10979E 02	-0.10016E 02	-0.53859E-01	0.18532E 00	0.61794E-02
2	0.10938E 00	0.21607E 00	0.17539E 00	0.88790E-01	0.14794E 00	0.23311E 00	0.93304E 00	-0.79200E 00
524	-0.93676E 00	-0.91115E 00	-0.81924E 00	-0.84003E 00	-0.87843E 00	-0.97094E 00	-0.33939E-02	-0.87901E-02

WARNING LONG THIN QUAD.
3	-0.76045E 01	-0.76045E 01	-0.91174E 01	-0.91174E 01	-0.83404E 01	-0.79679E-01	0.18126E 00	0.76156E-02
1	0.26460E 00	0.38540E 00	0.31743E 00	0.21607E 00	0.29678E 00	0.36711E 00	0.76178E 00	-0.67257E 00
525	-0.10187E 01	-0.97290E 00	-0.86916E 00	-0.91115E 00	-0.94468E 00	-0.92676E 00	-0.21593E-02	-0.11041E-01

. . . .
. . . .
. . . .

WARNING LONG THIN QUAD.
19	-0.10979E 02	-0.10979E 02	-0.13307E 02	-0.13307E 02	-0.12070E 02	-0.87142E-01	0.17827E 00	0.19366E-03
3	0.17539E 00	0.88789E-01	0.63354E-01	0.12515E 00	0.11422E 00	0.23256E 00	0.11689E 01	-0.10119E 01
597	0.26036E 00	0.28115E 00	0.78961E-01	0.64126E-01	0.17670E 00	0.96867E 00	0.12381E-02	0.11056E-01

WARNING LONG THIN QUAD.
19	-0.13307E 02	-0.13307E 02	-0.19242E 02	-0.19242E 02	-0.15285E 02	-0.85853E-01	0.18927E 00	-0.12408E 01
4	0.12515E 00	0.63354E-01	0.0	0.0	0.62833E-01	0.19770E 00	0.29785E 01	0.13478E-03
598	0.64126E-01	0.78961E-01	-0.42710E 00	-0.42710E 00	-0.94671E-01	0.97650E 00	-0.98737E-03	0.28199E-01

WARNING LONG THIN QUAD.
20	-0.10979E 02	-0.10979E 02	-0.13307E 02	-0.13307E 02	-0.12070E 02	-0.87576E-01	0.17828E 00	-0.10120E 01
3	0.88789E-01	0.12608E-05	0.89964E-06	0.63354E-01	0.38391E-01	0.78160E-01	0.11689E 01	0.22185E-03
599	0.28115E 00	0.28814E 00	0.83944E-01	0.78961E-01	0.18071E 00	0.99309E 00	0.12382E-02	0.20276E-02

WARNING LONG THIN QUAD.
20	-0.13307E 02	-0.13307E 02	-0.19242E 02	-0.19242E 02	-0.15285E 02	-0.86089E-01	0.18928E 00	0.29073E-03
4	0.63354E-01	0.89964E-06	0.0	0.0	0.21118E-01	0.11374E 00	0.29785E 01	-0.12409E 01
600	0.78961E-01	0.83944E-01	-0.42710E 00	-0.42710E 00	-0.88064E-01	0.98977E 00	-0.98730E-03	0.45077E-02

PROBABLE ERROR IN INPUT - SOLID ANGLE =   25.145

Figure F-5. Continued.

POTENTIAL FLOW PROGRAM SECTION 2

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY

POTENTIAL FLOW PROGRAM SECTION 3

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY

X VELOCITY=-1.0   Y VELOCITY= 0.0   Z VELOCITY= 0.0

              ITERATIVE MATRIX SOLUTION INFORMATION

ITERATION	SUM OF CHANGES	A	B1	B2
1	0.49780E 01			
2	0.32203E 01			
3	0.26497E 01			
4	0.22521E 01			
5	0.19827E 01			
		0.532E 00	0.420E 00	0.593E 00
6	0.17985E 01			
7	0.16792E 01			
8	0.16119E 01			
9	0.15901E 01			
10	0.16120E 01			
		0.497E 00	0.501E 00	0.259E 00
11	0.16777E 01			
12	0.17838E 01			
13	0.19420E 01			
14	0.21290E 01			
15	0.23450E 01			
		0.476E 00	0.500E 00	0.231E 00
16	0.25922E 01			
17	0.28788E 01			
18	0.32095E 01			
19	0.35924E 01			
20	0.40360E 01			

A EXTRAPOLATION
              0.471E 00   0.500E 00   0.232E 00

21	0.80136E-01
22	0.84754E-01
23	0.90062E-01
24	0.96259E-01
25	0.10577E 00

A EXTRAPOLATION
              0.475E 00   0.500E 00   0.233E 00

| 26 | 0.47522E-02 |

POTENTIAL FLOW PROGRAM SECTION 4

BEST CESSNA 182 WITH M=21 AND N=29 YIELDING 560 PANELS -- FUSELAGE ONLY   PAGE =   1

X FLOW

PT.	X	Y	Z	VX	VY	VZ	ABS.V	CP	SOURCE	V NORMAL
1	11.76403	0.06110	-1.22013	-0.44360	0.16320	-0.48693	0.67032	0.53988	0.00433	0.378-05
2	11.20092	0.10202	-1.77762	-0.96555	0.00935	-0.44131	1.06166	-0.12713	0.03079	0.000-05
3	11.76403	0.18093	-1.18403	-0.44189	0.23780	-0.45002	0.67499	0.53767	0.00814	0.368-05
4	11.19693	0.30250	-1.73560	-0.92927	0.06341	-0.47526	1.04587	-0.09943	0.04249	0.380-05
5	11.76403	0.29097	-1.13263	-0.43396	0.23214	-0.46903	0.65027	0.56660	0.03599	0.500-05
6	11.19651	0.49908	-1.67816	-0.86843	0.14254	-0.48651	1.02195	-0.04438	0.04704	-0.246-05
7	11.76403	0.39513	-1.07360	-0.39598	0.32563	-0.46310	0.65001	0.57749	0.10534	0.608-05
8	11.19325	0.68437	-1.58009	-0.85930	0.23574	-0.49912	1.00238	-0.00476	0.05457	-0.296-05
9	11.76403	0.49650	-1.00973	-0.40160	0.28109	-0.43244	0.65403	0.57225	0.10968	0.698-05
10	11.19487	0.89845	-1.47752	-0.85442	0.33212	-0.39105	0.99462	0.00675	0.09464	-0.418-05
⋮										
590	-15.28517	0.20467	-0.10159	-0.97906	-0.06736	-0.04448	0.98230	0.03493	-0.00676	-0.098-05
591	-12.07021	0.31777	0.07299	-0.97743	-0.00127	-0.05996	0.98093	0.03779	-0.00744	-0.248-05
592	-15.28517	0.17400	-0.19172	-0.97875	-0.06090	-0.05406	0.98211	0.03946	-0.00675	-0.648-03
593	-12.07021	0.29566	0.11012	-0.97702	-0.09090	-0.06600	0.98063	0.03837	-0.00769	-0.248-05
594	-15.28517	0.14063	-0.12690	-0.97820	-0.04888	-0.06776	0.98176	0.03614	-0.00720	-0.698-05
595	-12.07021	0.18729	0.15297	-0.97670	-0.05819	-0.07600	0.98079	0.03883	-0.00700	-0.248-05
596	-15.28517	0.10300	-0.10772	-0.97800	-0.03090	-0.07443	0.98150	0.03690	-0.00720	-0.698-05
597	-12.07021	0.11422	0.17670	-0.97647	-0.02367	-0.08241	0.98023	0.03913	-0.00801	-0.348-05
598	-15.28517	0.06283	-0.00467	-0.97770	-0.02182	-0.08221	0.98139	0.03607	-0.00744	-0.698-05
599	-12.07021	0.03939	0.18071	-0.97636	-0.00001	-0.08971	0.98015	0.03931	-0.00807	-0.348-05
600	-15.28517	0.02112	-0.08806	-0.97763	-0.00929	-0.08463	0.98133	0.03700	-0.00744	-0.698-05

PRESSURE LIFT AND DRAG COEFFICIENTS
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
          PRESSURE CL =   0.00444
          PRESSURE CD =   0.00491
       REFERENCE AREA =   174.00000
      REYNOLDS NUMBER =   0.30000E 00
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

                    TOTAL BODY COEFFICIENTS

●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●
       TOTAL BODY CL =   0.00444
       TOTAL BODY CD =   0.01249
       REFERENCE AREA =   174.00000
      REYNOLDS NUMBER =   0.30000E 00
          BODY LENGTH =   24.41699
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

Figure F-5. Continued.

404

APPENDIX G- Theoretical Basis of Oeller's Method

If an inviscid two-dimensional flow is everywhere parallel to the surface of a closed body, then the surface of the body can be represented by a streamline on which the stream function, ψ, is constant. Oeller (Ref. 26) used this fact to develop a method for obtaining the potential flow about an airfoil. He replaced the airfoil surface by a vortex sheet and required that the sum of the stream function for a uniform stream and the stream function for the vortex sheet be a constant on the airfoil surface. This requirement is represented by the integral equation

$$\psi = z(s)\, V_\infty \cos\alpha - x(s)\, V_\infty \sin\alpha - \frac{1}{2\pi}\oint \gamma(s')\, \ell n\left[r(s,s')\right]ds' \qquad \text{(G1)}$$

where ψ is the unknown constant stream function value on the airfoil surface, V_∞ is the free stream velocity, α is the angle between the free stream and the x-axis of the reference system, $\gamma(s')$ is the vorticity strength at any point s' on the surface, $r(s,s')$ is the distance between points s and s', and $x(s)$ and $z(s)$ are the coordinates of the point of interest s. s and s' are arc lengths measured along the airfoil surface starting from the trailing edge. s is any fixed point on the airfoil surface while s' is the integration variable point which moves from the trailing edge over the airfoil surface back to the trailing edge. (See Figure G-1.)
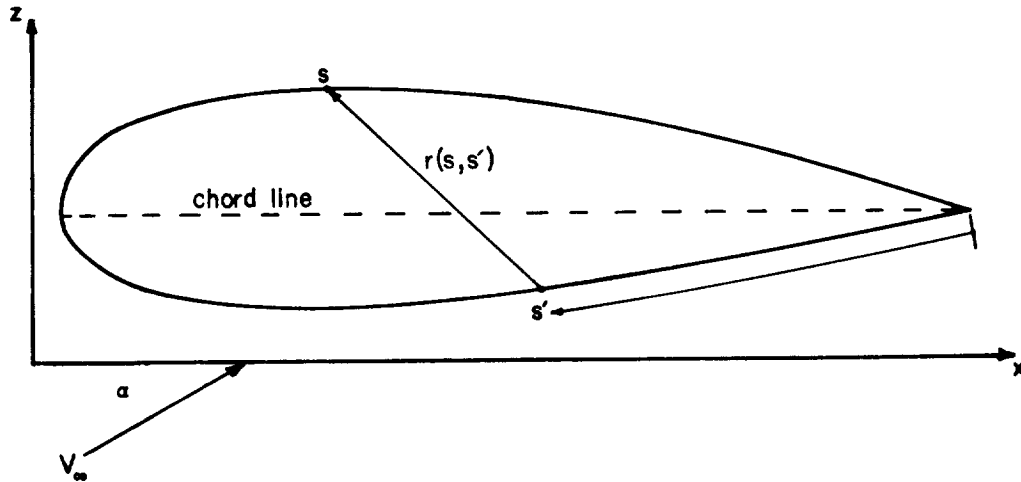
Figure G-1. Geometry for potential flow calculation.

Note that the reference system is chosen with its x-axis parallel to the airfoil chord line so that α will be the angle of attack of the airfoil.

To solve Equation (G1) for ψ and $\gamma(s')$, the integral is approximated by a summation. The airfoil is divided into N segments, $\gamma(s')$ is assumed constant on each segment, and Equation (G1) is applied at the mid-point of

each segment to yield the following system of simultaneous linear equations (see Figure G-2).

$$\psi = z_{c_i} \, V_\infty \cos \alpha - x_{c_i} \, V_\infty \sin \alpha - \sum_{j=1}^{N} K_{ij} \, \gamma_j \qquad \text{(G2)}$$

for $i = 1, 2, \ldots N$

where

$$K_{ij} = \frac{1}{2\pi} \int_{s_j}^{s_{j+1}} \ln \left[r(s_{c_i}, s') \right] ds'$$

and

$$x_{c_i} = (x_{i+1} + x_i)/2$$

$$z_{c_i} = (z_{i+1} + z_i)/2$$

are the coordinates of the midpoint of the ith segment. This point is called control point s_{c_i}, and it is the point where the requirement that ψ have a constant value is enforced.
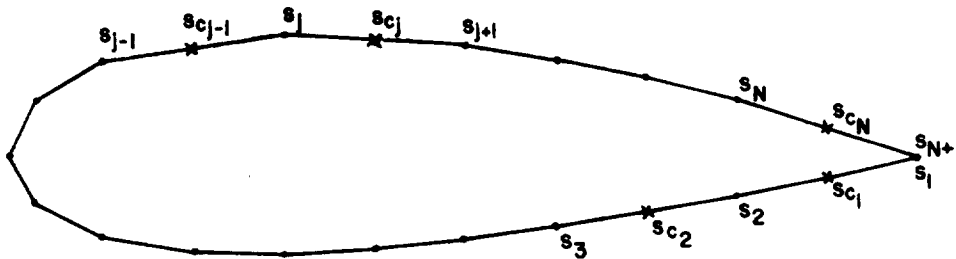
Figure G-2. Airfoil approximation by polygon.

K_{ij} represents the influence coefficient for the effect of the vorticity of line segment j at the control point i. The influence of all of the line segments at control point i is obtained by summing over j as is done in Equation (G2). To obtain the required expressions for the K_{ij} we must evaluate the integrals

$$\int_{s_j}^{s_{j+1}} \ln \left[r(s_{c_i}, s') \right] ds' \quad .$$

First consider the case where $i \neq j$, $i.e.$ the ith control point (x_{c_i}, z_{c_i}) does not lie on the jth segment of the airfoil as shown in the figure on the following page.

406
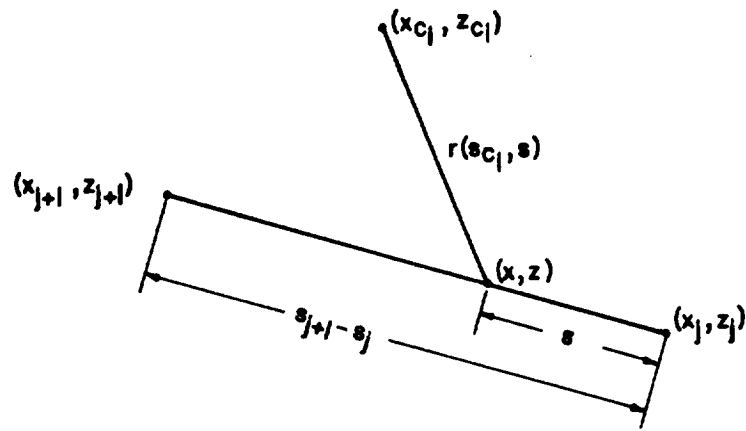
Figure G-3. Geometry for calculation of K_{ij}, $i \neq j$.

Now with a change in variables from s' to s ($s = s' - s_j$),

$$\int_{s_j}^{s_{j+1}} \ln \left[r(s_{c_i}, s') \right] ds' = \int_0^{s_{j+1} - s_j} \ln \left[r(s_{c_i}, s) \right] ds. \qquad (G3)$$

In the above figure, x and z are the coordinates of the moving integration variable point s which moves from point (x_j, z_j) to point (x_{j+1}, z_{j+1}) as s changes from 0 to $s_{j+1} - s_j$. Since we are considering a straight line segment

$$x = x_j + \frac{(x_{j+1} - x_j)}{(s_{j+1} - s_j)} s$$

$$z = z_j + \frac{(z_{j+1} - z_j)}{(s_{j+1} - s_j)} s$$

$$(G4)$$

$$r^2 = (x - x_{c_i})^2 + (z - z_{c_i})^2$$

$$= s^2 + \frac{2}{(s_{j+1} - s_j)} \left[(x_j - x_{c_i})(x_{j+1} - x_j) + (z_j - z_{c_i})(z_{j+1} - z_j) \right] s$$

$$+ (x_j - x_{c_i})^2 + (z_j - z_{c_i})^2$$

$$= s^2 + bs + a$$

for $0 \leq s \leq s_{j+1} - s_j$.

407

Now

$$\int \ln\left[r(s_{c_i},s)\right]ds = \int \ln\left\{\left[r^2(s_{c_i},s)\right]^{1/2}\right\}ds$$

$$= \tfrac{1}{2}\int \ln\left[r^2(s_{c_i},s)\right]ds \; ; \qquad\qquad (G5)$$

therefore

$$\int_{s_j}^{s_{j+1}} \ln\left[r(s_{c_i},s')\right]ds' = \tfrac{1}{2}\int_0^{s_{j+1}-s_j} \ln\left[r^2(s_{c_i},s)\right]ds$$

$$= \tfrac{1}{2}\int_0^{s_{j+1}-s_j} \ln\left[s^2 + bs + a\right]ds$$

$$= \tfrac{1}{2}\left[\left(s + \frac{b}{2}\right)\ln(s^2 + bs + a) - 2s \right.$$

$$\left. + \sqrt{4a - b^2}\,\tan^{-1}\left(\frac{2s + b}{\sqrt{4a - b^2}}\right)\right]_0^{s_{j+1}-s_j} .$$

$$(G6)$$

The integral has this value provided $4a - b^2 > 0$. That this is always true for this representation of airfoils can be shown by substituting the expressions for a and for b from Equation (G3) into $4a - b^2$ to obtain

$$4a - b^2 = 4\left[\frac{(x_j - x_{c_i})(z_{j+1} - z_j) - (z_j - z_{c_i})(x_{j+1} - x_j)}{(s_{j+1} - s_j)}\right]^2$$

$$> 0 .$$

Also

$$\sqrt{4a - b^2} = 2\left|\frac{(x_j - x_{c_i})(z_{j+1} - z_j) - (z_j - z_{c_i})(x_{j+1} - x_j)}{(s_{j+1} - s_j)}\right| \qquad (G7)$$

$$= 2C$$

where the absolute value signs are used to insure that the <u>positive</u> square root is obtained.

To obtain a final expression for Equation (G6) we must evaluate the terms $s + b/2$ and $s^2 + bs + a$ for $s = 0$ and for $s = s_{j+1} - s_j$, corresponding to the two ends of the jth line segment. For $s = s_{j+1} - s_j = \Delta s$,

$$s^2 + bs + a = (x_{j+1} - x_{c_i})^2 + (z_{j+1} - z_{c_i})^2$$

$$= R2 \qquad\qquad (G8)$$

$$s + b/2 = \left[(x_{j+1} - x_{c_i})(x_{j+1} - x_j) + (z_{j+1} - z_{c_i})(z_{j+1} - z_j)\right]/\Delta s$$

$$= T2 .$$

408

For s = 0

$$s^2 + bs + a = (x_j - x_{c_i})^2 + (z_j - z_{c_i})^2$$

$$= R1$$

$$s + b/2 = \left[(x_j - x_{c_i})(x_{j+1} - x_j) + (z_j - z_{c_i})(z_{j+1} - z_j)\right]/\Delta s \qquad (G9)$$

$$= T1 \quad .$$

Substituting the above expressions into Equation (G6) and Equation (G2) we obtain

$$K_{ij} = \frac{1}{4\pi}\left[T2 \ln (R2) - T1 \ln (R1)\right] - \frac{\Delta s}{2\pi} + \frac{C}{2\pi}\left[\tan^{-1}\left(\frac{T2}{C}\right) - \tan^{-1}\left(\frac{T1}{C}\right)\right] (G10)$$

for $i \neq j$.

Now consider the case where $i = j$, *i.e.* the ith control point lies on the jth segment as shown in the figure below.
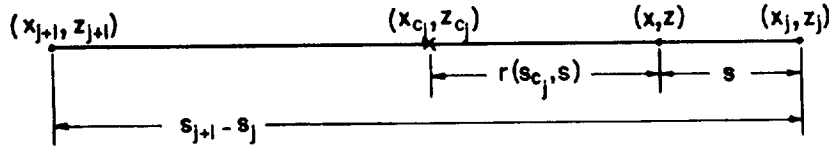
Figure G-4. Geometry for calculation of K_{jj}.

Here we have (letting $\Delta s = s_{j+1} - s_j$)

$$r(s_{c_j},s) = \frac{\Delta s}{2} - s \qquad \text{for} \qquad 0 \leq s \leq \frac{\Delta s}{2}$$

and

$$r(s_{c_j},s) = s - \frac{\Delta s}{2} \qquad \text{for} \qquad \frac{\Delta s}{2} \leq s \leq \Delta s \quad .$$

Thus for Equation (G3) we have

$$\int_0^{s_{j+1}-s_j} \ln\left[r(s_{c_j},s)\right]ds = \int_0^{\Delta s/2} \ln\left[\frac{\Delta s}{2} - s\right]ds + \int_{\Delta s/2}^{\Delta s} \ln\left[s - \frac{\Delta s}{2}\right]ds$$

$$= \Delta s\left[\ln\left(\frac{\Delta s}{2}\right) - 1\right] \qquad (G11)$$

and therefore

$$K_{jj} = \frac{\Delta s}{2\pi}\left[\ln\left(\frac{\Delta s}{2}\right) - 1\right] . \qquad (G12)$$

409

Equation (G2) represents a system of N equations in N + 1 unknowns, *i.e.* γ_1, γ_2,...γ_N and ψ. The additional equation needed to close the system is obtained from the Kutta condition which is represented by

$$\gamma_N \, (s_{N+1} - s_N) = - \gamma_1 \, (s_2 - s_1) \, . \tag{G13}$$

Although Equation (G13) appears to be a rather odd way to express the Kutta condition, the discussion below will show that it is indeed correct.

The solution of the system of Equations (G2) plus Equation (G13) gives a set of γ_j's for the N segments of the airfoil. These vorticity strengths, γ_j's, are the same as the actual local tangential velocities at the mid-chord control points. Because only the body points are stored in the program and all calculations are referred to these body points, one observes (refer to Figure G-2) that what is actually desired are the local tangential velocities at the <u>body</u> points which are the <u>end</u> points of the line segments. Therefore, some form of an averaging procedure is needed to obtain the vorticity at a body point from the vorticities on the two line segments that join at the body point. .Letting $\overline{\gamma}_j$ denote the vorticity (and thus the local tangential velocity) at body point s_j, the correct averaging procedure is

$$
\begin{aligned}
\overline{\gamma}_j &= \left[\gamma_j \, (s_{c_j} - s_j) + \gamma_{j-1} \, (s_j - s_{c_{j-1}}) \right] / (s_{c_j} - s_{c_{j-1}}) \\
&= \left[\gamma_j \, (s_{j+1} - s_j) + \gamma_{j-1} \, (s_j - s_{j-1}) \right] / (s_{j+1} - s_{j-1}) .
\end{aligned}
\tag{G14}
$$

Since γ times a line segment length is the circulation due to a vortex sheet of that length, Equation (G14) is equivalent to requiring the circulation due to $\overline{\gamma}_j$ on the line segment $s_{c_j} - s_{c_{j-1}}$ be the same as the circulation due to γ_{j-1} on line segment $s_j - s_{c_{j-1}}$ plus the circulation due to γ_j on line segment $s_{c_j} - s_j$.

Now consider the Kutta condition represented in Equation (G13). Substituting Equation (G13) into Equation (G14) gives

$$\overline{\gamma}_1 = \overline{\gamma}_{N+1} = 0$$

and thus there is a stagnation point at the trailing edge which is the Kutta condition.

As discussed on page 122 , the program actually uses a modified Kutta condition which states that the upper and lower surface velocities at the trailing edge are equal but not necessarily zero. For this case, the trailing edge velocities $\overline{\gamma}_1$ and $\overline{\gamma}_{N+1}$ are calculated from

$$\overline{\gamma}_1 = \overline{\gamma}_{N+1} = \frac{|\gamma_1|(s_2 - s_1) + |\gamma_N|(s_{N+1} - s_N)}{(s_2 - s_1 + s_{N+1} - s_N)} \, . \tag{G16}$$

All other body point velocities are calculated using Equation (G14).

410

APPENDIX H-Rapid, Inviscid Computation of the Pressure Distribution of

Symmetrical Airfoils for Mach Numbers Less Than or Equal To I.O

Prior to undertaking the present study, the senior author had the occasion to estimate the characteristics of some unusual symmetrical airfoils at Mach numbers from zero to unity. These airfoils were reportedly capable of relatively low drag at transonic speeds. It was the intention of the research to test models of these airfoils and to develop fairly simple, yet reasonably accurate, methods for predicting their behavior. The reader will recognize that the development of such methods usually follows one of two paths: either some new bit of physical or mathematical insight is uncovered which permits one to legitimately simplify the formulation of the problem or its method of solution; or one seeks to find or assemble correlations among empirical results. To pursue the first path—certainly the more elegant and distinctive of the two—requires that the researcher be struck by unusual inspiration, an occurrence that is not always within his power to command, at least during a fixed time interval. For this reason many simple, but reasonably accurate, prediction methods are at least semi-empirical.

In reviewing some of the semi-empirical methods given in the literature for predicting the pressure distribution on airfoils at free stream Mach numbers near unity it was found that they require as a starting condition the pressure distribution at $M_{CRITICAL}$. Thus in order for one to investigate the utility of these methods or modifications thereof it would be necessary to have some fairly reliable means of predicting the M_{CR} pressure distribution. The task of mounting the computer program discussed in NASA CR-1843 (Ref. 34) seemed to be more involved than was warranted by the uncertainties of the final result. For this reason it was decided to obtain the pressure distribution by using the 16-point Weber method (Ref. 20) to which had been added a Kármán-Tsien Mach number correction. This method is easily programmed for computer solution because Weber, by fixing the chordwise location of the 16 points (see Table H-1) at which the pressure is computed, was able to determine, once and for all, coefficients by which the airfoil ordinates at these 16 points could be multiplied and the results summed to find the surface pressures and velocities. One merely supplies these coefficients (which are given in Weber's paper and here in Tables H-2 through H-7) as a permanent data set and the ordinates of the airfoil for which the pressures are desired as a changeable data set. It must be understood, however, that this form of Weber's method is restricted to inviscid flow about symmetrical airfoils. Thus it can be expected to give reasonable lift and moment values only for relatively thin airfoils at moderate-to-small angles of attack. No drag values can be obtained.

Because of its simplicity the method permits the calculations to be carried out very rapidly by even the smallest computer. It therefore seems well suited for use by those who would be satisfied to investigate the characteristics of new symmetrical airfoils in a more qualitative fashion, those whose access to larger machines or computing funds is restricted, or those with limited skill or time to mount foreign programs. The accuracy of the method is quite good except in the immediate vicinity of the leading edge as can be seen in Figure H-1.

The computer program for performing the calculations required by the 16-point Weber method was given the name TRINSON. This program provides the pressure data upon which a second program, COMPR, operates to modify the pressure distribution for Mach numbers other than zero. As noted above, below M_{CR}, a Kármán-Tsien correction is used. For $M > M_{CR}$ a series of semi-empirical correlations are used to obtain the pressure distribution over the complete airfoil surface. These are described in more detail in Ref. 103. Essentially, the procedure uses an approximate analytical method (by Truitt, Ref. 104) to locate the shock on the airfoil at $M = 1$. The pressure distribution between the shock and the sonic point at $M = 1.0$ is found from the correlation of Thompson and Wilby (Ref. 106). Sinnotts' (Ref. 105) semi-empirical correlation, which gives the pressure distribution aft of the airfoil peak in terms of the pressure distributions at $M = 1$ and $M = M_{CR}$, is used for intermediate Mach numbers. A spline fit of pressure data at the airfoil peak and that near the sonic point plus the idea that the pressure distribution on blunt bodies in supersonic flow "freezes" (does not change with changes in M_∞) are used to represent the airfoil surface pressures between the leading edge and the airfoil peak. The pressure rise through the shockwave is then "smeared out" following the empirical result presented in Schlichting (Ref. 65) that the pressure rise occurs over a streamwise distance of about 50 boundary layer thicknesses for laminar boundary layers and over about 12 boundary layer thicknesses for turbulent boundary layers.

Comparisons between the predictions obtained through the use of TRINSON and COMPR and experimental data for one airfoil obtained in the NCSU transonic wind tunnel are shown in Figure H-2. Note that qualitatively the agreement is quite good; quantitatively, however, the predictions do not match the experimental results as closely as one might expect for Mach numbers just above the critical. In particular, it is evident that for Mach numbers between M_{CR} (~ 0.77) and $M_\infty = 0.85$ the pressure rise through the shock is even more "smeared out" than that suggested by the 50δ criterion. On the other hand, for $M > 0.85$ this concept seems to give very good results.

On the basis of these results an effort was made to develop a somewhat more accurate prediction by using the 32-point Weber method, given here as TRANSON, and a more accurate representation of the pressure rise through a shock. Experimental data showing pressures during the interaction of very weak normal shocks with laminar boundary layers seem to be quite scarce as are correlations identifying the governing parameters in a useable fashion.

412

As a result, it was not possible to find a model analogous to the 50δ concept which was capable of accurately representing the pressure rise through the shock at all Mach numbers from M_{CR} to M = 1.0. Also, the 32-point Weber method does not appear to give results markedly different from those obtained by the 16-point method except for the first 10 percent of the airfoil chord. Apparently, off-body viscous effects, which are of course inadequately described by any potential flow treatment or, for that matter, even by simple boundary layers added to potential flows, are sufficiently prominent in the experimental data to prevent one from achieving a better prediction using this approach.

Since these computer programs have not been previously made available in any form and since they provide comparatively rapid, at least qualitatively accurate, predictions of the pressure distribution on a limited, but highly useable class of airfoils for all Mach numbers less than or equal to unity, it was felt that inclusion of these programs here may prove helpful to many readers: Those who, on occasion, may not wish to incur the expense of running the 65-point airfoil program and who are thereby willing to accept the inherent limitations of the shorter programs; and those who wish at least a qualitative prediction of airfoil lift and moment characteristics at Mach numbers above the applicable range of the 65-point program. The following discussion provides listings of three programs, instructions for entering data, and typical output.

413
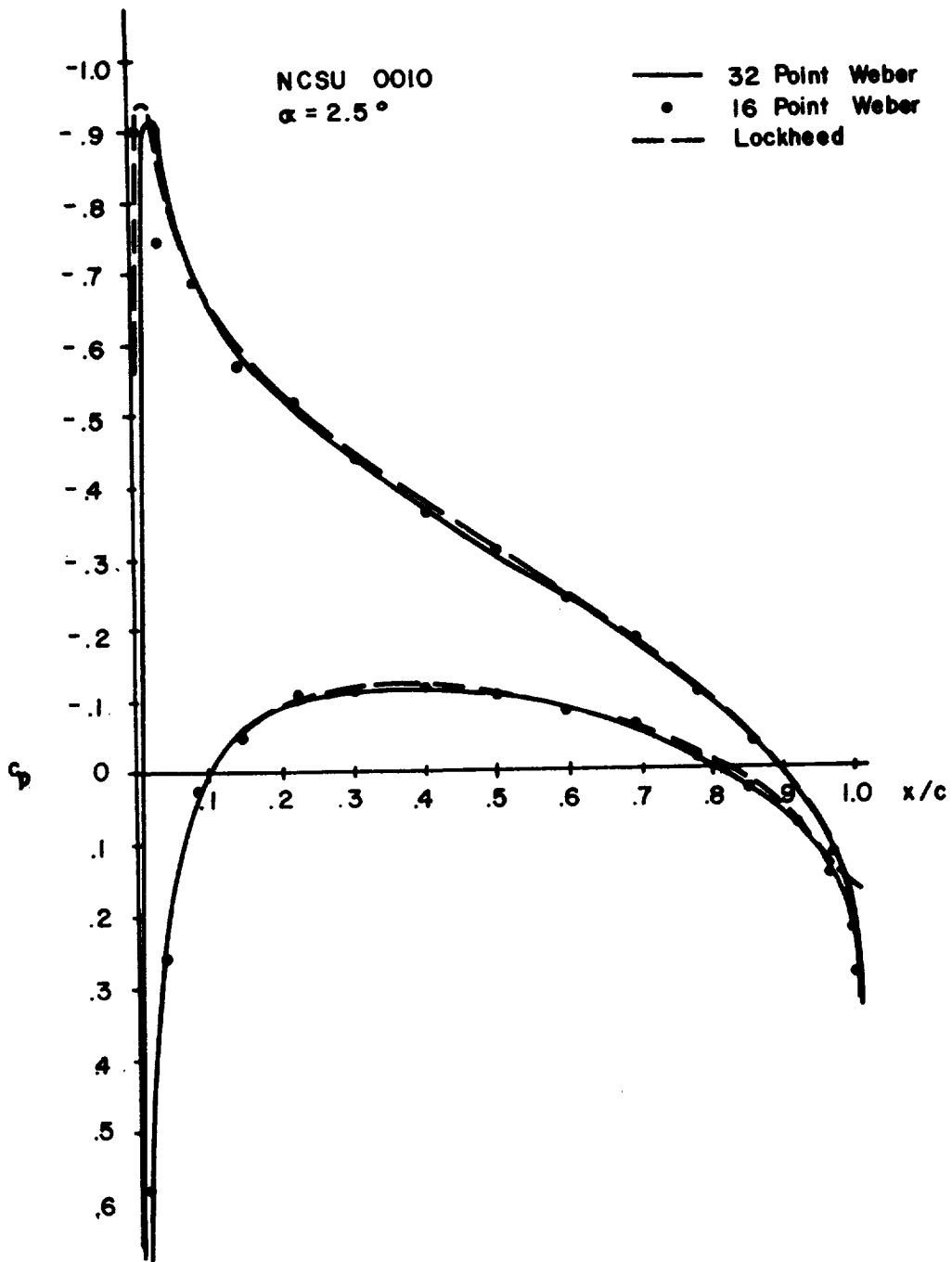
Figure H-1. Comparison of 16 point Weber, 32 point Weber, and 65 point Lockheed methods for predicting airfoil pressure distributions on the NCSU 0010 at α = 2.5°.

414

Position of the Pivotal Points

v	$N=8$ x_v	v	$N=16$ x_v	v	$N=32$ x_v
0	1.000	0	1.0000	0	1.0000
				1	0.9976
1	0.9619	1	0.9904	2	0.9904
				3	0.9785
		2	0.9619	4	0.9619
				5	0.9410
2	0.8536	3	0.9157	6	0.9157
				7	0.8865
		4	0.8536	8	0.8536
				9	0.8172
3	0.6913	5	0.7778	10	0.7778
				11	0.7357
		6	0.6913	12	0.6913
				13	0.6451
4	0.5000	7	0.5975	14	0.5975
				15	0.5490
		8	0.5000	16	0.5000
				17	0.4510
5	0.3087	9	0.4025	18	0.4025
				19	0.3549
		10	0.3087	20	0.3087
				21	0.2643
6	0.1464	11	0.2222	22	0.2222
				23	0.1828
		12	0.1464	24	0.1464
				25	0.1135
7	0.03806	13	0.08427	26	0.08427
				27	0.05904
		14	0.03806	28	0.03806
				29	0.02153
		15	0.00961	30	0.00961
				31	0.00241
8	0	16	0	32	0

Table H-1. Position of the pivotal points.

$s_{\mu\nu}^{(1)}$ for $N = 16$

μ \ ν	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	82·013	-15·061	0	-0·651	0	-0·136	0	-0·051	0	-0·026	0	-0·017	0	-0·013	0	-0·012
2	-29·544	41·810	-11·203	0	-0·705	0	-0·180	0	-0·076	0	-0·044	0	-0·031	0	-0·026	0
3	0	-16·265	28·799	-8·980	0	-0·690	0	-0·201	0	-0·094	0	-0·059	0	-0·045	0	-0·041
4	-2·360	0	-11·430	22·627	-7·698	0	-0·674	0	-0·217	0	-0·111	0	-0·075	0	-0·062	0
5	0	-1·532	0	-9·052	19·243	-6·954	0	-0·673	0	-0·236	0	-0·130	0	-0·095	0	-0·086
6	-0·646	0	-1·147	0	-7·727	17·318	-6·563	0	-0·692	0	-0·262	0	-0·157	0	-0·124	0
7	0	-0·462	0	-0·935	0	-6·968	16·314	-6·442	0	-0·735	0	-0·301	0	-0·196	0	-0·172
8	-0·260	0	-0·362	0	-0·810	0	-6·442	16·000	-6·442	0	-0·810	0	-0·362	0	-0·260	0
9	0	-0·196	0	-0·301	0	-0·735	0	-6·442	16·314	-6·968	0	-0·935	0	-0·462	0	-0·378
10	-0·124	0	-0·157	0	-0·262	0	-0·692	0	-6·563	17·318	-7·727	0	-1·147	0	-0·646	0
11	0	-0·095	0	-0·130	0	-0·236	0	-0·673	0	-6·954	19·243	-9·052	0	-1·532	0	-1·052
12	-0·062	0	-0·075	0	-0·111	0	-0·217	0	-0·674	0	-7·698	22·627	-11·430	0	-2·360	0
13	0	-0·045	0	-0·059	0	-0·094	0	-0·201	0	-0·690	0	-8·980	28·799	-16·265	0	-4·880
14	-0·026	0	-0·031	0	-0·044	0	-0·076	0	-0·180	0	-0·705	0	-11·203	41·810	-29·544	0
15	0	-0·013	0	-0·017	0	-0·026	0	-0·051	0	-0·136	0	-0·651	0	-15·061	82·013	-132·103

Table H-2. Weber coefficients $s^{(1)}$ for the 16 point method.

416

s_μ^m for $N = 16$

μ \ ν	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	25·769	17·917	-4·703	2·016	-1·104	0·706	-0·506	0·398	-0·338	0·310	-0·305	0·327	-0·388	0·535	-1·020
2	-68·941	6·309	14·908	-4·993	2·499	-1·531	1·071	-0·828	0·697	-0·634	0·622	-0·664	0·785	-1·082	2·060
3	-38·144	-31·420	2·694	12·636	-4·844	2·680	-1·780	1·336	-1·104	0·991	-0·963	1·021	-1·203	1·654	-3·143
4	-28·486	17·048	-20·468	1·414	11·224	-4·718	2·816	-2·000	1·598	-1·405	1·347	-1·414	1·654	-2·266	4·295
5	20·046	-11·798	10·849	-15·519	0·804	10·411	-4·704	2·993	-2·259	1·918	-1·800	1·862	-2·158	2·937	-5·548
6	-15·836	8·922	-7·411	8·054	-12·854	0·448	10·043	-4·828	3·261	-2·613	2·369	-2·398	2·739	-3·696	6·946
7	12·797	-7·033	5·548	-5·418	6·546	-11·318	0·203	10·055	-5·126	3·675	-3·143	3·075	-3·439	4·581	-8·551
8	-10·453	5·657	-4·330	4·000	-4·330	5·657	-10·453	0	10·453	-5·657	4·330	-4·000	4·330	-5·657	10·453
9	8·551	-4·581	3·439	-3·075	3·143	-3·675	5·126	-10·055	-0·203	11·318	-6·546	5·418	-5·548	7·033	-12·797
10	-6·946	3·696	-2·739	2·398	-2·369	2·613	-3·261	4·828	-10·043	-0·448	12·854	-8·054	7·411	-8·922	15·836
11	5·548	-2·937	2·158	-1·862	1·800	-1·918	2·259	-2·993	4·704	-10·411	-0·804	15·519	-10·849	11·798	-20·046
12	-4·295	2·266	-1·654	1·414	-1·347	1·405	-1·598	2·000	-2·816	4·718	-11·224	-1·414	20·468	-17·048	26·486
13	3·143	-1·654	1·203	-1·021	0·963	-0·991	1·104	-1·336	1·780	-2·680	4·844	-12·636	-2·694	31·420	-38·144
14	-2·060	1·082	-0·785	0·664	-0·622	0·634	-0·697	0·828	-1·071	1·531	-2·499	4·993	-14·908	-6·309	68·941
15	1·020	-0·535	0·388	-0·327	0·305	-0·310	0·338	-0·398	0·506	-0·706	1·104	-2·016	4·703	-17·917	-25·769

Table H-3. Weber coefficients S(2) for the 16 point method.

$$s_\mu^{(3)} \text{ for } N = 16$$

ν

μ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	82·013	7·458	0	4·031	0	2·006	0	1·256	0	0·914	0	0·742	0	0·659	0	0·635
2	-41·024	41·810	-4·134	0	1·068	0	0·716	0	0·507	0	0·398	0	0·341	0	0·317	0
3	0	-21·134	28·799	-5·362	0	0·313	0	0·340	0	0·276	0	0·234	0	0·211	0	0·204
4	-3·562	0	-14·273	22·627	-5·365	0	0·016	0	0·175	0	0·169	0	0·155	0	0·147	0
5	0	-2·349	0	-11·036	19·243	-5·215	0	-0·132	0	0·084	0	0·108	0	0·108	0	0·107
6	-1·098	0	-1·750	0	-9·293	17·318	-5·121	0	-0·224	0	0·026	0	0·066	0	0·074	0
7	0	-0·811	0	-1·433	0	-8·326	16·314	-5·136	0	-0·293	0	-0·019	0	0·032	0	0·042
8	-0·515	0	-0·662	0	-1·260	0	-7·850	16·000	-5·287	0	-0·360	0	-0·061	0	-0·005	0
9	0	-0·424	0	-0·584	0	-1·176	0	-7·749	16·314	-5·609	0	-0·437	0	-0·112	0	-0·062
10	-0·323	0	-0·380	0	-0·551	0	-1·160	0	-8·006	17·318	-6·162	0	-0·544	0	-0·193	0
11	0	-0·298	0	-0·369	0	-0·557	0	-1·215	0	-8·694	19·243	-7·068	0	-0·716	0	-0·376
12	-0·271	0	-0·305	0	-0·391	0	-0·609	0	-1·365	0	-10·031	22·627	-8·587	0	-1·068	0
13	0	-0·301	0	-0·351	0	-0·465	0	-0·742	0	-1·692	0	-12·599	28·799	-11·395	0	-2·220
14	-0·369	0	-0·403	0	-0·485	0	-0·660	0	-1·076	0	-2·479	0	-18·273	41·810	-18·064	0
15	0	-0·686	0	-0·776	0	-0·966	0	-1·357	0	-2·279	0	-5·334	0	-37·580	82·013	-65·410
16	-0·063	0	-0·068	0	-0·080	0	-0·105	0	-0·155	0	-0·281	0	-0·742	0	-6·505	

Table H-4. Weber coefficients $S^{(3)}$ for the 16 point method.

418

$s_\mu^{(1)}$ for $N = 32$

ν

μ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	326·474	−59·103	0	−2·410	0	−0·457	0	−0·148	0	−0·063	0	−0·033	0	−0·019	0	−0·012
2	−117·636	164·027	−42·889	0	−2·495	0	−0·565	0	−0·203	0	−0·094	0	−0·051	0	−0·031	0
3	0	−63·816	110·237	−33·198	0	−2·305	0	−0·581	0	−0·225	0	−0·110	0	−0·063	0	−0·040
4	−9·409	0	−43·765	83·620	−27·171	0	−2·102	0	−0·571	0	−0·234	0	−0·119	0	−0·070	0
5	0	−6·029	0	−33·470	67·883	−23·149	0	−1·928	0	−0·553	0	−0·236	0	−0·125	0	−0·076
6	−2·591	0	−4·411	0	−27·297	57·599	−20·321	0	−1·788	0	−0·536	0	−0·237	0	−0·129	0
7	0	−1·837	0	−3·484	0	−23·207	50·442	−18·258	0	−1·677	0	−0·526	0	−0·237	0	−0·133
8	−1·065	0	−1·416	0	−2·892	0	−20·350	45·255	−16·717	0	−1·591	0	−0·509	0	−0·238	0
9	0	−0·805	0	−1·153	0	−2·488	0	−18·275	41·397	−15·552	0	−1·525	0	−0·501	0	−0·240
10	−0·538	0	−0·645	0	−0·976	0	−2·198	0	−16·728	38·486	−14·689	0	−1·477	0	−0·496	0
11	0	−0·425	0	−0·538	0	−0·850	0	−1·984	0	−15·560	36·284	−14·008	0	−1·444	0	−0·496
12	−0·308	0	−0·350	0	−0·463	0	−0·758	0	−1·823	0	−14·674	34·637	−13·527	0	−1·425	0
13	0	−0·251	0	−0·298	0	−0·408	0	−0·688	0	−1·700	0	−14·010	33·440	−13·199	0	−1·420
14	−0·192	0	−0·211	0	−0·260	0	−0·367	0	−0·635	0	−1·606	0	−13·529	32·627	−13·010	0
15	0	−0·160	0	−0·182	0	−0·231	0	−0·335	0	−0·594	0	−1·535	0	−13·201	32·155	−12·948
16	−0·126	0	−0·137	0	−0·161	0	−0·209	0	−0·311	0	−0·563	0	−1·483	0	−13·011	32·000
17	0	−0·107	0	−0·119	0	−0·144	0	−0·192	0	−0·291	0	−0·538	0	−1·448	0	−12·948
18	−0·087	0	−0·092	0	−0·106	0	−0·131	0	−0·178	0	−0·276	0	−0·520	0	−1·427	0
19	0	−0·074	0	−0·081	0	−0·095	0	−0·120	0	−0·167	0	−0·264	0	−0·508	0	−1·420
20	−0·061	0	−0·064	0	−0·072	0	−0·086	0	−0·112	0	−0·158	0	−0·255	0	−0·500	0
21	0	−0·052	0	−0·057	0	−0·065	0	−0·079	0	−0·105	0	−0·151	0	−0·248	0	−0·496
22	−0·043	0	−0·045	0	−0·050	0	−0·059	0	−0·073	0	−0·099	0	−0·145	0	−0·243	0
23	0	−0·037	0	−0·040	0	−0·045	0	−0·054	0	−0·068	0	−0·093	0	−0·140	0	−0·240
24	−0·031	0	−0·032	0	−0·035	0	−0·040	0	−0·049	0	−0·064	0	−0·089	0	−0·136	0
25	0	−0·026	0	−0·028	0	−0·031	0	−0·036	0	−0·045	0	−0·059	0	−0·085	0	−0·133
26	−0·021	0	−0·022	0	−0·024	0	−0·027	0	−0·032	0	−0·041	0	−0·055	0	−0·080	0
27	0	−0·017	0	−0·018	0	−0·020	0	−0·023	0	−0·029	0	−0·037	0	−0·051	0	−0·076
28	−0·013	0	−0·013	0	−0·015	0	−0·017	0	−0·020	0	−0·025	0	−0·032	0	−0·046	0
29	0	−0·010	0	−0·010	0	−0·011	0	−0·013	0	−0·016	0	−0·020	0	−0·027	0	−0·040
30	−0·006	0	−0·007	0	−0·007	0	−0·008	0	−0·009	0	−0·012	0	−0·015	0	−0·021	0
31	0	−0·003	0	−0·003	0	−0·004	0	−0·004	0	−0·005	0	−0·007	0	−0·009	0	−0·012

Table H-5.   Weber coefficients $s^{(1)}$ for the 32 point method.

Table H-5. Continued.

μ	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	-0·009	0	-0·007	0	-0·005	0	-0·004	0	-0·004	0	-0·003	0	-0·003	0	-0·003
2	-0·021	0	-0·015	0	-0·012	0	-0·009	0	-0·008	0	-0·007	0	-0·007	0	-0·006	0
3	0	-0·027	0	-0·020	0	-0·016	0	-0·013	0	-0·011	0	-0·010	0	-0·010	0	-0·009
4	-0·046	0	-0·032	0	-0·025	0	-0·020	0	-0·017	0	-0·015	0	-0·013	0	-0·013	0
5	0	-0·051	0	-0·037	0	-0·029	0	-0·023	0	-0·020	0	-0·018	0	-0·017	0	-0·017
6	-0·080	0	-0·055	0	-0·041	0	-0·032	0	-0·027	0	-0·024	0	-0·022	0	-0·021	0
7	0	-0·085	0	-0·059	0	-0·045	0	-0·036	0	-0·031	0	-0·028	0	-0·026	0	-0·025
8	-0·136	0	-0·089	0	-0·064	0	-0·049	0	-0·040	0	-0·035	0	-0·032	0	-0·031	0
9	0	-0·140	0	-0·093	0	-0·068	0	-0·054	0	-0·045	0	-0·040	0	-0·037	0	-0·036
10	-0·243	0	-0·145	0	-0·099	0	-0·073	0	-0·059	0	-0·050	0	-0·045	0	-0·043	0
11	0	-0·248	0	-0·151	0	-0·105	0	-0·079	0	-0·065	0	-0·057	0	-0·052	0	-0·051
12	-0·500	0	-0·255	0	-0·158	0	-0·112	0	-0·086	0	-0·072	0	-0·064	0	-0·061	0
13	0	-0·508	0	-0·254	0	-0·167	0	-0·120	0	-0·095	0	-0·081	0	-0·074	0	-0·072
14	-1·427	0	-0·529	0	-0·276	0	-0·178	0	-0·131	0	-0·106	0	-0·092	0	-0·087	0
15	0	-1·448	0	-0·538	0	-0·291	0	-0·192	0	-0·144	0	-0·119	0	-0·107	0	-0·103
16	-13·011	0	-1·529	0	-0·568	0	-0·311	0	-0·209	0	-0·161	0	-0·137	0	-0·126	0
17	32·155	-13·201	0	-1·535	0	-0·584	0	-0·335	0	-0·231	0	-0·182	0	-0·160	0	-0·153
18	-13·010	32·627	-13·440	0	-1·606	0	-0·635	0	-0·367	0	-0·260	0	-0·211	0	-0·192	0
19	0	-13·199	33·440	-14·010	0	-1·700	0	-0·688	0	-0·408	0	-0·296	0	-0·251	0	-0·237
20	-1·425	0	-13·527	34·687	-14·674	0	-1·825	0	-0·758	0	-0·464	0	-0·350	0	-0·308	0
21	0	-1·444	0	-14·008	36·284	-15·560	0	-1·984	0	-0·850	0	-0·538	0	-0·425	0	-0·395
22	-0·496	0	-1·477	0	-14·669	38·486	-16·728	0	-2·198	0	-0·976	0	-0·645	0	-0·538	0
23	0	-0·501	0	-1·525	0	-15·552	41·397	-18·275	0	-2·488	0	-1·153	0	-0·805	0	-0·723
24	-0·248	0	-0·508	0	-1·591	0	-16·717	45·255	-20·442	0	-2·892	0	-1·416	0	-1·065	0
25	0	-0·237	0	-0·520	0	-1·677	0	-18·258	50·442	-23·207	0	-3·484	0	-1·837	0	-1·539
26	-0·129	0	-0·237	0	-0·536	0	-1·788	0	-20·321	57·598	-27·171	0	-4·411	0	-2·591	0
27	0	-0·125	0	-0·236	0	-0·553	0	-1·928	0	-23·149	67·883	-33·198	0	-6·029	0	-4·226
28	-0·070	0	-0·119	0	-0·234	0	-0·571	0	-2·102	0	-27·287	83·620	-43·765	0	-9·409	0
29	0	-0·063	0	-0·110	0	-0·225	0	-0·581	0	-2·305	0	-33·470	110·237	-63·816	0	-19·572
30	-0·031	0	-0·051	0	-0·094	0	-0·203	0	-0·565	0	-2·495	0	-42·889	164·027	-117·636	0
31	0	-0·031	0	-0·033	0	-0·083	0	-0·148	0	-0·457	0	-2·410	0	-59·103	326·474	-528·400

$s_v^{(2)}$ for $N = 32$

μ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	103·596	69·783	-17·658	7·184	-3·671	2·155	-1·391	0·962	-0·703	0·536	-0·424	0·346	-0·291	0·250	-0·220	0·197
2	-276·452	25·769	56·370	-17·917	8·372	-4·703	2·980	-2·016	1·457	-1·104	0·869	-0·706	0·590	-0·506	0·444	-0·398
3	154·876	-124·802	11·356	45·888	-16·417	8·329	-4·976	3·296	-2·328	1·740	-1·356	1·094	-0·910	0·777	-0·679	0·607
4	-109·508	68·941	-79·750	6·309	38·696	-14·908	7·997	-4·933	3·420	-2·499	1·918	-1·531	1·262	-1·071	0·931	-0·828
5	84·923	-48·880	43·293	-58·716	3·969	33·636	-13·645	7·627	-4·927	3·474	-2·604	2·044	-1·665	1·400	-1·209	1·069
6	-69·243	38·144	-30·507	31·420	-46·720	2·694	29·961	-12·635	7·293	-4·844	3·499	-2·680	2·146	-1·780	1·522	-1·336
7	58·263	-31·301	23·764	-21·976	24·713	-39·066	1·921	27·227	-11·841	7·018	-4·770	3·518	-2·747	2·238	-1·889	1·641
8	-50·084	26·487	-19·500	17·048	-17·161	20·469	-33·826	1·414	25·160	-11·224	6·803	-4·718	3·546	-2·816	2·333	-2·000
9	43·717	-22·878	16·512	-13·956	13·250	-14·120	17·581	-30·089	1·062	23·589	-10·755	6·648	-4·695	3·588	-2·896	2·437
10	-38·592	20·046	-14·273	11·798	-10·809	10·849	-12·055	15·519	-27·292	0·804	22·401	-10·411	6·551	-4·704	3·652	-2·993
11	34·356	-17·749	12·514	-10·186	9·115	-8·817	9·218	-10·583	13·999	-25·202	0·606	21·521	-10·177	6·509	-4·747	3·742
12	-30·778	15·836	-11·084	8·922	-7·851	7·411	-7·462	8·055	-9·496	12·854	-23·617	0·448	20·898	-10·043	6·522	-4·828
13	27·700	-14·207	9·890	-7·893	6·862	-6·365	6·250	-6·493	7·195	-8·677	11·982	-22·420	0·317	20·499	-10·002	6·593
14	-25·013	12·797	-8·866	7·033	-6·059	5·548	-5·350	5·418	-5·776	6·545	-8·050	11·318	-21·533	0·203	20·305	-10·055
15	22·634	-11·557	7·983	-6·298	5·386	-4·885	4·648	-4·621	4·800	-5·232	6·044	-7·568	10·818	-20·905	0·099	20·306
16	-20·503	10·453	-7·200	5·657	-4·811	4·330	-4·078	4·000	-4·078	4·330	-4·811	5·657	-7·200	10·453	-20·503	0
17	18·575	-9·457	6·499	-5·090	4·309	-3·854	3·602	-3·496	3·516	-3·663	3·963	-4·482	5·356	-6·924	10·202	-20·306
18	-16·813	8·551	-5·866	4·581	-3·864	3·439	-3·193	3·075	-3·059	3·142	-3·337	3·675	-4·223	5·126	-6·725	10·055
19	15·190	-7·718	5·286	-4·119	3·464	-3·071	2·837	-2·714	2·678	-2·721	2·849	-3·078	3·445	-4·020	4·953	-6·593
20	-13·682	6·946	-4·752	3·696	-3·100	2·739	-2·520	2·398	-2·350	2·369	-2·453	2·613	-2·889	3·261	-3·862	4·828
21	12·270	-6·226	4·254	-3·303	2·765	-2·437	2·234	-2·117	2·063	-2·066	2·121	-2·235	2·420	-2·698	3·113	-3·742
22	-10·940	5·548	-3·788	2·937	-2·454	2·158	-1·973	1·863	-1·806	1·800	-1·836	1·918	-2·054	2·259	-2·557	2·993
23	9·679	-4·906	3·347	-2·593	2·163	-1·898	1·732	-1·630	1·576	-1·563	1·585	-1·645	1·747	-1·900	2·121	-2·437
24	-8·476	4·295	-2·928	2·266	-1·888	1·654	-1·506	1·414	-1·364	1·347	-1·361	1·405	-1·482	1·598	-1·765	2·000
25	7·321	-3·708	2·527	-1·954	1·626	-1·423	1·294	-1·212	1·166	-1·149	1·156	-1·188	1·247	-1·336	1·464	-1·641
26	-6·206	3·143	-2·140	1·654	-1·376	1·203	-1·091	1·021	-0·981	0·963	-0·967	0·991	-1·035	1·104	-1·201	1·336
27	5·124	-2·594	1·766	-1·364	1·134	-0·990	0·898	-0·839	0·804	-0·789	0·790	-0·807	0·840	-0·883	0·967	-1·069
28	-4·069	2·060	-1·402	1·082	-0·899	0·785	-0·711	0·664	-0·635	0·622	-0·622	0·634	-0·659	0·697	-0·753	0·828
29	3·034	-1·536	1·045	-0·807	0·670	-0·584	0·529	-0·493	0·472	-0·462	0·461	-0·469	0·486	-0·514	0·553	-0·607
30	-2·015	1·020	-0·694	0·535	-0·444	0·388	-0·351	0·327	-0·313	0·305	-0·305	0·310	-0·321	0·338	-0·363	0·398
31	1·005	-0·509	0·346	-0·266	0·222	-0·194	0·175	-0·163	0·156	-0·152	0·152	-0·154	0·159	-0·167	0·180	-0·197

Table H-6. Weber coefficients $S^{(2)}$ for the 32 point method.

μ	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	-0·180	0·167	-0·159	0·154	-0·152	0·152	-0·156	0·163	-0·175	0·194	-0·222	0·266	-0·346	0·509	-1·005
2	0·363	-0·338	0·321	-0·310	0·305	-0·305	0·313	-0·327	0·351	-0·388	0·444	-0·535	0·694	-1·020	2·015
3	-0·553	0·514	-0·486	0·469	-0·461	0·462	-0·472	0·493	-0·529	0·584	-0·670	0·807	-1·045	1·536	-3·034
4	0·753	-0·697	0·659	-0·634	0·622	-0·622	0·635	-0·664	0·711	-0·785	0·899	-1·082	1·402	-2·060	4·069
5	-0·967	0·893	-0·840	0·807	-0·790	0·789	-0·804	0·839	-0·898	0·990	-1·134	1·364	-1·766	2·594	-5·124
6	1·201	-1·104	1·035	-0·991	0·967	-0·963	0·981	-1·021	1·091	-1·203	1·376	-1·654	2·140	-3·143	6·206
7	-1·464	1·336	-1·247	1·188	-1·156	1·149	-1·166	1·212	-1·294	1·423	-1·626	1·954	-2·527	3·708	-7·321
8	1·765	-1·598	1·482	-1·405	1·361	-1·347	1·364	-1·414	1·506	-1·654	1·888	-2·266	2·928	-4·295	8·476
9	-2·121	1·900	-1·747	1·645	-1·585	1·563	-1·576	1·630	-1·732	1·898	-2·163	2·593	-3·347	4·906	-9·679
10	2·557	-2·259	2·054	-1·918	1·836	-1·800	1·808	-1·863	1·973	-2·158	2·454	-2·937	3·788	-5·548	10·940
11	-3·113	2·698	-2·420	2·235	-2·121	2·066	-2·063	2·117	-2·234	2·437	-2·765	3·303	-4·254	6·226	-12·270
12	3·862	-3·261	2·869	-2·613	2·453	-2·369	2·350	-2·398	2·520	-2·739	3·100	-3·696	4·752	-6·946	13·682
13	-4·953	4·020	-3·445	3·078	-2·849	2·721	-2·678	2·714	-2·837	3·071	-3·464	4·119	-5·286	7·718	-15·190
14	6·725	-5·126	4·223	-3·675	3·337	-3·142	3·059	-3·075	3·193	-3·439	3·864	-4·581	5·866	-8·551	16·813
15	-10·202	6·924	-5·356	4·482	-3·963	3·663	-3·516	3·496	-3·602	3·854	-4·309	5·090	-6·499	9·457	-18·575
16	20·503	-10·453	7·200	-5·657	4·811	-4·330	4·078	-4·000	4·078	-4·330	4·811	-5·657	7·200	-10·453	20·503
17	-0·099	20·905	-10·818	7·568	-6·044	5·232	-4·800	4·621	-4·648	4·885	-5·386	6·298	-7·983	11·557	-22·634
18	-20·305	-0·203	21·533	-11·318	8·050	-6·545	5·776	-5·418	5·350	-5·548	6·059	-7·033	8·866	-12·797	25·013
19	10·002	-20·499	-0·317	22·420	-11·982	8·677	-7·195	6·493	-6·250	6·365	-6·862	7·893	-9·890	14·207	-27·700
20	-6·522	10·043	-20·898	-0·448	23·617	-12·854	9·496	-8·055	7·462	-7·411	7·851	-8·922	11·084	-15·836	30·778
21	4·747	-6·509	10·177	-21·521	-0·606	25·202	-13·999	10·583	-9·218	8·817	-9·115	10·186	-12·514	17·749	-34·356
22	-3·652	4·704	-6·551	10·411	-22·401	-0·804	27·292	-15·519	12·055	-10·849	10·809	-11·798	14·273	-20·046	38·592
23	2·896	-3·588	4·695	-6·648	10·755	-23·589	-1·062	30·069	-17·581	14·120	-13·250	13·956	-16·512	22·878	-43·717
24	-2·333	2·816	-3·546	4·718	-6·803	11·224	-25·160	-1·414	33·826	-20·469	17·161	-17·048	19·500	-26·487	50·084
25	1·889	-2·238	2·747	-3·518	4·770	-7·018	11·841	27·227	-1·921	39·066	-24·713	21·976	-23·764	31·301	-58·263
26	-1·522	1·780	-2·146	2·680	-3·499	4·844	-7·293	12·635	-29·961	-2·694	46·720	-31·420	30·507	-38·144	69·243
27	1·209	-1·400	1·665	-2·044	2·604	-3·474	4·927	-7·627	13·645	33·636	-3·969	58·716	-43·293	48·880	-84·923
28	-0·931	1·071	-1·262	1·531	-1·918	2·499	-3·420	4·993	-7·997	14·908	38·696	-6·309	79·750	-68·941	109·508
29	0·679	-0·777	0·910	-1·094	1·356	-1·740	2·328	-3·286	4·976	-8·329	16·417	-45·888	-11·356	124·802	-154·876
30	-0·444	0·506	-0·590	0·706	-0·869	1·104	-1·457	2·016	-2·960	4·703	-8·372	17·917	56·370	-25·789	276·452
31	0·220	-0·250	0·291	-0·346	0·424	-0·536	0·703	-0·962	1·391	-2·155	3·671	-7·184	17·658	-69·783	-103·586

Table H·6. Continued.

$$s_{\mu\nu}^{(3)} \text{ for } N = 32$$

μ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	326·474	29·462	0	15·475	0	7·333	0	4·279	0	2·838	0	2·049	0	1·575	0	1·299
2	−162·133	164·027	−16·018	0	3·986	0	2·519	0	1·647	0	1·164	0	0·877	0	0·695	0
3	0	−81·875	110·237	−20·176	0	1·127	0	1·143	0	0·848	0	0·640	0	0·502	0	0·410
4	−13·990	0	−53·644	83·620	−19·386	0	0·063	0	0·557	0	0·488	0	0·397	0	0·326	0
5	−3·965	−8·711	0	−39·790	67·883	−17·893	0	−0·411	0	0·260	0	0·295	0	0·281	0	0·225
6	0	0	−6·204	0	−31·747	57·599	−16·472	0	−0·646	0	0·089	0	0·179	0	0·178	0
7	−1·679	−2·785	0	−4·790	0	−26·578	50·442	−15·269	0	−0·771	0	−0·015	0	0·104	0	0·122
8	0	0	−2·124	0	−3·903	0	−23·032	45·255	−14·296	0	−0·841	0	−0·085	0	0·051	0
9	−0·880	−1·272	0	−1·712	0	−3·309	0	−20·499	41·397	−13·501	0	−0·883	0	−0·133	0	0·015
10	0	0	−1·020	0	−1·437	0	−2·889	0	−18·635	38·486	−12·883	0	−0·910	0	−0·167	0
11	−0·529	−0·703	0	−0·851	0	−1·244	0	−2·585	0	−17·244	36·234	−12·410	0	−0·931	0	−0·195
12	0	0	−0·586	0	−0·734	0	−1·105	0	−2·361	0	−16·199	34·637	−12·063	0	−0·950	0
13	−0·351	−0·440	0	−0·504	0	−0·649	0	−1·001	0	−2·192	0	−15·424	33·440	−11·827	0	−0·970
14	0	0	−0·378	0	−0·446	0	−0·588	0	−0·926	0	−2·067	0	−14·868	32·627	−11·697	0
15	−0·252	−0·302	0	−0·334	0	−0·402	0	−0·541	0	−0·869	0	−1·976	0	−14·495	32·155	−11·667
16	0	0	−0·268	0	−0·303	0	−0·371	0	−0·508	0	−0·828	0	−1·914	0	−14·286	32
17	−0·194	−0·223	0	−0·242	0	−0·279	0	−0·348	0	−0·483	0	−0·799	0	−1·877	0	−14·229
18	0	0	−0·203	0	−0·224	0	−0·263	0	−0·332	0	−0·467	0	−0·783	0	−1·862	0
19	−0·159	−0·177	0	−0·189	0	−0·211	0	−0·251	0	−0·321	0	−0·458	0	−0·777	0	−1·870
20	0	0	−0·165	0	−0·179	0	−0·203	0	−0·245	0	−0·316	0	−0·456	0	−0·781	0
21	−0·140	−0·150	0	−0·159	0	−0·174	0	−0·199	0	−0·243	0	−0·317	0	−0·461	0	−0·797
22	0	0	−0·144	0	−0·155	0	−0·172	0	−0·199	0	−0·245	0	−0·323	0	−0·473	0
23	−0·135	−0·137	0	−0·144	0	−0·156	0	−0·175	0	−0·204	0	−0·252	0	−0·335	0	−0·495
24	0	0	−0·138	0	−0·146	0	−0·159	0	−0·181	0	−0·214	0	−0·266	0	−0·357	0
25	−0·144	−0·138	0	−0·144	0	−0·154	0	−0·169	0	−0·193	0	−0·229	0	−0·289	0	−0·388
26	0	0	−0·148	0	−0·155	0	−0·167	0	−0·185	0	−0·214	0	−0·256	0	−0·322	0
27	−0·183	−0·159	0	−0·165	0	−0·175	0	−0·190	0	−0·213	0	−0·247	0	−0·297	0	−0·377
28	0	0	−0·187	0	−0·196	0	−0·209	0	−0·230	0	−0·259	0	−0·301	0	−0·366	0
29	−0·330	−0·232	0	−0·239	0	−0·252	0	−0·272	0	−0·301	0	−0·341	0	−0·400	0	−0·490
30	0	0	−0·338	0	−0·351	0	−0·373	0	−0·406	0	−0·453	0	−0·519	0	−0·615	0
31	−0·031	−0·648	0	−0·668	0	−0·702	0	−0·753	0	−0·827	0	−0·903	0	−1·080	0	−1·293
32	0	0	−0·032	0	−0·033	0	−0·035	0	−0·038	0	−0·042	0	−0·048	0	−0·057	0

Table H-7. Weber coefficients $S^{(3)}$ for the 32 point method.

ν

μ	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	1·062	0	0·920	0	0·818	0	0·745	0	0·694	0	0·662	0	0·642	0	0·636
2	0·573	0	0·489	0	0·429	0	0·388	0	0·357	0	0·337	0	0·324	0	0·318	0
3	0	0·346	0	0·301	0	0·269	0	0·246	0	0·230	0	0·219	0	0·212	0	0·211
4	0·274	0	0·237	0	0·209	0	0·190	0	0·175	0	0·166	0	0·161	0	0·157	0
5	0	0·195	0	0·173	0	0·155	0	0·144	0	0·135	0	0·129	0	0·125	0	0·124
6	0·162	0	0·146	0	0·132	0	0·121	0	0·113	0	0·107	0	0·104	0	0·102	0
7	0	0·119	0	0·111	0	0·103	0	0·097	0	0·092	0	0·088	0	0·086	0	0·086
8	0·084	0	0·088	0	0·086	0	0·083	0	0·079	0	0·076	0	0·074	0	0·073	0
9	0	0·055	0	0·066	0	0·068	0	0·067	0	0·065	0	0·064	0	0·063	0	0·063
10	-0·013	0	0·033	0	0·047	0	0·053	0	0·054	0	0·055	0	0·054	0	0·054	0
11	0	-0·035	0	0·015	0	0·033	0	0·041	0	0·044	0	0·045	0	0·046	0	0·045
12	-0·219	0	-0·054	0	0	0	0·021	0	0·031	0	0·035	0	0·037	0	0·037	0
13	0	-0·239	0	-0·070	0	-0·013	0	0·011	0	0·021	0	0·027	0	0·029	0	0·029
14	-0·992	0	-0·257	0	-0·085	0	-0·024	0	0·001	0	0·012	0	0·019	0	0·020	0
15	0	-1·019	0	-0·277	0	-0·099	0	-0·036	0	-0·009	0	0·004	0	0·009	0	0·011
16	-11·736	0	-1·052	0	-0·298	0	-0·114	0	-0·047	0	-0·019	0	-0·006	0	0	0
17	32·155	-11·907	0	-1·094	0	-0·319	0	-0·129	0	-0·080	0	-0·030	0	-0·018	0	-0·014
18	-14·324	32·627	-12·190	0	-1·145	0	-0·344	0	-0·146	0	-0·074	0	-0·044	0	-0·033	0
19	0	-14·571	33·440	-12·596	0	-1·208	0	-0·375	0	-0·167	0	-0·092	0	-0·062	0	-0·053
20	-1·900	0	-14·991	34·637	-13·149	0	-1·285	0	-0·411	0	-0·192	0	-0·114	0	-0·087	0
21	0	-1·957	0	-15·606	36·284	-13·876	0	-1·383	0	-0·456	0	-0·225	0	-0·147	0	-0·126
22	-0·825	0	-2·044	0	-16·455	38·486	-14·821	0	-1·507	0	-0·515	0	-0·270	0	-0·196	0
23	0	-0·869	0	-2·167	0	-17·603	41·397	-16·051	0	-1·667	0	-0·594	0	-0·338	0	-0·281
24	-0·528	0	-0·933	0	-2·341	0	-19·148	45·255	-17·668	0	-1·881	0	-0·708	0	-0·451	0
25	0	-0·578	0	-1·025	0	-2·583	0	-21·247	50·442	-19·836	0	-2·178	0	-0·889	0	-0·671
26	-0·436	0	-0·653	0	-1·161	0	-2·930	0	-24·170	57·599	-22·827	0	-2·618	0	-1·217	0
27	0	-0·511	0	-0·767	0	-1·366	0	-3·445	0	-28·405	67·883	-27·150	0	-3·347	0	-1·981
28	-0·466	0	-0·635	0	-0·956	0	-1·699	0	-4·267	0	-34·956	83·620	-33·891	0	-4·828	0
29	0	-0·628	0	-0·860	0	-1·296	0	-2·305	0	-5·737	0	-46·220	110·237	-45·757	0	-9·569
30	-0·757	0	-0·979	0	-1·352	0	-2·053	0	-3·649	0	-8·976	0	-69·760	164·027	-73·139	0
31	0	-1·613	0	-2·115	0	-2·984	0	-4·575	0	-8·247	0	-20·295	0	-147·668	326·474	-283·566
32	-0·069	0	-0·088	0	-0·118	0	-0·171	0	-0·275	0	-0·529	0	-1·451	0	-12·979	—

Table H-7. Continued.

424
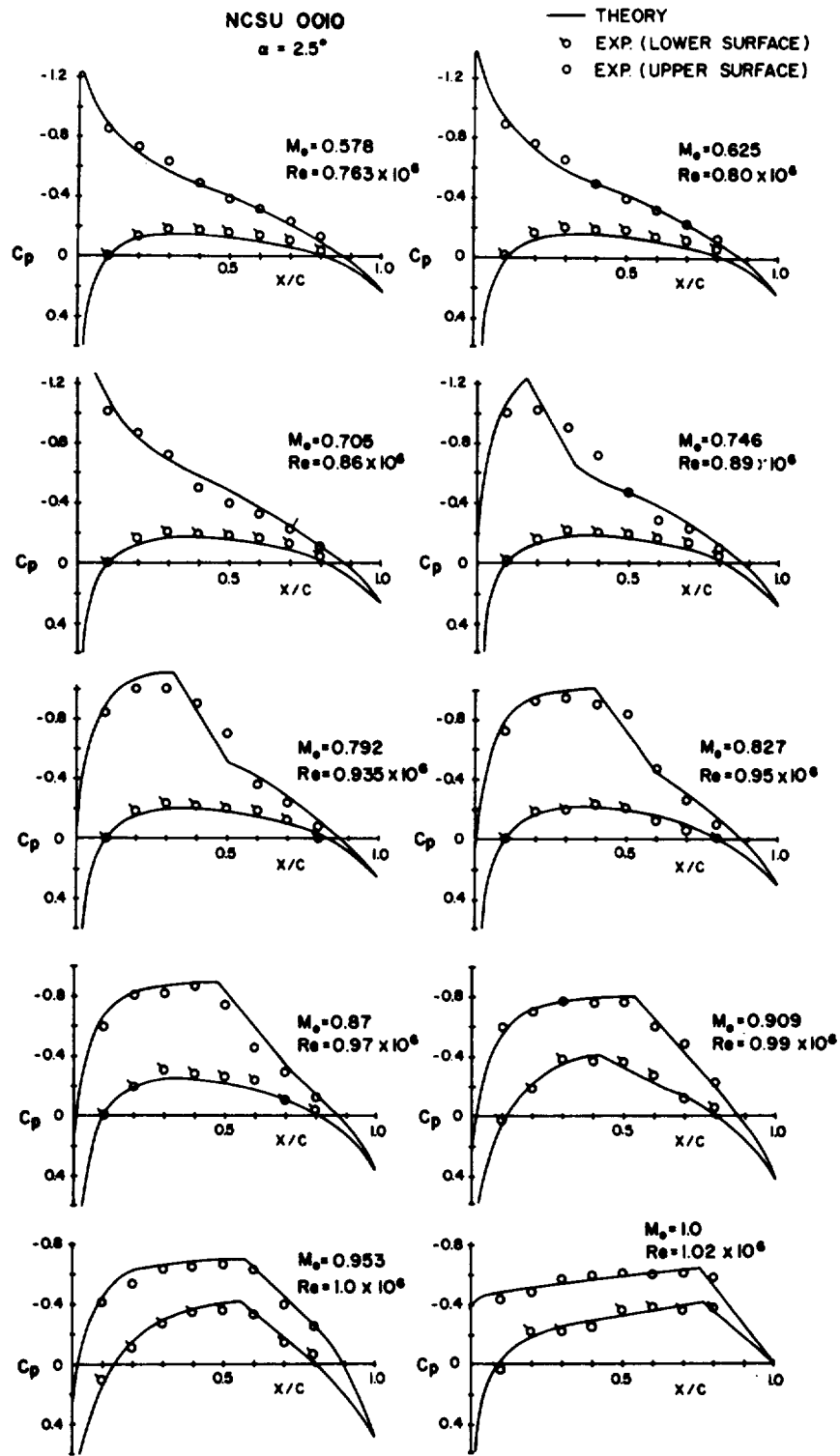
Figure H-2. Comparison of theory and experiment
for one airfoil.

This program is written in FORTRAN IV and is designed to run in single precision on an IBM 370-165 computer. Execution requires 32,000 bytes of core storage and approximately 3 seconds to produce the two-dimensional zero Mach number pressure distribution for an airfoil at six angles of attack by the 16-point Weber method. For a given run, the program requires the following input data:

(1) The 240 elements of the Matrix S1 given in A.R.C. R&M 2918 and listed herein as Table H-2.

(2) The 240 elements of the Matrix S2 also given in A.R.C. R&M 2918 and listed herein as Table H-3.

(3) The 256 elements of Matrix S3 also given in A.R.C. R&M 2918 and listed herein as Table H-4.

(4) The 72 characters of the array TITLE which are used as a header for identifying output. It is an alphameric array and usually contains information about the type of airfoil, run number, and any other such information desired by the programmer as a header.

(5) The values of X, the x-coordinates (given as X/C of the airfoil to be considered) from Table H-1.

(6) The values of Y (the normalized y-coordinate of the airfoil to be considered) corresponding to the values of X.

(7) The values of angle of attack, A, and airfoil leading edge radius, RHO, for a given airfoil, followed by a last card indicator giving A a value of greater than 15.

(8) Additional data sets containing information from item (4) through item (7), for another airfoil, if needed. Only two airfoils may be considered per run.

Format specifications for these variables may be found in Figure H-3. A sample data set of the NCSU 0010 may be found in Figure H-4 and the output for that data set is given in Figure H-5.

426
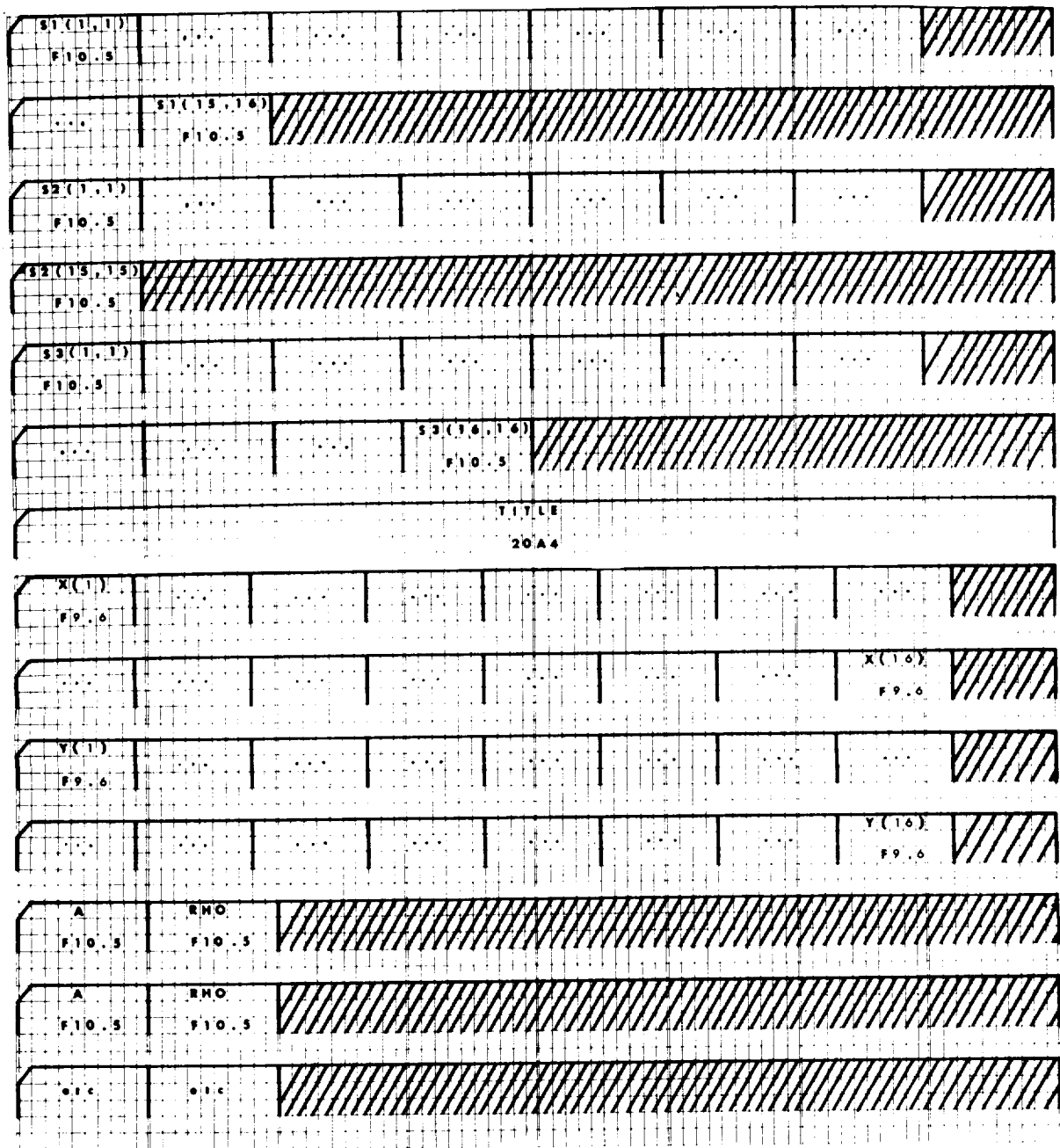
Figure H-3. Format specification of input data for TRINSON.

A   RHO
2.50   0.011

A   RHO
1.73   0.011

A   RHO
1.00   0.011

...   .   Y(16)
0.049   0.0499   0.0478   0.0422   0.0344   0.0238   0.0126   0.001

Y(1)   ...
0.0013   0.0046   0.0104   0.0173   0.0252   0.0333   0.0403   0.0468

...   .   X(16)
0.4025   0.3087   0.2222   0.1464   0.08427 0.03806 0.00961 0.00001

X(1)   ...
0.99   0.9619   0.915   0.8536   0.7778   0.6913   0.5975   0.5

                        TITLE
                      NCSU 0010

...   .   S3(16,16)
-.742   .0   -6.505   .0

...

S3(1,1) ...
82.013   7.458   0.   4.031   0.   2.006   0.

S2(16,16)
-25.769

...

S2(1,1) ...
25.769   17.917   -4.703   2.016   -1.104   .706   -.506

...   S1(16,16)
82.013   -132.103

...

S1(1,1) ...
82.013   -15.061 0.   -.651   0.   -.136   0.

Figure H-4. Example data set for TRINSON.

428

Program Listing - TRINSON

```
C     PROGRAM TO CALCULATE THE INCOMPRESSIBLE PRESSURE DISTRIBUTION
C     ABOUT 2-D SYMMETRICAL AIRFOILS (REFERENCE- WEBER,J.- A.R.C. R.&M.
C     2918, JULY, 1953.)
C     THE INPUT DATA NEEDED ARE AS FOLLOWS :
C
C     S11(I,J),S21(I,J),& S31(I,J) ARE COEFFICIENTS LISTED IN R.&M. 2918
C     FOR N=16
C     TITLE= PERTINENT INFORMATION TO BE LISTED AT FIRST OF OUTPUT
C     X(I) & Y(I) ARE THE AIRFOIL ABSCISSA AND ORDINATES (X/C & Y/C)
C        THESE ARE INPUT FROM THE TRAILING EDGE FORWARD FOR THE
C        SPECIFIC LOCATIONS FOR N = 16
C     A= AIRFOIL ANGLE OF ATTACK (DEGREES)
C     RHO= AIRFOIL LEADING EDGE RADIUS (DECIMAL FRACTION OF CHORD)

      DIMENSION S11(16,16),S21(16,16),S31(16,16),X(16),Y(16),S11(16),S22(16
     1,S33(16),CPU(16),CPL(16)
      DIMENSION TITLE(18)
      COMMON /IHOUT/JREAD,JWRITE
      JREAD=1
      JWRITE=3
      READ (JREAD,4) ((S11(I,J),J=1,16),I=1,15)
      READ (JREAD,4) ((S21(I,J),J=1,15),I=1,15)
      READ (JREAD,4) ((S31(I,J),J=1,16),I=1,16)
    4 FORMAT (7F10.5)
      READ (JREAD,101) (TITLE(I),I=1,18)
  101 FORMAT (18A4)
      DO 20 JJJ=1,2
      JSKIP=1
    5 READ (JREAD,5) (X(I),I=1,16)
    7 READ (JREAD,5) (Y(I),I=1,16)
    6 FORMAT (8F9.6)
      READ (JREAD,6) A,RHO
      FORMAT (2F10.5)
      IF (A.GT.15.) GO TO 19
      Y(16)=SQRT(RHO/2.)
      JSKIP=JSKIP+1
      IF (JSKIP.EQ.1) GO TO 18
      WRITE (JWRITE,39)
   39 FORMAT (1H1,///,19X,35HINCOMPRESSIBLE PRESSURE COEFFICIENT)
      JSKIP=0
   18 WRITE (JWRITE,13) (TITLE(I),I=1,18)
   13 FORMAT (///,18A4)
   14 WRITE (JWRITE,14) A
      FORMAT (///,24X,17HANGLE OF ATTACK= ,F10.6)
      IF (JJJ.EQ.2) GO TO 91
   12 WRITE (JWRITE,12)
      FORMAT (///,15X,3HX/C,3HV/C,12X,8HCP-UPPER,//)
      GO TO 90
   91 WRITE (JWRITE,31)
   31 FORMAT (///,15X,3HX/C,3HV/C,12X,8HCP-LOWER,//)
   90 DO 100 J=1,15
      S11(J)=0.
      S22(J)=0.
      S33(J)=0.
      DO 8 I=1,15
      S11(J)=S11(J)+S11(I,J)*Y(I)
    8 CONTINUE
      DO 9 I=1,15
      S22(J)=S22(J)+S21(I,J)*Y(I)
    9 CONTINUE
      DO 10 I=1,15
      S33(J)=S33(J)+S31(I,J)*Y(I)
   10 CONTINUE
      B=COS(A/57.2958)*(1.+S11(J))
      C=SIN(A/57.2958)*SQRT((1.-X(J))/X(J))*(1.+S33(J))
      D=1.+(S22(J))**2
      IF (JJJ.EQ.2) GO TO 22
      CPU(J)=1.-((B+C)**2/D)
      GO TO 107
   22 CPU(J)=1.-(((B-C)**2)/D)
  107 WRITE (JWRITE,11) X(J),Y(J),CPU(J)
   11 FORMAT (3F20.6)
  100 CONTINUE
      WRITE (JWRITE,32)
   32 FORMAT (///,72H************************************************)
      GO TO 7
   19 IF (JJJ.EQ.2.AND.A.GT.15.) CALL EXIT
   20 CONTINUE
      STOP
      END
```

INCOMPRESSIBLE PRESSURE COEFFICIENT

NCSU 0010

ANGLE OF ATTACK= 1.000000

X/C	Y/C	CP-UPPER
0.990000	0.001300	0.231517
0.961900	0.004800	0.127022
0.915000	0.010400	0.040493
0.853600	0.017300	-0.018052
0.777800	0.025200	-0.079037
0.691300	0.033300	-0.146581
0.597500	0.040300	-0.192896
0.500000	0.045800	-0.244597
0.402500	0.049000	-0.284362
0.308700	0.049900	-0.337142
0.222200	0.047800	-0.384627
0.144400	0.042200	-0.401874
0.084270	0.034400	-0.438526
0.038060	0.023800	-0.400684
0.009610	0.012600	-0.379138

INCOMPRESSIBLE PRESSURE COEFFICIENT

NCSU 0010

ANGLE OF ATTACK= 1.000000

X/C	Y/C	CP-LOWER
0.990000	0.001300	0.234393
0.961900	0.004800	0.137868
0.915000	0.010400	0.059098
0.853600	0.017300	0.008939
0.777800	0.025200	-0.047006
0.691300	0.033300	-0.097166
0.597500	0.040300	-0.129807
0.500000	0.045800	-0.164456
0.402500	0.049000	-0.184141
0.308700	0.049900	-0.209759
0.222200	0.047800	-0.225841
0.144400	0.042200	-0.191749
0.084270	0.034400	-0.169170
0.038060	0.023800	0.001849
0.009610	0.012600	0.279527

NCSU 0010

ANGLE OF ATTACK= 1.730000

X/C	Y/C	CP-UPPER
0.990000	0.001300	0.230637
0.961900	0.004800	0.123577
0.915000	0.010400	0.034281
0.853600	0.017300	-0.027378
0.777800	0.025200	-0.092056
0.691300	0.033300	-0.164194
0.597500	0.040300	-0.215624
0.500000	0.045800	-0.273753
0.402500	0.049000	-0.321191
0.308700	0.049900	-0.384461
0.222200	0.047800	-0.451107
0.144400	0.042200	-0.481902
0.084270	0.034400	-0.571242
0.038060	0.023800	-0.563684
0.009610	0.012600	-0.671492

NCSU 0010

ANGLE OF ATTACK= 1.730000

X/C	Y/C	CP-LOWER
0.990000	0.001300	0.238915
0.961900	0.004800	0.142332
0.915000	0.010400	0.066266
0.853600	0.017300	0.013299
0.777800	0.025200	-0.028072
0.691300	0.033300	-0.078734
0.597500	0.040300	-0.106573
0.500000	0.045800	-0.135163
0.402500	0.049000	-0.147079
0.308700	0.049900	-0.164111
0.222200	0.047800	-0.167905
0.144400	0.042200	-0.114599
0.084270	0.034400	-0.070690
0.038060	0.023800	0.132413
0.009610	0.012600	0.467542

INCOMPRESSIBLE PRESSURE COEFFICIENT

NCSU 0010

ANGLE OF ATTACK= 2.500000

X/C	Y/C	CP-UPPER
0.990000	0.001300	0.229642
0.961900	0.004800	0.120244
0.915000	0.010400	0.028048
0.853600	0.017300	-0.036893
0.777800	0.025200	-0.105446
0.691300	0.033300	-0.182501
0.597500	0.040300	-0.239398
0.500000	0.045800	-0.304420
0.402500	0.049000	-0.360113
0.308700	0.049900	-0.434727
0.222200	0.047800	-0.516876
0.144400	0.042200	-0.562230
0.084270	0.034400	-0.694113
0.038060	0.023800	-0.744757
0.009610	0.012600	-1.009651

INCOMPRESSIBLE PRESSURE COEFFICIENT

NCSU 0010

ANGLE OF ATTACK= 2.500000

X/C	Y/C	CP-LOWER
0.990000	0.001300	0.241019
0.961900	0.004800	0.147350
0.915000	0.010400	0.074236
0.853600	0.017300	0.030514
0.777800	0.025200	-0.013013
0.691300	0.033300	-0.059087
0.597500	0.040300	-0.081843
0.500000	0.045800	-0.106280
0.402500	0.049000	-0.109876
0.308700	0.049900	-0.116511
0.222200	0.047800	-0.107897
0.144400	0.042200	-0.043574
0.084270	0.034400	0.028751
0.038060	0.023800	0.260499
0.009610	0.012600	0.635264

Figure H-5. Sample output for TRINSON.

This program is written in FORTRAN IV and designed to run in single precision on an IBM 370-165 computer. Execution requires 23 seconds and 68,000 bytes of core to produce complete pressure distributions about a two-dimensional airfoil with a round leading edge for ten Mach numbers from zero to one at one angle of attack. The calculations are carried out first for the upper surface and then the lower. For each airfoil the program requires the following input:

(1) The 72 characters of the array TITLE. This array, which should contain an accurate description of the airfoil, is printed at the first of the output for identification.

(2) The variable NNN and the value of JPRINT, an integer from 2 to 10 which specifies the print information. NNN should first take on the value 1, which implies that the data to follow is that of the upper surface and later, the value 2, which implies that the data to follow is that of the lower surface. The number of points printed may be calculated by (10 + 90/JPRINT).

(3) The 28 values of the array X, the x-coordinate (given in fraction of chord) as measured from the leading edge. These values must be specified at every one percent chord starting from x/c = 0.01 to x/c = 0.10. From x/c = 0.10 to x/c = 1.00 values at each five percent chord are sufficient.

(4) The 28 corresponding values of the array Y, the absolute value of a y-coordinate of either the upper or lower surface according to the value of NNN previously specified.

(5) The 28 values of the array CPI. These are the incompressible pressure coefficients (for the angle of attack of interest) corresponding to X and Y above. Note from (2) and (3) above that these values of CPI are not given at the same points as specified by TRINSON or TRANSON (the 16-point or 32-point incompressible pressure distribution programs respectively, from which this information is supplied). The data from these programs must be plotted, smoothed, and the correct values of CPI read off at the specific coordinate locations of (2) and (3).

(6) The angle of attack A in degrees, the airfoil maximum thickness in fraction of chord, the airfoil slope, E, given at x/c = 0.01, the airfoil slope, G, given at x/c = 1.0, the maximum y-coordinate above the chord line ZMAX in fraction of chord, and CREST, the x-position of the maximum y-coordinate given in percent chord.

(7) The value of N1N, the number of free stream Mach numbers for which the calculations are to be made.

(8) The values of a free stream Mach number XM, a Reynolds Number, RE, corresponding to that Mach number, and KTPE which takes on the value one to specify the type of boundary layer as laminar and the value zero to indicate a turbulent boundary layer. Item (8) is repeated until N1N Mach numbers have been entered.

(9) Items (2) through (8) should be entered now for the lower surface. Only one airfoil may be considered for each run.

Format specification for these variables may be found in Figure H-6. A sample data set of the NCSU 0010 may be found in Figure H-7 and the output of that data set is given in Figure H-8.

Figure H-6. Format specification of input data for COMPR.

433

Figure H-7. Example data set for COMPR.

434

Program Listing – COMPR

```
C     PROGRAM TO CALCULATE PRESSURE DISTRIBUTIONS ABOUT TWO-DIMENSIONAL
C     AIRFOILS WITH ROUND LEADING EDGES AT ONE GIVEN ANGLE OF ATTACK AND
C     AT MACH NUMBERS FROM M=0 TO M=1.0 . THE CALCULATIONS ARE
C     CARRIED OUT FOR FIRST THE UPPER SURFACE AND THEN FOR THE LOWER
C     SURFACE.
C
C     THE INPUT DATA NEEDED ARE DESCRIBED BELOW.
C
C     TITLE= PERTINENT DESCRIPTION OF AIRFOIL THAT IS TO BE PRINTED
C            AT THE FIRST OF THE OUTPUT.
C
C     NNN=1, TO SPECIFY UPPER SURFACE
C     JPRINT= AN INTEGER FROM 2 TO 10 WHICH SPECIFIES THE PRINT
C             INFORMATION.  THE NUMBER OF POINTS PRINTED MAY BE
C             CALCULATED BY (10*90/JPRINT).
C     X(I)= X/C, AIRFOIL ABSCISSA MEASURED FROM LEADING EDGE IN FRACTION
C           OF CHORD.  THESE MUST BE GIVEN EVERY ONE PERCENT CHORD
C           STARTING FROM X/C=0.01 TO X/C=0.1 . FROM X/C=0.1 TO X/C=1.0
C           COORDINATES EACH 5% CHORD ARE SUFFICIENT.  THIS GIVES
C           28 POINTS ON THE UPPER SURFACE.
C     Y(I)= Y/C, AIRFOIL ORDINATES CORRESPONDING TO THE SAME
C     CPI(I)= INCOMPRESSIBLE PRESSURE COEFFICIENTS (FOR THE ANGLE OF
C             INTEREST) CORRESPONDING TO X(I) AND Y(I) ABOVE.
C     A= ANGLE OF ATTACK IN DEGREES.
C     THIC= AIRFOIL MAXIMUM THICKNESS IN FRACTION OF CHORD, THIC=T/C
C     E= AIRFOIL SLOPE AT X/C=0.01 .
C     G= AIRFOIL SLOPE AT X/C=1.0 .
C     ZMAX= THE MAXIMUM Y(I) ORDINATE ABOVE THE CHORD LINE.
C     CREST= THE (X/C)*100. POSITION OF MAX Y/C.
C     NIN= NUMBER OF MACH NUMBERS FOR WHICH CALCULATIONS ARE TO BE MADE.
C     XM= FREE-STREAM MACH NUMBERS FOR WHICH CALCULATIONS ARE TO BE
C         MADE.  THERE MUST BE NIN VALUES OF XM.
C     RE= REYNOLDS NUMBER CORRESPONDING TO XM
C     KTPE= TYPE OF BOUNDARY LAYER CORRESPONDING TO XM AND RE
C     KTPE= 1, FOR LAMINAR BOUNDARY LAYER
C     KTPE= 0, FOR TURBULENT BOUNDARY LAYER
C
C     NNN= 2 TO SPECIFY LOWER SURFACE.
C     REPEAT ABOVE INPUTS FOR THE LOWER SURFACE.
C     READ IN THE LOWER SURFACE ORDINATES Y(I) AS POSITIVE EVEN THOUGH
C     THEY ARE BELOW THE CHORD LINE.
C
C     REFER TO FORMAT STATEMENTS IN PROGRAM FOR PRECISE FORMAT SPECS.
C
C     XMC = CRITICAL MACH NUMBER
C     DIV(J) = SLOPE OF AIRFOIL SURFACE (RADIANS)
C     P = STATIC PRESSURE
C     XX = X/C COORDINATE OF AIRFOIL
C     YY = Y/C COORDINATE OF AIRFOIL
C     THETA(J) = ARCTANGENT OF DIV(J)
C     LAMBDA = PROPORTIONALITY FACTOR FOR CALCULATION OF THE SONIC
C              PRESSURE DISTRIBUTION
C     TO = STAGNATION TEMPERATURE (DEGREES RANKINE)
C     PO = STAGNATION PRESSURE (POUNDS/FT.**2)
C     RO = STAGNATION DENSITY (SLUGS/FT.**3)
C     TI = FREE-STREAM STATIC TEMPERATURE (DEGREES RANKINE)
C     CI = FREE-STREAM SPEED OF SOUND (FT./SEC.)
C     UNI = FREE-STREAM VELOCITY (FT./SEC.)
C
C     LOGICAL LOG
      REAL*8 H,Y,AM,E,G,A,B,C,D,X,XT,FUM,FDIV,CPI,UI

      REAL*8 CPS,XX,YY,CREST,CL,CD,DIV,CPP,CL1,CD1
      DIMENSION XMS(200),XML(200),XMLC(200),XMM(200),PMM(200),PFPO(150)
      DIMENSION X(99),Y(99),U1(99),H1(100),AM(100),XX(100),DIV(10
     10),T(100),P(100),U(100),XML(110),XA(110),FF(100),CM(100),CPS(100),
     1PS(100),CPI(100),CPG(100),XMLG(100),PG(100),P2G(100),P2GPD(100),PPD
     1(100)
      DIMENSION XMLGS(100),CPP(200),CPI(200),TITLE(18),XMM(100)
      COMMON /INOUT/JREAD,JWRITE,JPRINT
      JREAD=1
      JWRITE=3
      READ (JREAD,5) (TITLE(I),I=1,18)
    5 FORMAT (18A4)
      WRITE (JWRITE,10) (TITLE(I),I=1,18)
   10 FORMAT (1H1,18A4)
   15 K=28
      READ (JREAD,20) NNN,JPRINT
   20 FORMAT (2I10)
      READ (JREAD,25) (X(I),I=1,K)
   25 FORMAT (D15.6)
      READ (JREAD,25) (Y(I),I=1,K)
      READ (JREAD,25) (CPI(I),I=1,K)
      READ (JREAD,30) A,THIC,E,G,ZMAX,CREST
   30 FORMAT (6F12.6)
      DUMB=A
      WRITE (JWRITE,35) A
   35 FORMAT (///,23X,16HANGLE OF ATTACK=,F5.2,2X,7HDEGREES)
      IF (NNN.EQ.2) GO TO 45
      WRITE (JWRITE,40)
   40 FORMAT (///,30X,13HUPPER SURFACE)
      GO TO 55
   45 WRITE (JWRITE,50)
   50 FORMAT (///,30X,13HLOWER SURFACE)
   55 IF (NNN.EQ.2)A=-A
      WRITE (JWRITE,60)
   60 FORMAT (///,7X,65HTHE AIRFOIL COORDINATES AND INCOMPRESSIBLE PRESSU
     1RE DISTRIBUTION )
      WRITE (JWRITE,65)
   65 FORMAT (///,13X,3HX/C,17X,3HY/C,17X,3HCPI//)
      DO 70 I=1,K
   70 WRITE (JWRITE,75) X(I),Y(I),CPI(I)
   75 FORMAT (3F20.6)
C     USING THE KARMEN-TSIEN RULE, CALCULATE THE CRITICAL MACH NUMBER
      DO 80 I=2,K
      IF (CPI(I).GE.CPI(I-1)) GO TO 85
   80 CONTINUE
   85 C=CPI(I-1)
      XMC=(-1.8394)*C**10+(-18.4109)*C**9+(-78.8408)*C**8+(-188.533)*C**
     17+(-275.433)*C**6+(-252.315)*C**5+(-142.469)*C**4+(-45.9726)*C**3
     1+(-6.45162)*C**2+(-.612756)*C+(1.00011)
      WRITE (JWRITE,90) XMC
   90 FORMAT (///,18X,26HTHE CRITICAL MACH NUMBER= F6.3)
C     CALCULATE THE INCOMPRESSIBLE VELOCITY RATIOS CORRESPONDING
C     TO THE INCOMPRESSIBLE PRESSURE COEFFICIENTS
      DO 95 I=1,K
   95 UI(I)=(1.-CPI(I))**0.5
C     CALCULATE THE SONIC (M=1) PRESSURE DISTRIBUTION (THOMPSON & WILBY)
```

```
121      DO 100 I=2,K
122      H(I)=X(I)-X(I-1)
123 C
124 C    SUBROUTINES 'SPLINE', 'FUNCT', AND 'FIRST' CALCULATE THE SURFACE
125 C    ORDINATES AND SLOPES
126      KTYPE=3
127      CALL SPLINE(K,Y,X,AM,KTYPE,E,G)
128      XT=.0
129      WRITE (JWRITE,105)
130  105 FORMAT (///,9X,2HXX,13X,2HYY,12X,3HDIV,12X,5HTHETA//)
131      DO 120 J=1,100
132      XT=XT-.01
133      DO 110 I=2,K
134      IF (XT.LT.X(I)) GO TO 115
135  110 CONTINUE
136      I=K
137  115 CALL FUNCT(AM(I-1),AM(I),X(I-1),X(I),Y(I-1),Y(I),XT,FUN)
138      CALL FIRST(AM(I-1),AM(I),X(I-1),X(I),Y(I-1),Y(I),XT,FDIV)
139      XX(J)=XT
140      YY(J)=FUN
141      DIV(J)=FDIV
142      T(J)=DATAN(DIV(J))
143  120 IF (LOG(J)) WRITE (JWRITE,125) XX(J),YY(J),DIV(J),T(J)
144  125 FORMAT (4F15.6)
145      WRITE (JWRITE,130)
146  130 FORMAT (///,10X,58HCALCULATION OF THE LOCAL MACH NUMBER NEAR THE N
147     1OSE FOR M=1)
148      WRITE (JWRITE,135)
149  135 FORMAT (///,9X,3HXX/C,12X,3HY/C,12X,6HP/PO,11X,7HM-LOCAL,9X,2HCP//)
150      DO 145 I=1,K
151      U(I)=U1(I)
152      TO=518.69
153      RO=.002377
154      XM=1.0
155      PO=2116.22
156      TI=TO/(1.+.2*XM*XM)
157      CI=49.1*SQRT(TI)
158      UN=CI*XM
159      UL=UN/U(I)
160      PLPO=(1.-(UL**2.)*O)/17.*PO1)**3.5
161      XM(I)=SQRT(5.*(1./PLPO)**.2857-1.))
162      XMS(I)=XM(I)
163      CPS(I)=(1.-.4285*(XM*XM))*((PLPO/.52828)-1.)
164      CPS(I)=(1.-.4285*(XM*XM))*((PLPO/.52828)-1.)
165      J=1
166  140 IF (I.NE.K(+1) GO TO 150
167      IF (I.GE.1.0) GO TO 150
168      IF (XML(I).GE.1.0) WRITE (JWRITE,140) XX(I),YY(I),PLPO,XMS(I),CPS(I)
169  145 CONTINUE
170  150 IF (I.EQ.K(+1) GO TO 175
171      WRITE (JWRITE,155) I
172  155 FORMAT (///,10X,43HLOCAL MACH NUMBER HAS EXCEEDED UNITY BEFORE,I4,1
173     14H PERCENT CHORD)
174      F=0.1/12.*2MAXI
175      XM=1XM(I))**2
176      XMW=2.45*ATAN(.4082*SQRT(XM-1.))-ATAN(SQRT(XM-1.))
177      TOP=(XMW*57.2958*(T(I)*57.2958)-A)*F
178      J=1
179      XA(J)=XH(J)**.33333
180      FF(J)=(14090.4)*AA**8*(-72092.1)*AA**7*(158582.)*AA**6*(-195406.)*
```

```
181      1AA**5*(147818.)*AA**4*(-70640.2)*AA**3*(21045.1)*AA**2*(-3660.37)*
182      1AA*(304.474)
183      BOT=FF(J)
184      ALAM=TOP/BOT
185      WRITE (JWRITE,160) ALAM
186  160 FORMAT (///,31X,8HLAMBDA= ,F10.6,//)
187      WRITE (JWRITE,165)
188  165 FORMAT (1H1,//,8X,67HPRESSURE DISTRIBUTION BETWEEN SONIC POINT AND
189     1TRAILING EDGE FOR M=1)
190      WRITE (JWRITE,135)
191      DO 170 J=1,100
192      XA(J)=(XX(J))**.33333
193      AA=XA(J)
194      FF(J)=(14090.4)*AA**8*(-72092.1)*AA**7*(158582.)*AA**6*(-195406.)*
195      1AA**5*(147818.)*AA**4*(-70640.2)*AA**3*(21045.1)*AA**2*(-3660.37)*
196      1AA*(304.474)
197      OM(J)=(FF(J)*ALAM)/F)*A-T(J)*57.2958
198      O=OM(J)
199      XML(J)=(-.14759E-6)*O**6*(.86365BE-5)*O**5*(-.198719E-3)*O**4*(.22
200     197A7E-2)*O**3*(-.14349TE-1)*O**2*(.835506E-1)*O*(1.00889)
201      XMS(J)=XML(J)
202      XR=100.*XL(J)
203      PS(J)=(1.+.2*XML(J))**2)**(-3.5)
204      PDP(J)=(1.2)**3.5
205      CPS(J)=(1.-.4285)*((PS(J)*PDP)-1.)
206  170 IF (LOG(J)) WRITE (JWRITE,140) XX(J),YY(J),PS(J),XML(J),CPS(J)
207      I=1
208  175 M=1
209      E=(CP(I2)-CP(I))/(X(2)-X(1))
210      G=(CP(IK)-CP(IK-1))/(X(IK)-X(K-1))
211      CALL SPLINE(K,CP,X,AM,KTYPE,E,G)
212      XT=0.0
213      DO 180 J=1,100
214      XT=XT-.01
215      DO 180 L=2,K
216      IF (XT.LT.X(L)) GO TO 185
217  180 CONTINUE
218      L=K
219  185 CALL FUNCT(AM(L-1),AM(L),X(L-1),X(L),CP(L-1),CP(L),XT,FUN)
220      CPP(J)=FUN
221      XX(J)=XT
222      IF (I.EQ.K(+1) WRITE (JWRITE,195)
223  195 FORMAT (1H1,//,10X,35HMACH NUMBER=1 PRESSURE DISTRIBUTION//)
224      IF (I.EQ.K(+1) WRITE (JWRITE,200)
225  200 FORMAT (///,9X,3HX/C,12X,3HY/C,12X,2HCP//)
226      DO 205 MM=1,100
227      IF (I.NE.K(+1) GO TO 205
228      CPS(MM)=CPP(MM)
229      PS(MM)=-0.528*CPS(MM)*(.3696)
230      IF (LOG(MM)) WRITE (JWRITE,210) XX(MM),YY(MM),CPS(MM)
231  205 CONTINUE
232  210 FORMAT (3F15.6)
233      CONTINUE
234      DO 215 IT=2,100
235      DUMA=(1VY(IT)-YY(IT-1))/(XX(IT)-XX(IT-1))
236      DUMA=ATAN(DUMA)*(180./3.14159)
237      IF (DUMA.LE.DUMB) GO TO 220
238  215 J=IT
239  220 J=IT
240      READ (JREAD,225) NIN
```

```
MN 241   225 FORMAT (I2,
MN 242       DO 360 I=1,MIN
MN 243   230 READ (JREAD,230) XM,RE,KTPE
MN 244       FORMAT (F10.,E20.5,I2)
MN 245       IF (XM.LE.XMC) GG TO 235
MN 246       GO TO 250
MN 247   235 WRITE (JWRITE,240) XM
MN 248   240 FORMAT (1H1,//,30X,13HMACH NUMBER= ,F10.5,//)
MN 249       WRITE (JWRITE,135)
MN 250       DO 245 III=1,99
MN 251       PHMI=(1.+.2*(XM**2))**(-3.5)
MN 252       BETA=SQRT(1.-XM**2)
MN 253       BE=2.*(BETA+1.)
MN 254       CP(III)=CPP(III)/(BETA*(CPP(III)*(XM**2))/BE)
MN 255       PHMI(III)=CP(III)*(.7*PHMI*XM*XM)+PHMI
MN 256       XMM(III)=SQRT(5.*(PHMI(III)**(-.2857)-1.))
MN 257   245 IF (LOG(III)) WRITE (JWRITE,140) XX(III),YY(III),PHMI(III),XMM(III)
MN 258      1,CP(III)
MN 259       GO TO 360
MN 260
MN 261       THE FOLLOWING SECTION USES THE PRANDTL GLAUERT RULE TO FIND THE
MN 262       PRESSURE DISTRIBUTION FROM THE CREST OF THE AIRFOIL TO THE
MN 263       TRAILING EDGE
MN 264
MN 265   250 BM=(XM-XMC)/(1.-XMC)
MN 266       PIPSR=(-.430821)*BM**5+(1.32859)*BM**4+(-1.6355)*BM**3+(1.06519)*B
MN 267      1M**2+(-.428408)*BM+0.100895+(1.-BM)*(.0117)*A
MN 268       PCRPSR=(-.49244)*BM**5+(1.49793)*BM**4+(-1.7943)*BM**3+(1.1174)*BM
MN 269      1**2+(-.408045)*BM+0.0791467
MN 270       PHMI=(1.+.2*(XM**2))**(-3.5)
MN 271       DO 255 K=J,100
MN 272       CPG(K)=CPP(K)/SQRT(1.-XM**2)
MN 273       PG(K)=CPG(K)*(.7*PHMI*XM*XM)+PHMI
MN 274       IF (PG(K).LT.0.0) GO TO 255
MN 275       XMLGS(K)=(5.*(PG(K)**(-.2857)-1.))
MN 276       IF (XMLGS(K).LT.0.) GO TO 255
MN 277       XMLGS(K)=SQRT(XMLGS(K))
MN 278       PIPG=PIPSR+PSI(K)
MN 279       P2GPI=(63.1078)*PIPD**3+(-75.3377)*PIPD**2+(26.4082)*PIPD+1.2277
MN 280      1)
MN 281   255 P2GPO(K)=P2GPI*PIPO
MN 282       DO 260 II=J,100
MN 283       IF (PG(II).GE.P2GPO(II)) GO TO 265
MN 284   260 CONTINUE
MN 285   265 RES=RE*XX(II)
MN 286
MN 287       CALCULATION OF THE INCOMPRESSIBLE BOUNDARY LAYER THICKNESS
MN 288
MN 289       IF (KTPE.EQ.1)DEL=(256.*XX(II))/SQRT(RES)
MN 290       IF (KTPE.NE.1)DEL=(1.85*XX(II))/(RES**.2)
MN 291       XB=XX(II)-DEL
MN 292       XAA=100.*XB
MN 293       IXA=XAA
MN 294       WRITE (JWRITE,270) II,XM
MN 295   270 FORMAT (///,13X,17HSHOCK LOCATED AT ,I3,32H PERCENT CHORD FOR MACH
MN 296      1 NUMBER= ,F6.3,//)
MN 297       IF (KTPE.EQ.1) WRITE (JWRITE,275) RE
MN 298   275 FORMAT (//,14X,41HLAMINAR BOUNDARY LAYER--REYNOLDS NUMBER= ,E14.3,/
MN 299      1)
MN 300
MN 301       IF (KTPE.NE.1) WRITE (JWRITE,280) RE
MN 302   280 FORMAT (//,14X,42HTURBULENT BOUNDARY LAYER--REYNOLDS NUMBER=,E14.3/
MN 303      1)
MN 304       WRITE (JWRITE,135)
MN 305       PCR=PCRPSR+PSI(J)
MN 306       JRR=J+1
MN 307       DO 305 L=J,JRR
MN 308       IF (II.EQ.J) GO TO 285
MN 309       GO TO 300
MN 310   285 WRITE (JWRITE,290)
MN 311   290 FORMAT (60X,14HSHOCK AT CREST)
MN 312       DO 295 III=1,99
MN 313       PHMI=(1.+.2*(XM**2))**(-3.5)
MN 314       BETA=SQRT(1.-XM**2)
MN 315       BE=2.*(BETA+1.)
MN 316       CP(III)=CPP(III)/(BETA*(CPP(III)*(XM**2))/BE)
MN 317       PHMI(III)=CP(III)*(.7*PHMI*XM*XM)+PHMI
MN 318       XMM(III)=SQRT(5.*(PHMI(III)**(-.2857)-1.))
MN 319   295 IF (LOG(III)) WRITE (JWRITE,140) XX(III),YY(III),PHMI(III),XMM(III)
MN 320      1,CP(III)
MN 321       GO TO 360
MN 322   300 XXX=(XX(L)-XX(J))/(XX(JRR)-XX(J))
MN 323       PPSRPO=PCRPSR*(PIPSR-PCRPSR)*XXX
MN 324       PPO(L)=PPSRPO*PSI(L)
MN 325   305 CONTINUE
MN 326       AK3=PPO(J)
MN 327       AK4=(PPO(JRR)-PPO(J))/(XX(JRR)-XX(J))
MN 328       CCC=0.9-(0.05*A)
MN 329       AAA=(2./(XX(J)**.5))*(AK3-AK4*XX(J)-CCC)
MN 330       BBB=AK4-(1./XX(J))*(AK3-AK4*XX(J)-CCC)
MN 331       JR=J-1
MN 332       PHMI=(1.+.2*(XM**2))**(-3.5)
MN 333       DO 310 JJ=1,JR
MN 334       PHMI(JJ)=AAA*DSQRT(XX(JJ))*BBB*XX(JJ)+CCC
MN 335       IF (PHMI(JJ).LE.0.0) GO TO 310
MN 336       XHMI(JJ)=(5.*(1./PHMI(JJ)**(.2857)-1.))
MN 337       IF (XHMI(JJ).LT.0.0) GO TO 310
MN 338       XHMI(JJ)=SQRT(XHMI(JJ))
MN 339       CP(JJ)=(1.4285/(XM*XM))*((PHMI(JJ)/PHMI)-1.)
MN 340   310 CONTINUE
MN 341       IF (IXA.LE.J) GO TO 315
MN 342       GO TO 325
MN 343   315 DO 320 JJ=1,IXA
MN 344   320 IF (LOG(JJ)) WRITE (JWRITE,140) XX(JJ),YY(JJ),PHMI(JJ),XMM(JJ),CP(J
MN 345      1J)
MN 346       GO TO 335
MN 347   325 DO 330 JJ=1,JR
MN 348   330 IF (LOG(JJ)) WRITE (JWRITE,140) XX(JJ),YY(JJ),PHMI(JJ),XMM(JJ),CP(J
MN 349      1J)
MN 350   335 DO 340 L=J,IXA
MN 351       IF (IXA.LE.J) GO TO 345
MN 352       XXX=(XX(L)-XX(J))/(XX(III)-XX(J))
MN 353       PPSRPO=PCRPSR*(PIPSR-PCRPSR)*XXX
MN 354       PPO(L)=PPSRPO*PSI(L)
MN 355       XML(L)=SQRT(5.*((1./PPO(L))**(.2857)-1.))
MN 356       CP(L)=(1.4285/(XM*XM))*((PPO(L)/PHMI)-1.)
MN 357   340 IF (LOG(L)) WRITE (JWRITE,140) XX(L),YY(L),PPO(L),XML(L),CP(L)
MN 358   345 IF (IXA.LE.J)SL=(PG(II)-PHMI(IXA))/(XX(III)-XX(IXA))
MN 359       IF (IXA.GT.J)SL=(PG(II)-PPO(IXA))/(XX(III)-XX(IXA))
MN 360       IX=IXA-1
```

437

```
      DO 350 IS=IX,II                                            MM  361
      IF (IXA.LE.J)PPO(IS)=SL*(X(IS)-XX(IXA))+PPM(IXA)           MM  362
      IF (IXA.GT.J)PPO(IS)=SL*(X(IS)-XX(IXA))+PPO(IXA)           MM  363
      XML(IS)=SQRT(5.*(1.-PPO(IS))**(.2857)-1.)                 MM  364
      CP(IS)=(1.4285/(XM*XM))*((PPO(IS)/PPM)-1.)                MM  365
  350 IF (LOG(IS)) WRITE (JWRITE,140) XX(IS),YY(IS),PPO(IS),XML(IS),CP(IM) MM  366
                                                                 MM  367
      MM=II+1                                                    MM  368
      DO 355 M=MM,100                                            MM  369
  355 IF (LOG(M)) WRITE (JWRITE,140) XX(M),YY(M),PG(M),XMLG(M),CPG(M) MM  370
      IF (XM.GE.1.0) GO TO 365                                   MM  371
  360 CONTINUE                                                   MM  372
  365 IF (MMM.EQ.1) GO TO 15                                     MM  373
      CALL EXIT                                                  MM  374
      STOP                                                       MM  375
      END                                                        MM  376
```

```
C     SUBROUTINES 'SPLINE', 'FUNCT', AND 'FIRST' CALCULATE THE SLOPE SPL   1
C     OF THE AIRFOIL SURFACE                                     SPL   2
      SUBROUTINE SPLINE(N,Y,X,AA,IKEY,YP1,YPN)                   SPL   3
      IMPLICIT REAL*8(A-H,O-Z)                                   SPL   4
      DIMENSION AA(100),Q(99),U(99),X(99),Y(99)                 SPL   5
C                                                                SPL   6
C     THIS SUBROUTINE PERFORMS A SPLINE FIT ON THE TABULATED     SPL   7
C     DATA Y VS. X.  THE SPLINE FIT PROVIDES A CURVE FIT OF THE  SPL   8
C     TABULATED DATA THAT HAS A SMOOTH FIRST DERIVATIVE.  THE    SPL   9
C     FORM OF THE CURVE FIT IS                                   SPL  10
C                                                                SPL  11
C        Y = AA(1,I)*X**3 + AA(2,I)*X**2                         SPL  12
C            + AA(3,I)*X + AA(4,I)                               SPL  13
C                                                                SPL  14
C     FOR       X(I) <= X <= X(I+1).                             SPL  15
C                                                                SPL  16
C     N    - NUMBER OF TABULATED DATA POINTS                     SPL  17
C     Y    - TABULATED FUNCTION VALUES                           SPL  18
C     X    - TABULATED ARGUMENT VALUES                           SPL  19
C     AA   - ARRAY OF COEFFICIENTS OF THE CUBIC POLYNOMIAL       SPL  20
C            SPLINE FIT.  DIMENSIONS OF AA ARE 4 BY N-1.         SPL  21
C     IKEY - CONTROL PARAMETER THAT SPECIFIES THE TYPE OF        SPL  22
C            END CONDITION TO BE USED IN THE SPLINE FIT.         SPL  23
C            IKEY = 1 -> LEAST-SQUARE END CONDITION              SPL  24
C            IKEY = 2 -> SECOND DERIVATIVE OF Y(X) SET           SPL  25
C                        TO ZERO AT END POINTS                   SPL  26
C            IKEY = 3 -> FIRST DERIVATIVE OF Y(X) IS             SPL  27
C                        SPECIFIED AT BOTH END POINTS.           SPL  28
C                        IF IKEY=3, THEN THE CALLING             SPL  29
C                        PROGRAM MUST SUPPLY VALUES              SPL  30
C                        FOR YP1 AND YPN.                        SPL  31
C     YP1  - FIRST DERIVATIVE OF Y(X) AT THE LEFT                SPL  32
C            END POINT X(1)                                      SPL  33
C     YPN  - FIRST DERIVATIVE OF Y(X) AT THE RIGHT               SPL  34
C            END POINT X(N)                                      SPL  35
C                                                                SPL  36
C     THE EQUATION NUMBERS REFERRED TO IN THE COMMENTS BELOW ARE FROM SPL 37
C     THE THEORY OF SPLINES AND THEIR APPLICATIONS               SPL  38
C                                                                SPL  39
C     CHAPTER 11, PAGES 9 - 15                                   SPL  40
C     J.H. AHLBERG, ET AL, ACADEMIC PRESS, NEW YORK , 1967       SPL  41
C                                                                SPL  42
C     SOLVE EQ.(2-1.8) USING THE ALGORITHM GIVEN BY EQ.(2-1.20)  SPL  43
C     AND EQ.(2-1.21).  GENERATE Q(1) AND U(1) OF EQ.(2-1.20) USING SPL 44
C     THE APPROPRIATE END CONDITION SPECIFIED BY IKEY.           SPL  45
C                                                                SPL  46
      NM1=N-1                                                    SPL  47
      H1=X(2)-X(1)                                               SPL  48
      GO TO (5,10,15),IKEY                                       SPL  49
C                                                                SPL  50
C     IKEY=1 -> LEAST-SQUARE LEFT END CONDITION                  SPL  51
C                                                                SPL  52
    5 Q(1)=-31.0D0/16.0D0                                        SPL  53
      H2=X(3)-X(2)                                               SPL  54
      H3=X(4)-X(3)                                               SPL  55
      U(1)=-Y(1)*(32.0D0*H1+42.0D0*H2+21.0D0*H3)/(H1*H2)/(H1+H2+H3)-Y(2)* SPL 56
     1(11.0D0*H1+42.0D0*H2+21.0D0*H3)/(H2*H3)/(H1*H2)+Y(3)*(11.0D0*H1+21.0 SPL 57
     1D0*H2+21.0D0*H3)/(H2*H3)-Y(4)*(11.0D0*H1+21.0D0*H2+H3)/(H2*H3)/(H SPL  58
     11+H2+H3)/H3                                                SPL  59
      U(1)=3.0D0*U(1)/H1/8.0D0                                   SPL  60
      GO TO 20                                                   SPL  61
C                                                                SPL  62
C     IKEY=2 -> SECOND DERIVATIVE OF Y(X) SET TO ZERO AT X(1).   SPL  63
C               THIS IMPLIES M(0)=0 OR LAMDA(0) & D(0) OF        SPL  64
C               EQ.(2-1.6) ARE ZERO.                             SPL  65
C                                                                SPL  66
   10 Q(1)=0.0D0                                                 SPL  67
      U(1)=0.0D0                                                 SPL  68
      GO TO 20                                                   SPL  69
C                                                                SPL  70
C     IKEY=3 -> FIRST DERIVATIVE OF Y(X) SET EQUAL TO YP1 AT X(1), SPL 71
C               LAMDA(0) & D(0) OF EQ.(2-1.6) DETERMINED BY EQ.(2-1.55) SPL 72
C                                                                SPL  73
   15 Q(1)=-0.500                                                SPL  74
      U(1)=3.0D0*((Y(2)-Y(1))/H1-YP1)/H1                         SPL  75
C                                                                SPL  76
C     GENERATE U(K), K=2,N-1, OF EQ.(2-1.20) USING DEFINITIONS OF SPL 77
C     LAMDA(J), MU(J), AND D(J) GIVEN BY EQ.(2-1.7)              SPL  78
C                                                                SPL  79
   20 HM=H1                                                      SPL  80
      YV=Y(2)                                                    SPL  81
      YM=Y(1)                                                    SPL  82
      DO 25 I=2,NM1                                              SPL  83
      HM=X(I+1)-X(I)                                             SPL  84
      YP=Y(I+1)                                                  SPL  85
      D=6.0D0*((YP-YV)/HM-(YV-YM)/HM)/(HM+HM)                    SPL  86
      C=HM/(HM+HM)                                               SPL  87
      A=1.0D0-C                                                  SPL  88
      P=A*Q(I-1)+2.0D0                                           SPL  89
      Q(I)=-C/P                                                  SPL  90
      U(I)=(D-A*U(I-1))/P                                        SPL  91
      HM=HM                                                      SPL  92
      YM=YV                                                      SPL  93
   25 YV=YP                                                      SPL  94
C                                                                SPL  95
C     GENERATE U(N) OF EQ.(2-1.20) USING THE APPROPRIATE END     SPL  96
C     CONDITION SPECIFIED BY IKEY                                SPL  97
C                                                                SPL  98
      GO TO (30,35,40),IKEY                                      SPL  99
C                                                                SPL 100
C     IKEY=1 -> LEAST-SQUARE RIGHT END CONDITION                 SPL 101
```

```
                                                              RETURN                                    SPL   9
                                                              END                                      SPL  10

                                                              FUNCTION LOG(J)                          FIR   1
                                                              LOGICAL LOG                              FIR   2
                                                              COMMON /INOUT/ JREAD,JWRITE,JPRINT       FIR   3
                                                              IF (J.LE.10) GO TO 5                     FIR   4
                                                              IF (J/JPRINT*JPRINT.EQ.J) GO TO 5        FIR   5
                                                              LOG=.FALSE.                              FIR   6
                                                              RETURN                                   FIR   7
                                                            5 LOG=.TRUE.                               FIR   8
                                                              RETURN                                   FIR   9
                                                              END                                      FIR  10


   30 A=31.000/16.000                                                                                  SPL 102
      P=A*Q*(N-1)*2.000                                                                                SPL 103
      H1=HH                                                                                            SPL 104
      H2=X(NM1)-X(NM1-1)                                                                               SPL 105
      H3=X(NM1-1)-X(NM1-2)                                                                             SPL 106
      D=Y(N)*(32.000*H1+42.000*H2*21.000*H3)/(H1*H2)/(H1*H2*H3)-Y(NM1)*(                               SPL 107
     111.000*H1+42.000*H2*21.000*H3)/(H2*H3)-Y(NM1-1)*(11.000*H1+H2)*(H2                               SPL 108
     111.000*(H2+H3)/(H1+H2)/(H2/H3-Y(NM1-2)*H1*(11.000*H1*21.000*H2)/(H2                              SPL 109
     1*H3)/(H1*H2*H3)/H3                                                                               SPL 110
      D=3.000*D/H1/8.000                                                                               SPL 111
      U(N)=(D-A*U(N-1))/P                                                                              SPL 112
      GO TO 45                                                                                         SPL 113
C                                                                                                      SPL 114
C     IKEY=2 -> SECOND DERIVATIVE OF Y(N) SET TO ZERO AT X(N).                                         SPL 115
C     THIS IMPLIES M(N)=0 OR M(N) & D(N) OF EQ.(2.1.6) ARE ZERO.                                       SPL 116
   35 U(N)=0.000                                                                                       SPL 117
      GO TO 45                                                                                         SPL 118
C                                                                                                      SPL 119
C     IKEY=3 -> FIRST DERIVATIVE OF Y(N) SET EQUAL TO YPN AT X(N).                                     SPL 120
C              M(N) & D(N) OF EQ.(2.1.6) DETERMINED BY EQ.(2.1.55).                                    SPL 121
   40 P=Q*(N-1)*2.000                                                                                  SPL 122
      D=6.000*(YPN-(Y(N)-Y(NM1))/HH)/HH)/HH                                                            SPL 123
      U(N)=(D-U(NM1))/P                                                                                SPL 124
C                                                                                                      SPL 125
C     SOLVE FOR THE SPLINE COEFFICIENTS CORRESPONDING TO AHLBERG'S                                     SPL 126
C     M(I) TO M(N) USING EQ.(2.1.21) AND STORE THEM IN THE U(K).                                       SPL 127
C                                                                                                      SPL 128
   45 DO 50 J=1,NM1                                                                                    SPL 129
      I=N-J                                                                                            SPL 130
      U(I)=Q(I)*U(I+1)+U(I)                                                                            SPL 131
   50 AA(I)=U(I)                                                                                       SPL 132
      AA(N)=U(N)                                                                                       SPL 133
      RETURN                                                                                           SPL 134
      END                                                                                              SPL 135


      SUBROUTINE FUNCT(AM1,AM2,X1,X2,Y1,Y2,X,FSPL)                                                     FUN   1
      REAL*8 AM1,AM2,X1,X2,Y1,Y2,X,H,T1,T2,T3,T4,U1,U2,FSPL                                            FUN   2
      H=X2-X1                                                                                          FUN   3
      T1=((X2-X)**3)*AM1/6./H                                                                          FUN   4
      T2=AM2*(X-X1)**3/6./H                                                                            FUN   5
      U1=Y1-AM1*H**2/6.                                                                                FUN   6
      U2=Y2-AM2*H**2/6.                                                                                FUN   7
      T3=U1*(X2-X)/H                                                                                   FUN   9
      T4=U2*(X-X1)/H                                                                                   FUN  10
      FSPL=T1+T2+T3+T4                                                                                 FUN  11
      RETURN                                                                                           FUN  12
      END                                                                                              FUN  12


      SUBROUTINE FIRST(AM1,AM2,X1,X2,Y1,Y2,X,DFSPL)                                                    FIR   1
      REAL*8 AM1,AM2,X1,X2,Y1,Y2,X,H,T1,T2,T3,T4,DFSPL                                                 FIR   2
      H=X2-X1                                                                                          FIR   3
      T1=-AM1*((X2-X)**2)/2./H                                                                         FIR   4
      T2=AM2*(X-X1)**2/2./H                                                                            FIR   5
      T3=(Y2-Y1)/H                                                                                     FIR   6
      T4=(AM1-AM2)*H/6.                                                                                FIR   7
      DFSPL=T1+T2+T3+T4                                                                                FIR   8
```

439

Sample Output - COMPR

ANGLE OF ATTACK= 2.50 DEGREES

UPPER SURFACE

THE AIRFOIL COORDINATES AND INCOMPRESSIBLE PRESSURE DISTRIBUTION

X/C	Y/C	CPI
0.01C000	C.012800	-0.938000
0.02C000	C.018000	-0.932000
0.03C000	C.021800	-0.910000
0.04C000	C.024900	-0.840000
0.05C000	0.027500	-0.796000
0.06C000	0.029500	-0.762000
0.07C000	0.031600	-0.732000
0.080000	0.033500	-0.705000
0.09C000	0.035000	-0.690000
0.10C000	0.036900	-0.672000
0.15C000	C.042700	-0.602000
0.200000	C.046400	-0.541000
0.25C000	C.048700	-0.485000
0.30C000	0.049800	-0.437000
0.35C000	C.049900	-0.395000
0.40C000	C.049200	-0.360000
0.45C000	C.047900	-0.332000
0.50C000	C.049900	-0.301000
0.55C000	C.043300	-0.271000
0.60C000	0.040200	-0.237000
0.65C000	0.036600	-0.202000
0.700000	0.032600	-0.165000
0.75C000	C.028100	-0.127000
0.8CC000	C.023200	-0.085000
0.85C000	C.017900	-0.039000
0.9CC000	C.012300	0.017000
0.95C000	0.006300	0.095000
1.00C000	0.CC0000	1.CC0000

THE CRITICAL MACH NUMBER= 0.597

XX	YY	DIV	THETA
0.010000	0.012800	0.548000	C.501306
C.020000	0.018000	C.455124	0.427108
0.030000	0.021800	C.331498	0.320097
0.040000	0.0249C0	0.289884	0.281278
0.050000	0.0275C0	0.222965	0.219377
C.060000	0.0295C0	0.159254	0.196678
0.070000	0.0316C0	0.210018	0.2C7009

0.080000	0.033500	0.140673	0.199312
0.090000	0.035000	0.147288	C.165753
0.100000	0.036900	0.140175	0.187951
0.150000	0.0427C0	0.078461	0.079294
0.200000	0.0444C0	0.061981	0.061907
C.250000	0.0487CC	0.032613	0.032602
0.300000	0.0498CC	0.011565	0.011565
0.350000	0.0499C0	-0.006074	-C.C06074
0.4C0000	0.0492C0	-0.020068	-C.C20066
C.450000	0.0479C0	-0.032852	-C.032841
0.5C0000	0.0499C0	-0.046622	-0.046689
0.550000	0.043300	-0.057058	-0.056996
0.600000	0.0402C0	-0.067246	-C.C67145
0.650000	0.036600	-0.075959	-0.075813
0.700000	0.032600	-0.084918	-0.084715
0.750000	0.078100	-0.094369	-0.094090
0.8C0000	0.0232C0	-0.101607	-C.101260
0.850000	0.0179C0	-0.111202	-0.110747
0.900000	0.012300	-0.107986	-0.107174
0.950000	0.006300	-0.154454	-0.153243
1.000000	-0.000000	-0.012599	-0.012598

CALCULATION OF THE LOCAL MACH NUMBER NEAR THE NOSE FOR M=1

X/C	Y/C	P/PO	P-LOCAL	CP
0.01000	0.01280	0.25393	1.54815	-0.74186

LOCAL MACH NUMBER HAS EXCEEDED UNITY BEFORE 1 PERCENT CHORD

LAMBDA= 1.391757

PRESSURE DISTRIBUTION BETWEEN SONIC POINT AND TRAILING EDGE FOR M=1

X/C	Y/C	P/PO	P-LOCAL	CP
0.01000	0.01280	0.25417	1.54756	-0.74121
0.02000	0.01800	0.38664	1.24888	-0.38302
C.03000	0.02180	0.36956	1.28294	-0.42919
0.04000	0.0249C	0.38448	1.29307	-0.38885
0.05000	0.02750	0.39685	1.30830	-0.46395
0.06000	0.02950	0.36071	1.30041	-0.49312
0.07000	0.03160	0.39474	1.23422	-0.36246
0.08000	0.03350	0.36090	1.300C2	-0.45260
0.09000	0.03500	0.38248	1.25698	-0.39426
0.10000	0.0369C	0.42191	1.18241	-0.28764
0.15000	0.0427C	0.36533	1.33226	-0.49472
0.20000	0.04640	0.35188	1.31858	-0.47701
0.25000	0.04670	0.34002	1.34349	-0.50906
C.30000	0.04980	0.33340	1.35769	-0.52698
0.35000	0.0499C	0.32706	1.37147	-0.54411
0.40000	0.04920	0.32451	1.17708	-0.55101
0.45000	0.0479C	0.32068	1.38556	-0.56136
0.50000	0.04990	0.31467	1.39905	-0.57761
0.55000	0.04330	0.31082	1.40777	-0.58803
0.A0000	0.04020	0.30673	1.41713	-0.59908
C.65000	0.0366C	0.30327	1.42515	-0.60846
0.70000	0.0326C	0.29873	1.43576	-0.62072
0.75000	0.02810	0.29419	1.44669	-0.63298
0.80000	0.02320	0.29121	1.45383	-0.64107
0.85000	0.01790	0.28542	1.46763	-0.65671
0.90000	0.0123C	0.29221	1.45121	-0.63834
0.95000	0.00630	0.29765	1.53826	-0.73180
1.00000	-0.0C0C0	0.28346	1.255C6	-0.39160

MACH NUMBER= 0.978CC

X/C	Y/C	P/PO	P-LOCAL	CP
0.01000	0.01280	0.55772	0.95273	-1.28536
0.02000	C.D18CC	0.55943	0.95002	-1.27617
0.03000	0.02180	0.56569	0.94010	-1.24260
0.04000	0.02490	0.58538	0.90915	-1.13702
0.05000	C.02750	0.59758	0.89012	-1.07159
0.06000	0.0295C	0.60692	0.87569	-1.02152
0.07000	0.0316C	0.61510	0.86297	-0.97769
0.08000	0.03350	0.62240	0.85169	-0.93951
0.09000	0.03500	0.62644	0.84546	-0.91686
0.10000	0.03690	0.63180	0.83720	-0.88811
0.15000	0.0427C	0.64983	0.80948	-0.79141
0.20000	0.04640	0.66576	0.78503	-0.70601
0.25000	C.0487C	0.68018	0.76289	-0.62871
0.30000	0.04980	0.69238	0.74412	-0.56326
0.35000	0.0499C	C.70295	0.72783	-0.50660
0.40000	0.04920	0.71167	0.71434	-0.45982
0.45000	0.04790	0.71860	0.70360	-0.47266
0.50000	0.04590	0.72622	0.69179	-0.38181
0.55000	0.04330	0.73354	0.68032	-0.34256
0.60000	0.04020	0.74177	0.66742	-0.29840
0.65000	0.03660	0.75018	0.65417	-0.25331
0.70000	0.0326C	0.75900	0.64020	-0.20603
0.75000	0.02810	0.76797	0.62597	-0.15789
0.80000	0.0232C	0.77781	0.61004	-0.10917
0.85000	0.01790	0.78847	0.59270	-0.04800
0.90000	0.01230	0.80130	0.57156	0.02079
0.95000	0.00630	0.81890	0.54195	0.11518

MACH NUMBER= 0.95300

SHOCK LOCATED AT 93 PERCENT CHORD FOR MACH NUMBER= 0.953

LAMINAR BOUNDARY LAYER--REYNOLDS NUMBER= 0.100E 07

X/C	Y/C	P/PC	P-LOCAL	CP
0.01000	0.01280	0.62623	0.84578	0.19369
0.02000	0.018CC	0.57370	0.92749	0.04549
0.03000	0.02180	0.53699	0.98589	-0.05805
C.C4000	C.0249C	C.50848	1.03237	-0.13848
0.05000	0.02750	0.48521	1.07122	-0.20413
C.06000	0.02950	0.46566	1.10462	-0.25928
0.07000	0.03160	0.44892	1.13382	-0.30669
0.08CC0	0.03350	0.43442	1.15966	-0.34739
0.09000	0.035CC	0.42175	1.18247	-0.38315
0.10000	0.03690	0.41060	1.20328	-0.41461
0.15000	C.04270	C.37138	1.27865	-C.52522
0.20000	0.04640	0.35050	1.32140	-0.58414
0.25C00	0.0487C	0.34115	1.34104	-0.61050
0.30000	0.04980	0.33503	1.35413	-0.62778
0.35000	0.0499C	0.32920	1.36677	-0.64423
0.40000	0.04920	0.32714	1.37125	-0.65002
0.45000	0.04790	0.32302	1.37856	-0.65940
0.50000	0.04590	0.31831	1.39081	-0.67494
0.55C00	C.0433C	C.31496	1.39834	-0.68439
C.60000	0.04020	C.31873	1.38987	-0.67376
0.65000	0.03660	0.33454	1.31303	-0.57274
0.70000	C.0326C	C.39035	1.24166	-0.47172
0.75000	0.02810	0.42616	1.17661	-0.37070
0.80000	0.0232C	0.44197	1.111C0	-0.76968
0.85000	0.01790	0.51194	1.07667	-0.12072
0.90C00	C.0123C	0.51746	0.92157	0.05611
0.95000	0.00630	0.66072	0.78048	0.31357
1.C0000	-0.0C0C0	1.72757UUUUULLLLLLCUULU	0.78048	3.30066

Figure H-8. Sample output for COMPR.

ANGLE OF ATTACK= 2.50 DEGREES

LOWER SURFACE

THE AIRFOIL COORDINATES AND INCOMPRESSIBLE PRESSURE DISTRIBUTION

X/C	Y/C	CPI
0.010000	0.011300	0.505000
0.020000	0.017100	0.400000
0.030000	0.020800	0.310000
0.040000	0.024500	0.241000
0.050000	0.027500	0.180000
0.060000	0.029500	0.143000
0.070000	0.031600	0.108000
0.080000	0.033500	0.078000
0.090000	0.035500	0.050000
0.100000	0.036900	0.030000
0.150000	0.042700	-0.050000
0.200000	0.046400	-0.094000
0.250000	0.048700	-0.111000
0.300000	0.049800	-0.111000
0.350000	0.049900	-0.111000
0.400000	0.049200	-0.110000
0.450000	0.047900	-0.104000
0.500000	0.045900	-0.098000
0.550000	0.043300	-0.090000
0.600000	0.040200	-0.080000
0.650000	0.036600	-0.068000
0.700000	0.032600	-0.050000
0.750000	0.028100	-0.030000
0.800000	0.023200	-0.004000
0.850000	0.017900	0.026000
0.900000	0.012300	0.068000
0.950000	0.006300	0.129000
1.000000	0.000000	1.000000

THE CRITICAL MACH NUMBER= 0.898

XX	YY	DIV	THETA
0.010000	0.011300	0.508000	0.470027
0.020000	0.017100	0.500419	0.463989
0.030000	0.020800	0.340323	0.328028
0.040000	0.024500	0.358284	0.344040
0.050000	0.027500	0.236522	0.232254
0.060000	0.029500	0.195622	0.193192
0.070000	0.031600	0.210992	0.207942
0.080000	0.033500	0.160412	0.159057
0.090000	0.035500	0.167361	0.165924
0.100000	0.036900	0.190144	0.187901
0.150000	0.042700	0.079469	0.079101
0.200000	0.046400	0.061979	0.061900
0.250000	0.048700	0.037614	0.032602
0.300000	0.049800	0.011565	0.011565
0.350000	0.049900	-0.006874	-0.006874
0.400000	0.049200	-0.020068	-0.020068
0.450000	0.047900	-0.032952	-0.032940
0.500000	0.045900	-0.046523	-0.046489
0.550000	0.043300	-0.057057	-0.056995
0.600000	0.040200	-0.067249	-0.067148
0.650000	0.036600	-0.075948	-0.075802
0.700000	0.032600	-0.084960	-0.084756
0.750000	0.028100	-0.094212	-0.093935
0.800000	0.023200	-0.102192	-0.101838
0.850000	0.017900	-0.109020	-0.108591
0.900000	0.012300	-0.115728	-0.115215
0.950000	0.006300	-0.124068	-0.123437
1.000000	-0.000000	-0.126000	-0.125339

CALCULATION OF THE LOCAL MACH NUMBER NEAR THE NOSE FOR M=1

X/C	Y/C	P/PO	M-LOCAL	CP
0.01000	0.01130	0.73906	0.67168	0.56996
0.02000	0.01710	0.69072	0.74667	0.43925
0.03000	0.02080	0.65112	0.80750	0.33217
0.04000	0.02450	0.62188	0.85249	0.25511
0.05000	0.02750	0.59682	0.89131	0.18535
0.06000	0.02950	0.58198	0.91448	0.14521
0.07000	0.03160	0.56819	0.93617	0.10791
0.08000	0.03350	0.55651	0.95460	0.07644
0.09000	0.03500	0.54584	0.97168	0.04748
0.10000	0.03690	0.53828	0.98381	0.02705

LOCAL MACH NUMBER HAS EXCEEDED UNITY BEFORE 11 PERCENT CHORD

LAMBDA= 1.458002

PRESSURE DISTRIBUTION BETWEEN SONIC POINT AND TRAILING EDGE FOR M=1

X/C	Y/C	P/PO	M-LOCAL	CP
0.15000	0.04270	0.41597	1.15691	-0.24962
0.20000	0.04640	0.44715	1.13700	-0.21939
0.25000	0.04870	0.43114	1.16562	-0.26269
0.30000	0.04980	0.42296	1.18049	-0.28481
0.35000	0.04990	0.41522	1.19472	-0.30572
0.40000	0.04920	0.41246	1.19985	-0.31319
0.45000	0.04790	0.40795	1.20827	-0.32539
0.50000	0.04590	0.40094	1.22222	-0.34542
0.55000	0.04330	0.39589	1.23107	-0.35800
0.60000	0.04020	0.39089	1.24066	-0.37152
0.65000	0.03680	0.38664	1.24886	-0.38300
0.70000	0.03260	0.38087	1.26014	-0.39862
0.75000	0.02810	0.37534	1.27103	-0.41956
0.80000	0.02320	0.37080	1.28005	-0.42583
0.85000	0.01790	0.36614	1.28942	-0.43845
0.90000	0.01230	0.36461	1.29290	-0.44258
0.95000	0.00630	0.35670	1.30863	-0.46397
1.00000	-0.00000	0.35508	1.31196	-0.46836

MACH NUMBER= 0.57800

X/C	Y/C	P/PO	M-LOCAL	CP
0.01000	0.01130	0.90661	0.37688	0.58552
0.02000	0.01710	0.88484	0.42165	0.46903
0.03000	0.02080	0.86587	0.45828	0.38706
0.04000	0.02450	0.85104	0.48560	0.28752
0.05000	0.02750	0.83773	0.50934	0.21419
0.06000	0.02950	0.82958	0.52359	0.17244
0.07000	0.03160	0.82180	0.53699	0.13074
0.08000	0.03350	0.81509	0.54842	0.03475
0.09000	0.03500	0.80878	0.55906	0.06093
0.10000	0.03690	0.80425	0.56663	0.03664
0.15000	0.04270	0.78593	0.59685	-0.06162
0.20000	0.04640	0.77571	0.61343	-0.11442
0.25000	0.04870	0.77197	0.61946	-0.13649
0.30000	0.04980	0.77173	0.61984	-0.13775
0.35000	0.04990	0.77173	0.61984	-0.13649
0.40000	0.04920	0.77197	0.61946	-0.12896
0.45000	0.04790	0.77337	0.61720	-0.12141
0.50000	0.04590	0.77477	0.61494	-0.11142
0.55000	0.04330	0.77664	0.61192	-0.09893
0.60000	0.04020	0.77897	0.60816	-0.08397
0.65000	0.03660	0.78176	0.60363	-0.06162
0.70000	0.03260	0.78593	0.59685	-0.03689
0.75000	0.02810	0.79054	0.58931	-0.00490
0.80000	0.02320	0.79652	0.57949	0.03177
0.85000	0.01790	0.80334	0.56815	0.08270
0.90000	0.01230	0.81284	0.55272	0.15582
0.95000	0.00630	0.82648	0.52896	

MACH NUMBER= 0.95100

SHOCK LOCATED AT 76 PERCENT CHORD FOR MACH NUMBER= 0.951

LAMINAR BOUNDARY LAYER--REYNOLDS NUMBER= 0.100E 07

X/C	Y/C	P/PO	M-LOCAL	CP
0.01000	0.01130	0.82427	0.53279	0.75236
0.02000	0.01710	0.75334	0.64918	0.55225
0.03000	0.02080	0.70375	0.72658	0.41237
0.04000	0.02450	0.66522	0.78585	0.30368
0.05000	0.02750	0.61376	0.83619	0.21491
0.06000	0.02950	0.60731	0.87503	0.14030
0.07000	0.03160	0.58466	0.91029	0.07641
0.08000	0.03350	0.56502	0.94117	0.02102
0.09000	0.03500	0.54784	0.96847	-0.02744
0.10000	0.03690	0.53273	0.99277	-0.07009
0.15000	0.04270	0.47944	1.08100	-0.22039
0.20000	0.04640	0.45088	1.13038	-0.30098
0.25000	0.04870	0.43787	1.15346	-0.33766
0.30000	0.04980	0.42926	1.16897	-0.36195
0.35000	0.04990	0.42110	1.18386	-0.38498
0.40000	0.04920	0.41790	1.18973	-0.39399
0.45000	0.04790	0.41296	1.19888	-0.40794
0.50000	0.04590	0.40512	1.21355	-0.43004
0.55000	0.04330	0.41233	1.20006	-0.40973
0.60000	0.04020	0.44002	1.14962	-0.33160
0.65000	0.03660	0.46772	1.10101	-0.25367
0.70000	0.03260	0.49541	1.05407	-0.17534
0.75000	0.02810	0.52311	1.00838	-0.09721
0.80000	0.02320	0.55289	0.96042	-0.01320
0.85000	0.01790	0.58799	0.90508	0.08982
0.90000	0.01230	0.63713	0.82900	0.22444
0.95000	0.00630	0.70850	0.71924	0.42579
1.00000	-0.00000	1.72757UUUUUUUUULLLLLLLU		3.30066

Figure H-8. Continued.

This program is written in FORTRAN IV and is designed to run in single precision on a IBM 370-165 computer. Execution requires 42,000 bytes of core storage and approximately 6 seconds to produce the two-dimensional zero Mach number pressure distribution for an airfoil at six angles of attack by the 32-point Weber method. For a given run the program requires the following input data:

(1) The 992 elements of the Matrix S1 given in A.R.C. R&M 2918 and listed herein as Table H-5.

(2) The 992 elements of the Matrix S2 also given in A.R.C. R&M 2918 and listed herein as Table H-6.

(3) The 1024 elements of Matrix S3 also given in A.R.C. R&M 2918 and listed herein as Table H-7.

(4) The 72 characters of the array TITLE which are used as a header for identifying output. It is an alphameric array and usually contains information about the type of airfoil, run numbers, and any other such information desired by the programmer as a header..

(5) The values of X, from Table H-1.

(6) The values of Y, the y-coordinates (given in fraction of chord) of the airfoil to be considered at the previously specified values of X.

(7) The values of angle of attack, A, and airfoil leading edge radius, RHO, for a given airfoil, followed by a last card indicator giving A a value of greater than 15.

(8) Additional data sets containing information from item (4) through item (7), if needed. Only two airfoils may be considered per run.

Format specifications for these variables may be found in Figure H-9. A sample data set of the NCSU 0010 may be found in Figure H-10 and the output for that data set is given in Figure H-11.

442

Figure H-9. Format specification of input data for TRANSON.

Figure H-10. Example data set for TRANSON.

444

Program Listing - TRANSON

```
C     PROGRAM TO CALCULATE THE INCOMPRESSIBLE PRESSURE DISTRIBUTION          MN    1
C     ABOUT 2-D SYMMETRICAL AIRFOILS (REFERENCE- WEBER,J.- A.R.C. R.CM.       MN    2
C     2918, JULY, 1953.)                                                      MN    3
C     THE INPUT DATA NEEDED ARE AS FOLLOWS :                                  MN    4
C                                                                             MN    5
C     S1(I,J),S2(I,J),E S3(I,J) ARE COEFFICIENTS LISTED IN R.CM. 2918         MN    6
C     FOR N=32                                                                MN    7
C     TITLE= PERTINENT INFORMATION TO BE LISTED AT FIRST OF OUTPUT            MN    8
C     X(I) E Y(I) ARE THE AIRFOIL ABSCISSA AND ORDINATES (X/C E Y/C)          MN    9
C       THESE ARE INPUT FROM THE TRAILING EDGE FORWARD FOR THE                MN   10
C       SPECIFIC LOCATIONS FOR N = 32                                         MN   11
C     A= AIRFOIL ANGLE OF ATTACK (DEGREES)                                    MN   12
C     RHO= AIRFOIL LEADING EDGE RADIUS (DECIMAL FRACTION OF CHORD)            MN   13
C                                                                             MN   14
      DIMENSION S1(32,32),S2(32,32),S3(32,32),X(32),Y(32),S11(32),S22(32     MN   15
     1,S33(32),CPU(32),CPL(32)                                               MN   16
      DIMENSION TITLE(18)                                                     MN   17
      COMMON /INOUT/JREAD,JWRITE                                             MN   18
      JREAD=1                                                                 MN   19
      JWRITE=3                                                                MN   20
      READ (JREAD,4) ((S1(I,J),J=1,32),I=1,31)                               MN   21
      READ (JREAD,4) ((S2(I,J),J=1,31),I=1,31)                               MN   22
      READ (JREAD,4) ((S3(I,J),J=1,32),I=1,32)                               MN   23
    4 FORMAT (8F10.5)                                                         MN   24
      READ (JREAD,101) (TITLE(I),I=1,18)                                      MN   25
  101 FORMAT (18A4)                                                           MN   26
      DO 20 JJ=1,2                                                            MN   27
      READ (JREAD,5) (X(I),I=1,32)                                            MN   28
      READ (JREAD,5) (Y(I),I=1,32)                                            MN   29
    5 FORMAT (8F10.6)                                                         MN   30
      READ (JREAD,6) A,RHO                                                    MN   31
    6 FORMAT (2F10.5)                                                         MN   32
      IF (A.GT.15.) GO TO 19                                                  MN   33
      Y(32)=SQRT(RHO/2.)                                                      MN   34
      WRITE (JWRITE,39)                                                       MN   35
   39 FORMAT (1H1,///,19X,35HINCOMPRESSIBLE PRESSURE COEFFICIENT)             MN   36
      WRITE (JWRITE,32)                                                       MN   37
      WRITE (JWRITE,13) (TITLE(I),I=1,18)                                     MN   38
   13 FORMAT (///,18A4)                                                       MN   39
      WRITE (JWRITE,14) A                                                     MN   40
   14 FORMAT (///,24X,17HANGLE OF ATTACK= ,F10.6)                             MN   41
      IF (JJ.EQ.2) GO TO 91                                                   MN   42
      WRITE (JWRITE,12)                                                       MN   43
   12 FORMAT (///,15X,3HX/C,17X,3HY/C,12X,8HCP-UPPER,//)                      MN   44
      GO TO 90                                                                MN   45
   91 WRITE (JWRITE,31)                                                       MN   46
   31 FORMAT (///,15X,3HX/C,17X,3HY/C,12X,8HCP-LOWER,//)                      MN   47
   90 DO 100 J=1,31                                                           MN   48
      S11(J)=0.                                                               MN   49
      S22(J)=0.                                                               MN   50
      S33(J)=S3(32,J)*Y(32)                                                   MN   51
      DO 8 I=1,31                                                             MN   52
      S11(J)=S11(J)+S1(I,J)*Y(I)                                             MN   53
    8 CONTINUE                                                                MN   54
      DO 9 I=1,31                                                             MN   55
      S22(J)=S22(J)+S2(I,J)*Y(I)                                             MN   56
    9 CONTINUE                                                                MN   57
      DO 10 I=1,31                                                            MN   58
      S33(J)=S33(J)+S3(I,J)*Y(I)                                             MN   59
   10 CONTINUE                                                                MN   60
      B=COS(A/57.2958)*(1.+S11(J))                                            MN   61
      C=SIN(A/57.2958)*SQRT((1.-X(J))/X(J))*(1.+S33(J))                       MN   62
      D=1.+(S22(J))**2                                                        MN   63
      IF (JJ.EQ.2) GO TO 22                                                   MN   64
      CPU(J)=1.-((B+C)**2)/D)                                                 MN   65
      GO TO 107                                                               MN   66
   22 CPU(J)=1.-((B-C)**2)/D)                                                 MN   67
  107 WRITE (JWRITE,11) X(J),Y(J),CPU(J)                                      MN   68
   11 FORMAT (3F20.6)                                                         MN   69
  100 CONTINUE                                                                MN   70
      WRITE (JWRITE,32)                                                       MN   71
   32 FORMAT (///,72H**********************************************           MN   72
     1****************)                                                       MN   73
      GO TO 7                                                                 MN   74
   19 IF (JJ.EQ.2.AND.A.GT.15.) CALL EXIT                                     MN   75
   20 CONTINUE                                                                MN   76
      STOP                                                                    MN   77
      END                                                                     MN   78
```

445

Sample Output - TRANSON

INCOMPRESSIBLE PRESSURE COEFFICIENT

●●●

NCSU 0010   32 POINT

ANGLE OF ATTACK= 1.000000

X/C	Y/C	CP-UPPER
0.997600	0.000300	0.338278
0.990400	0.001200	0.256765
0.978500	0.002800	0.160352
0.961900	0.004800	0.131678
0.941000	0.007400	0.080033
0.915700	0.010400	0.047191
0.886500	0.013800	0.010094
0.853600	0.017500	-0.010339
0.817200	0.021500	-0.062837
0.777800	0.025400	-0.080945
0.735700	0.029400	-0.112421
0.691300	0.033300	-0.139244
0.645100	0.037030	-0.169633
0.597500	0.040410	-0.191240
0.549000	0.043490	-0.225570
0.500000	0.045920	-0.241208
0.451000	0.047890	-0.267856
0.402500	0.049240	-0.290399
0.354900	0.049940	-0.315150
0.308700	0.049890	-0.354026
0.264300	0.049140	-0.357075
0.222200	0.047640	-0.379512
0.182800	0.045350	-0.394442
0.146400	0.042400	-0.417375
0.113500	0.038740	-0.439997
0.084270	0.034220	-0.407974
0.059040	0.029680	-0.475940
0.038060	0.024640	-0.523752
0.021530	0.018600	-0.421792
0.009610	0.012400	-0.390279
0.002410	0.004900	0.831975

ANGLE OF ATTACK= 2.500000

X/C	Y/C	CP-UPPER
0.997600	0.000300	0.337996
0.990400	0.001200	0.254671
0.978500	0.002800	0.155970
0.961900	0.004800	0.124953
0.941000	0.007400	0.070941
0.915700	0.010400	0.034455
0.886500	0.013800	-0.005982
0.853600	0.017500	-0.037192
0.817200	0.021500	-0.085629
0.777800	0.025400	-0.107562
0.735700	0.029400	-0.143370
0.691300	0.033300	-0.174986
0.645100	0.037030	-0.210584
0.597500	0.040410	-0.237021
0.549000	0.043490	-0.278661
0.500000	0.045920	-0.301007
0.451000	0.047890	-0.339332
0.402500	0.049240	-0.366810
0.354900	0.049940	-0.401311
0.308700	0.049890	-0.432605
0.264300	0.049140	-0.467677
0.222200	0.047640	-0.506651
0.182800	0.045350	-0.539010
0.146400	0.042400	-0.586824
0.113500	0.038740	-0.637625
0.084270	0.034220	-0.658991
0.059040	0.029680	-0.762879
0.038060	0.024640	-0.909561
0.021530	0.018600	-0.884527
0.009610	0.012400	-1.150023
0.002410	0.004900	0.719461

INCOMPRESSIBLE PRESSURE COEFFICIENT

●●●

NCSU 0010   32 POINT

ANGLE OF ATTACK= 1.000000

X/C	Y/C	CP-LOWER
0.997600	0.000300	0.340111
0.990400	0.001200	0.261133
0.978500	0.002800	0.167949
0.961900	0.004800	0.142444
0.941000	0.007400	0.094434
0.915700	0.010400	0.065578
0.886500	0.013800	0.032676
0.853600	0.017500	0.005366
0.817200	0.021500	-0.019994
0.777800	0.025400	-0.043726
0.735700	0.029400	-0.069648
0.691300	0.033300	-0.090072
0.645100	0.037030	-0.113703
0.597500	0.040410	-0.128077
0.549000	0.043490	-0.154009
0.500000	0.045920	-0.161114
0.451000	0.047890	-0.178005
0.402500	0.049240	-0.184299
0.354900	0.049940	-0.201146
0.308700	0.049890	-0.207135
0.264300	0.049140	-0.213676
0.222200	0.047640	-0.213991
0.182800	0.045350	-0.210059
0.146400	0.042400	-0.203550
0.113500	0.038740	-0.193477
0.084270	0.034220	-0.124579
0.059040	0.029680	-0.129434
0.038060	0.024640	-0.076299
0.021530	0.018600	0.044618
0.009610	0.012400	0.360053
0.002410	0.004900	0.939879

ANGLE OF ATTACK= 2.500000

X/C	Y/C	CP-LOWER
0.997600	0.000300	0.342522
0.990400	0.001200	0.265579
0.978500	0.002800	0.174967
0.961900	0.004800	0.151042
0.941000	0.007400	0.104705
0.915700	0.010400	0.080448
0.886500	0.013800	0.050890
0.853600	0.017500	0.030251
0.817200	0.021500	-0.005108
0.777800	0.025400	-0.014587
0.735700	0.029400	-0.036147
0.691300	0.033300	-0.070901
0.645100	0.037030	-0.080094
0.597500	0.040410	-0.099932
0.549000	0.043490	-0.100483
0.500000	0.045920	-0.110948
0.451000	0.047890	-0.114331
0.402500	0.049240	-0.113361
0.354900	0.049940	-0.113715
0.308700	0.049890	-0.104631
0.264300	0.049140	-0.099256
0.222200	0.047640	-0.078497
0.182800	0.045350	-0.057839
0.146400	0.047460	-0.021984
0.113500	0.038740	0.068742
0.084270	0.034220	0.101465
0.059040	0.029680	0.212848
0.038060	0.024640	0.409517
0.021530	0.018600	0.743783
0.009610	0.012400	0.905260
0.002410	0.004900	0.905260

Figure H-11.   Sample output for TRANSON.

446

APPENDIX I - A Computer Program Providing Rapid Evaluation

of Closed-Form Solution for the Flow About a

Prolate Spheroid Moving Along Its X-Axis

In order to evaluate the accuracy of the method described in the body of this work for predicting potential flow about an arbitrary three-dimensional body, a search was undertaken for a suitable body with a closed-form velocity distribution. The most obvious was the sphere. While the sphere does provide a means for such comparison, it is a very crude approximation of an aircraft fuselage or similar type three-dimensional body. A body which had a higher length-to-width ratio would be more representative; therefore, an analytical method for generating the potential flow about an ellipsoid was examined.

The solution for potential flow about an ellipsoid is well known and can be found in Lamb (Ref. 6) and Munk (Ref. 66). The solution and formulae for generating the velocity distributions are also given in Timman (Ref. 49). The equations in the latter work, originally developed for a yawed ellipsoid, were degenerated to apply to the case of a spheroid (ellipsoid with y and z axes equal) in a uniform flow field of velocity U_o at zero incidence to the x-axis of the spheroid.

The velocity distribution of the spheroid was calculated by a computer program in the following manner:

Knowing the x-axis (a), the y and z-axes (b) of the spheroid, and the x, y, and z coordinates of the point for which the potential flow is to be evaluated, a series of coefficients is then determined:

$$S = \sqrt{1 - b^2/a^2}$$

$$\alpha_o = \frac{2\ ab^2}{(a^2 - b^2)^{3/2}} \left[\ \tanh^{-1}(S) - S\right]$$

$$P = \frac{2\ U_o}{(2 - \alpha_o)}$$

$$G = \frac{x^2}{a^4} + \frac{1}{b^4}(y^2 + z^2)$$

447

Following Timman, one may manipulate the degenerate form of the equations to obtain the velocity components in terms of these coefficients:

$$u = \frac{-P}{Gb^4} (y^2 + z^2)$$

$$v = \frac{P}{G} (\frac{xy}{a^2 b^2})$$

$$w = \frac{P}{G} (\frac{xz}{a^2 b^2})$$

Given the dimensions of the spheroid and the free stream velocity, the program then computes u, v, and w at a specified x, y, and z on the surface of the spheroid by means of the relations above.

From the equations it can be seen that the potential flow about a sphere may also be generated by letting a be equal to b. Because of the indeterminant form of the solution for this case, it is necessary to obtain a solution for a sphere by letting a = 1.01b.

Because of its speed and accuracy this program was most helpful in establishing when the general three-dimensional body potential flow program was running correctly and how many panels were required and in what distribution to represent a body adequately. It is presented here so that the reader may use it for that purpose or as a direct solution for a special class of fuselages. The latter case may be of special interest to those with limited computing facilities or those who wish to demonstrate the design concepts related herein without incurring the large computing expenses of the general three-dimensional body program. The following discussion provides a listing, instructions for entering data, and typical output.

448

Sample Output

3-1 SPHEROID

THE THEORETICAL VELOCITY SOLUTION FOR A SPHEROID

A= 3.00000
B= 1.00000
UO= 1.00000

I	X	Y	Z	U	V	W	CP
1	3.00000	0.00000	0.00000	-0.00000	0.00000	0.00000	1.00000
2	2.50000	0.00000	0.55177	-0.89511	0.00000	0.45062	-0.00428
3	2.00000	0.00000	0.74417	-1.03011	0.00000	0.30761	-0.15575
4	1.50000	0.00000	0.86603	-1.08190	0.00000	0.20021	-0.21305
5	1.00000	0.00000	0.94258	-1.10659	0.00000	0.13044	-0.24156
6	0.50000	0.00000	0.98613	-1.11842	0.00000	0.06301	-0.25403
7	0.00000	0.00000	1.00000	-1.12197	0.00000	0.00000	-0.25801

SPHERE

THE THEORETICAL VELOCITY SOLUTION FOR A SPHEROID

A= 1.00100
B= 1.00000
UO= 1.00000

I	X	Y	Z	U	V	W	CP
1	1.00100	0.00000	0.00104	-0.00000	0.00000	0.00155	1.00000
2	1.00000	0.00000	0.04472	-0.00301	0.00000	0.06706	0.99549
3	0.99900	0.00000	0.06322	-0.00601	0.00000	0.09470	0.99100
4	0.99800	0.00000	0.07741	-0.00900	0.00000	0.11584	0.98650
5	0.99700	0.00000	0.08936	-0.01200	0.00000	0.13358	0.98201
6	0.99599	0.00000	0.09988	-0.01499	0.00000	0.14916	0.97753
7	0.99400	0.00000	0.10939	-0.01798	0.00000	0.16319	0.97305
8	0.99399	0.00000	0.11812	-0.02096	0.00000	0.17604	0.96857
9	0.99299	0.00000	0.12625	-0.02394	0.00000	0.18796	0.96410
10	0.99199	0.00000	0.13387	-0.02692	0.00000	0.19911	0.95963
⋮			⋮				⋮
990	0.01097	0.00000	0.99994	-1.49922	0.00000	0.01641	-1.24704
991	0.00997	0.00000	0.99995	-1.49926	0.00000	0.01491	-1.24709
992	0.00896	0.00000	0.99996	-1.49928	0.00000	0.01341	-1.24803
993	0.00796	0.00000	0.99997	-1.49931	0.00000	0.01192	-1.24807
994	0.00696	0.00000	0.99998	-1.49933	0.00000	0.01042	-1.24810
995	0.00596	0.00000	0.99998	-1.49935	0.00000	0.00892	-1.24813
996	0.00496	0.00000	0.99999	-1.49937	0.00000	0.00742	-1.24815
997	0.00396	0.00000	0.99999	-1.49938	0.00000	0.00592	-1.24817
998	0.00296	0.00000	1.00000	-1.49939	0.00000	0.00443	-1.24819
999	0.00196	0.00000	1.00000	-1.49940	0.00000	0.00293	-1.24820
1000	0.00096	0.00000	1.00000	-1.49940	0.00000	0.00143	-1.24821

Figure I-3. Sample output for SPHEROID.

Program Listing

```
C
C
C
C          THE POTENTIAL FLOW ABOUT A SPHEROID AS DERIVED FROM:
C
C          THE POTENTIAL FLOW ABOUT A YAWED ELLIPSOID AT ZERO INCIDENCE
C          BY: R. TIMMAN - NATIONAL AEROSPACE LABORATORY - THE NETHERLANDS
C
C
           REAL*8 S
           DATA END/'END '/
           DIMENSION TITLE(20)
        20 READ(1,120) TITLE
       120 FORMAT(20A4)
           IF(TITLE(1).EQ.END ) GO TO 30
           WRITE(3,125) TITLE
       125 FORMAT('1'//20X,20A4)
           READ(1,130) A,B,UO,IX,N
       130 FORMAT(3F10.5,2I10)
           C=B
           WRITE(3,150) A,B,UO
       150 FORMAT(//20X,'THE THEORETICAL VELOCITY SOLUTION FOR A SPHEROID
          1'//2X,'A=',F10.5/2X,'B=',F10.5/1X,'UO=',F10.5//
          2T4,'I',T10,'X',T20,'Y',T30,'Z',T40,'U',T50,'V',T60,'W',T70,'CP'/)
           ASQ=A*A
           BSQ=B*B
           CSQ=C*C
           A4=ASQ*ASQ
           B4=BSQ*BSQ
           C4=CSQ*CSQ
           S=SQRT(1.-CSQ/ASQ)
           AO=2.*A*B*C/(ASQ-CSQ)**1.5*(.5*DLOG((1.DO+S)/(1.DO-S))-S)
           P=2.*UO/(2.-AO)
           Y=0.
           DXA=1./N
           XA=1.+DXA
           DO 25 I=1,N
           IF(IX)7,7,8
         7 XA=XA-DXA
           X=XA*A
           Z=ABS(SQRT(1.-XA*XA))
           GO TO 10
         8 READ(1,130)X,Y,Z
        10 XSQ=X*X
           YSQ=Y*Y
           ZSQ=Z*Z
           G=XSQ/A4+YSQ/B4+ZSQ/C4
           U=-P/G*(YSQ/B4+ZSQ/C4)
           V=P/G*(X*Y/ASQ/BSQ)
           W=P/G*(X*Z/ASQ/CSQ)
           VSQ=U*U+V*V+W*W
           CP=1.-VSQ/UO/UO
           WRITE(3,300)I,X,Y,Z,U,V,W,CP
       300 FORMAT(I5,12F10.5)
        25 CONTINUE
           GO TO 20
        30 RETURN
           END
```

Figure I-2. Sample data set for SPHEROID.

This program is written in Fortran IV and is designed to run on an IBM 370-165 computer. Execution requires 30,000 bytes of core storage and approximately 10 seconds to predict the velocity distribution and pressure coefficient for 1600 points. For a given run, the program requires the following input data:

(1) The 80 characters of the alphameric array TITLE which are used as a header for identifying output. It is used to terminate execution by setting the first four characters to "ENDﬗ".

(2) The x-axis (a), the y and z-axes (b), the free stream x-velocity, U_o , the mode indicator, IX, and the number of points for this run, N. The mode indicator, IX, may take on two values. If IX is zero a meridian solution of N points is generated by the program. By indicating IX as one, the programmer implies that a meridian solution is not desired and that N specifically selected points will follow.

(3) The points x, y, and z (if IX equals one) at which the solution is to be calculated.

Format specifications for these variables may be found in Figure I-1. A sample data set of a three-to-one prolate spheroid is given in Figure I-2 and the output for that data set is given in Figure I-3.

Figure I-1. Format specification for SPHEROID.

449

APPENDIX J - Supplementary Bibliography

Listed below is bibliographic information on some post-1970 documents in the NASA information collection dealing, at least in a general way, with the estimation of aerodynamic characteristics of subsonic aircraft. The methods described are, for the most part, intended for computer solution. A listing of the appropriate program is given in quite a number of these documents along with user instructions. The documents are available (except where noted) from the National Technical Information Service by requesting the first number (document accession number) shown in each citation.

The list was assembled from a computer search of the NASA information collection for documents indexed under both aerodynamic characteristics and computer methods or numerical methods. Results from a second search employing the terms Prediction Analysis Techniques, Flow characteristics, Aerodynamic Drag, Aerodynamic Coefficient, and Performance Prediction are also included. Titles and mini-abstracts were then examined to select those citations given below. The present authors have not seen the complete document in most cases. The list is intended to provide some contemporary references for the worker just entering the field as well as an indication of the direction in which contemporary research is moving for the benefit of the more casual reader. Because the task of assembling this work as a whole necessarily limited the period of preparing the literature review to late 1972 it was not possible to include the documents listed here in the review.

1. 72B10618   Langley-11047
    Vortex-Lattice Fortran Program for Estimating Subsonic Aerodynamic
    Characteristics of Complex Planforms
    Margason, R. J.; Lamar, J. E.

2. 71B10153   Lewis-11382
    Computer Program for Calculating Aerodynamic Forces on Blade Sections
    MC Nally, W. D.

3. 74A15445   9 pages
    Computerized Design of Transport Airplane
    Takao, K.
    Japan, Defense Academy, Memoirs, Vol. 13, Sept. 1973, pp. 327-335.

4. 73A22433   2 pages
    Calculation of Forces on Stores in the Vicinity of Aircraft
    Serbin, H.
    Journal of Aircraft, Vol. 10, Feb. 1973, pp. 123, 124.

5. 73A17213   4 pages
    Leading-Edge Force Features of the Aerodynamic Finite Element Method
    Lan, C.-T.; Roskam, J.
    Journal of Aircraft, Vol. 9, Dec. 1972, pp. 864-867.

6. 73A14377   37 pages
    Transonic Profile Theory - Critical Comparison of Various Procedures
    Eberle, A.; Sacher, P.
    Deutsche Gesellschaft Fuer Luft - und Raumfahrt, Symposium Ueber
    Tragfluegel-Aerodynamik Bei Schallnahen Stroemungen, Goettingen, West
    Germany, Oct. 26, 27, 1972, Paper in German.

7. 72A43455   SAWE Paper 908   15 pages
    An Aerodynamics Model Applicable to the Synthesis of Conventional
    Fixed-Wing Aircraft
    Peyton, R. S.
    Society of Aeronautical Weight Engineers, Annual Conference, 31st,
    Atlanta, Ga., May 22-25, 1972.

8. 72A16798   AIAA Paper 72-221
    A Simple Model for the Theoretical Study of Slat-Airfoil Combinations
    Liebeck, R. H.; Smyth, D. N.

9. 71A24253   SAE Paper 710389
    Low Speed Airfoil Analysis Using a Small Digital Computer
    Koepsel, R. E.; Miller, J. A.; Wentz, W. H., Jr.
    New York, Society of Automotive Engineers, Inc., Society of Automotive
    Engineers, National Business Aircraft Meeting, Wichita, Kan., Mar.
    24-26, 1971.

10. 73N70722   GDC-TN-70-AVLABS-09   54 pages
    User's Manual for Tilt Wing and Deflected Slipstream Aerodynamics Program
    Pederson, S. K.

11. 74N14741   NASA-TR-R-421
    A Study of the Nonlinear Aerodynamics of Bodies in Nonplanar Motion
    (Numerical Analysis of Aerodynamic Force and Moment Systems During
    Amplitude, Arbitrary Motions)
    Schiff, L. B.
    Ph.D. Thesis - Stanford University, Calif.

12. 74N14739   NASA-TM-X-62321   31 pages
    Plotting Program for Aerodynamic Lifting Surface Theory -- User Manual
    for Fortran Computer Program
    Medan, R. T.; Ray, K. S.

13. 74N11810   NASA-TM-X-62309
    Geometry Program for Aerodynamic Lifting Surface Theory
    Medan, R. T.

454

14. 73N31226
    Starting Vortex, Separation Bubbles and Stall -- A Numerical Study of
    Laminar Unsteady Flow Around an Airfoil
    Mehta, U. B.
    Ph.D. Thesis, Illinois Inst. of Tech., Chicago, Univ. Microfilms Order
    No. 73-12222.

15. 73N27890   NASA-CR-2217
    Analytical Method for Predicting the Pressure Distribution About a
    Nacelle at Transonic Speeds
    Keith, J. S.; Ferguson, D. R.; Merkle, C. L.; Heck, P. H.; Lahti, D. J.

16. 73N27889   NASA-TN-D-6933
    On the Numerical Simulation of Three-Dimensional Transonic Flow with
    Application to the C-141 Wing
    Lomax, H.; Bailey, F. R.; Ballhaus, W. F.

17. 73N27212   NAL-TR-309
    In Japanese; English Summary
    A Numerical Calculation of a Two-Dimensional Incompressible Potential
    Flow Around a Set of Airfoils Applying the Relaxation Method
    Nakamura, M.
    National Aerospace Lab., Tokyo, Japan.

18. 73N25010   ARC-R/M-3487
    The Calculation of the Spanwise Loading of Sweptback Wings with Flaps
    or All-Moving Tips at Subsonic Speeds
    Brebner, G. G.; Lemaire, D. A.

19. 73N24997
    Kelvin Impulse Theory Applied to Lift on Airfoils
    Delaney, J. A.
    Ph.D. Thesis, Cincinnati Univ., O.   Univ. Microfilms Order No. 72-31627.

20. 73N24325   SC-CR-72-3182
    Numerical Solution of the Three-Dimensional Boundary Layer on a Spinning
    Sharp Body at Angle of Attack
    Watkins, C. B., Jr.

21. 73N24319   NASA-TT-F-14918   24 pages
    The Kutta-Joukowsky Conditions in Three-Dimensional Flow
    Legendre, R.

22. 73N24302   90 pages
    The Three-Dimensional Structure of Transonic Flows Involving Lift
    Hafez, M. M.
    Ph.D. Thesis, Univ. of Southern Calif., Univ. Microfilms Order No.
    72-27661.

23. 73N24054   14 pages
Review of Two Methods of Optimizing Aircraft Design
Kirkpatrick, D. L. I.
In AGARD Aircraft Performance Prediction Methods and Optimization.

24. 73N22998   AFAPL-TR-72-100   124 pages
Lifting Surface Theory for Statically Operating Propellers
Murray, J. C.; Carta, F. C.

25. 73N21952   FTD-MT-24-1646-72   20 pages
Approximate Method of Calculating the Aerodynamic Load Distribution on a
Low-Flying Wing with a Fuselage
Tsvetkov, L. G.

26. 73N21913   NWL-TR-2796   122 pages
Body Alone Aerodynamics of Guided and Unguided Projectiles at Subsonic,
Transonic, and Supersonic Mach Numbers
Moore, F. G.

27. 73N21292   MDC-J5679-02   81 pages
Calculation of Potential Flow About Arbitrary Three-Dimensional Lifting
Bodies, User's Manual
Mack, D.

28. 73N19999   NASA-TN-D-7251   35 pages
Steady, Subsonic, Lifting Surface Theory for Wings with Swept, Partial
Span, Trailing Edge Control Surfaces
Medan, R. T.

29. 73N18054   FTD-HT-23-834-72   12 pages
Flow Around Wing Profile with the Presence of the Surface of a System
of Sources and Sinks
Baev, B. S.; Zhuravlev, V. N.

30. 73N17035   AFFDL-TR-72-26-VOL-3   207 pages
V/STOL Aircraft Aerodynamic Prediction Methods Investigation, Volume 3
Manual for Computer Programs
Wooler, P. T.; Kao, H. C.; Schwendemann, M. F.; Wasson, H. R.; Ziegler, H.

31. 73N17033   AFFDL-TR-72-26-VOL-1   238 pages
V/STOL Aircraft Aerodynamic Prediction Methods Investigation, Volume 1
Theoretical Development of Prediction Methods
Wooler, P. T.; Kao, H. C.; Schwendemann, M. F.; Wasson, H. R.; Ziegler, H.

32. 73N16985   120 pages
A Computer Program for the Prediction of Aerodynamic Characteristics of
Wing-Body-Tail Combinations at Subsonic and Supersonic Speeds, Part 2
Anders, S.; Bustavsson, L.
Aeronautical Research Inst. of Sweden, Stockholm.

456

33. 73N15035   32 pages
The Avsyn Air Vehicle Synthesis Program for Conceptual Design
Sanders, K. L.; Staley, P. A.

34. 73N15004   9 pages
Design of Airfoils with High Lift at Low and Medium Subsonic Mach numbers
Wortmann, F. X.
In AGARD Fluid Dyn. of Aircraft Stalling.

35. 73N14989   NASA-CR-2186   58 pages
Steady Inviscid Transonic Flows Over Planar Airfoils - A Search for a
Simplified Procedure
Magnus, R.; Yoshihara, H.

36. 73N13007   RIAS-TR-72-166   50 pages
On Lifting Wings with Parabolic Tips
Jordan, P. F.

37. 73N13006   AFOSR-72-1737TR   50 pages
Exact Solution for Lifting Surfaces
Jordan, P. F.

38. 73N12315   NPS-59NN72082A   48 pages
Flow Studies of Axisymmetric Bodies at Extreme Angles of Attack
Smith, L. H.; Nunn, R. H.

39. 73N12224   NLR-TR-70088-U   38 pages
Computer Application of Subsonic Lifting Surface Theory
Labrujere, T. E.; Wooters, J. G.
National Aerospace Lab., Amsterdam, Netherlands.

40. 73N11999   NASA-CR-2157   115 pages
Calculative Techniques for Transonic Flows About Certain Classes of
Wing-Body Combinations, Phase 2
Stahara, S. S.; Spreiter, J. R.

41. 73N10242   MDC-J5264-VOL-2   310 pages
Investigation of Aerodynamic Analysis Problems in Transonic Maneuvering,
Volume 2 Airfoil Analysis Computer Program
Gentry, A. E.

42. 73N10042   AFFDL-TR-71-155-PT-1   50 pages
Takeoff and Landing Analysis (Tola) Computer Program
Lynch, U. H. D.

43. 72N32302   NASA-TT-F-14547   20 pages
Subsonic and Supersonic Flow Around Nonaxisymmetric Fuselages
Rothman, H.

44. 72N31991   18 pages
Lift-Curve Slope and Aerodynamic Centre Position of Wings in Inviscid
Subsonic Flow
Engineering Sciences Data Unit, London, England, Avail on Subscription
from Engineering Sciences Data Unit, 251-259 Regent Street, London,
WIR 7AD.

45. 72N31909   SRL-TR-70-0009   24 pages
A Method for Aerodynamic and Propulsion Design Optimization of the Short
Range Dogfight Missile
Forbrich, C. A.; Gallington, R. W.; Hatlelid, J. E.; Hyde, J. P.;
Murrow, R. C.

46. 72N30264   NASA-TT-F-14538   6 pages
Numerical Study of the Influence of the Wing Tip Shape on the Vortex
Sheet Rolling Up
Rehback, C.

47. 72N29002   RIAS-TR-72-040   70 pages
Complete Solution for Lifting Wings with Parabolic Tips
Jordan, P. F.

48. 72N26023   NASA-CR-112065-3   114 pages
Theoretical Prediction of Interference Loading on Aircraft Stores, Part 3
Programmer's Manual
Danfernandes, F.

49. 72N26003   ONERA-TP-1088   16 pages
In French; English Summary
Aerofoil Stall Prediction in Incompressible Flow
Vincentdepaul, M.

50. 72N26001   26 pages
Calculation of Pressure Distributions for an Airfoil in Subcritical Flow
Including the Effect of the Boundary Layer
Anders, S.; Gustavsson, L.; Hillgren, R. S.; Toll, G. I.
Aeronautical Research Inst. of Sweden, Stockholm.

51. 72N24005   AFOSR-72-0034TR   19 pages
The Discrete Vortex Approximation of a Finite Vortex Sheet
Moore, D. W.

52. 72N24003   MDC-J5108   AFOSR-72-0370   92 pages
A General Class of Airfoils Conformally Mapped from a Circle
James, R. M.

53. 72N22002   ARC-R/M-3630
The Linearized Subsonic Flow Over the Centre-Section of a Lifting Swept
Wing
Rossiter, P. J.

458

54. 72N15010   48 pages
In German; English Summary
Downwash Investigations on a Missile Tails
Gregoriou, G.
Messerschmitt-Boelkow-Blohm G. M. B. H., Ottobrunn, West Germany.

55. 72N11289   NASA-TN-D-6530   16 pages
Contribution to Methods for Calculating the Flow About Thin Lifting
Wings at Transonic Speeds. Analytic Expressions for the Far Field
Klunker, E. B.

56. 71N30488   NASA-TT-F-702
The Calculation of the Pressure Distribution on a Cascade of Thick
Airfoils by Means of Fredholm Integral Equations of the Second Kind
Martensen, E.
(Translation of Ref. 24).

57. 71N19385   9 pages
A Method for Predicting Interference Forces and Moments on Aircraft
Stores at Subsonic Speeds
Fernandes, F. D.
In AGARD Aerodyn. Interference, Jan. 1971.

58. 74N18680   NASA-CR-2334   187 pages
Correlation of Full-Scale Drag Predictions with Flight Measurements on
the C-141A Aircraft: Phase 2 Wind Tunnel Tests, Analysis, and Prediction
Techniques. Volume 2 Wind Tunnel Test and Basic Data, Final Report
MacWilkinson, D. G.; Blackerby, W. T.; Paterson, J. H.

59. 74N18679   NASA-CR-2333   166 pages
Correlation of Full-Scale Drag Predictions with Flight Measurements on
the C-141A Aircraft Phase 2 Wind Tunnel Test, Analysis, and Prediction
Techniques, Volume 1 Drag Predictions, Wind Tunnel Data Analysis and
Correlation, Final Report
MacWilkinson, D. G.; Blackerby, W. T.; Paterson, J. H.

60. 74N14716   50 pages
Aircraft Drag Prediciton for Project Appraisal and Performance Estimation
Butler, S. F. J.
In AGARD Aerodyn. Drag.

61. 74N14712   11 pages
Aerodynamic Drag and Lift of General Body Shapes at Subsonic, Transonic,
and Supersonic Mach Numbers
Moore, F. G.
In AGARD Aerodynamic Drag.

459

62. 74N14711   38 pages
A Survey of Drag Prediction Techniques Applicable to Subsonic and
Transonic Aircraft Desgin
Paterson, J. H.; MacWilkinson, D. G.; Blackerby, W. T.
In AGARD Aerodyn. Drag.

63. 73N15010   16 pages
The Effect of Leading Edge Geometry on High Speed Stalling
Moss, G. F.; Haines, A. B.; Jordon, R.
In AGARD Fluid Dyn. of Aircraft Stalling.

64. 72N22995   NASA-TM-X-67791   94 pages
Nonplanar Method for Predicting Incompressible Aerodynamic Coefficients
of Rectangular Wings with Circular-Arc Camber
Lamar, J. E.
Ph.D. Thesis, Virginia Polytechnic Institute, Avail. NTIS HC $6.75.

65. 72N11869   NASA-TM-X-67413   9 pages
A Comparison of Some Aerodynamic Drag Factors as Determined in Full-
Scale Flight with Wind-Tunnel and Theoretical Results
Saltzman, E. J.; Bellman, D. R.
In AGARD Facilities and Tech. for Aerodyn. Testing at Transonic Speeds
and High Reynolds Number.

66. 74N1423   9 pages
New Investigations for Reducing the Base Drag of Wings with a Blunt
Trailing Edge -- Effects of Splitter Plates and Splitter Wedges on
Aerodynamic Drag Coefficients
Tanner, M.
In AGARD Aerodyn. Drag.

67. 74N14719   20 pages
Drag of Supercritical Airfoils in Transonic Flow -- Comparison with
Conventional Airfoil Drag Coefficients
Kacprzynski, J. J.
In AGARD Aerodyn. Drag.

68. 74N14709   AGARD-CP-124   469 pages
Partly in English and Partly in French
Aerodynamic Drag
Advisory Group for Aerospace Research and Development, Paris, France,
Avail. NTIS HC $25.50, Proceedings of Specialist Meeting, Izmir, Turkey,
10-13 April 1973.

69. 73N31228   ESOL-71020   ESDU-67010   23 pages
Aerofoils having a Specified Form of Upper Surface Pressure Distribution
Details and Comments on Design
Engineering Sciences Data Unit, London, England, Avail Issuing Activity,
Sponsored by Min. of Defence and Roy. Aeron. Scc.

460

70. 73N23362   ESDU-BODIES-02.04.02   3 pages
Drag of Streamline Solids of Revolution (Transition at 0.11 Behind Nose
(Numerical Analysis of Drag of Streamlined Bodies of Revolution for
Various Length to Diameter Ratios)
Engineering Science Data Unit, London, England, Avail. Issuing Activity.

71. 74N14729   11 pages
The Drag of Externally Carried Stores Its Prediction and Alleviation --
Drag Reduction by Redesign or Development of New Aircraft Installations
Pugh, P. G.; Hutton, P. G.
In AGARD Aerodyn. Drag.

72. 74N14728   22 pages
The Drag Resulting from Three-Dimensional Separations Caused by Boundary-
Layer Diverters and Nacelles in Subsonic and Supersonic Flow
Peake, D. J.; Rainbird, W. J.
In AGARD Aerodyn. Drag.

73. 74N14724   15 pages
A Study of Flow Separation in the Base Region and Its Effects During
Powered Flight -- Interaction Between Propulsive Jet and Free Stream
Flow
Addy, A. L.; Korst, H. H.; White, R. A.; Walker, B. J.
In AGARD Aerodyn. Drag.

74. 74N14718   12 pages
Remarks on Methods for Predicting Viscous Drag -- Aerodynamic Drag
Prediction for High Angles of Attack and Multielement Airfoils
Smith, A. M. O.; Cebeci, T.
In AGARD Aerodynamic Drag.

75. 74N14717   9 pages
Appendix A Data Item Service for Aircraft Drag Estimation -- Collection,
Dissemination, and Development of Aerodynamic Drag Prediction Data
Engineering Sciences Data Unit, London, England, Avail. NTIS, In AGARD
Aerodyn. Drag.

76. 74N12704   ESDU-73028   23 pages
Drag of Two-Dimensional Steps and Ridges Immersed in a Turbulent
Boundary Layer at Subsonic and Supersonic Speeds
Engineering Sciences Data Unit, London, England, Avail Issuing Activity,
Sponsored by Roy. Aeron. Soc.

77. 74N10311   ESDU-BODIES-02.04.01-AMEND-A   3 pages
Drag of Streamline Solids of Revolution (Transition at Nose)
Engineering Sciences Data Unit, London, England, Avail. Issuing Activity.

78. Not in File   23 pages
    Measurements in the Thick Axisymmetric Turbulent Boundary Layer Near
    the Tail of a Body of Revolution
    Patel, V. C.; Nakayama, A.; Damian, R.
    Journal of Fluid Mechanics, Vol. 63, pp. 345-368, April 1974.

79. 74A19581   13 pages
    Numerical Solution of the Three-Dimensional Boundary Layer on a Spinning
    Sharp Body at Angle of Attack
    Watkins, C. B. Jr.
    Computers and Fluids, Vol. 1, Dec. 1973, pp. 317-329.

80. 74A11957   21 pages
    The Numerical Solution of the Navier-Stokes Equations for Laminar
    Incompressible Flow Past a Paraboloid of Revolution
    Veldman, A. E. P.
    Computers and Fluids, Vol. 1, Sept. 1973, pp. 251-271.

81. 72A41264   2 pages
    Lift on Airfoils with Separated Boundary Layers
    Ness, N.
    Journal of Aircraft, Vol. 9, Aug. 1972, pp. 607, 608.

82. 74N13984   AFOSR-73-1265TR-PT-7   45 pages
    Three-Dimensional Laminar Boundary Layer Over Body of Revolution at
    Incidence, Part 7   The Extremely High Incidence Case
    Wang, K. C.

83. 73N25006   ARC-R/M-3221   20 pages
    Numerical Methods for Calculating the Zero-Lift Wave Drag and the Lift-
    Dependent Wave Drag of Slender Wings
    (Evaluation of Double Integral Equation for Calculation of Wave Drag
    Due to Volume and Aerodynamic Lift of Slender Wings)
    Weber, J.
    In Arc Aerodyn. Res., Including Heating, Airfoils, and Boundary Layer
    Studies, Vol. 1, pp. 155-173.

84. 72N32330   200 pages
    The Optimum Shaping of Axisymmetric Bodies for Minimum Drag in
    Incompressible Flow
    (Optimum Hydrodynamic Configurations for Submerged Minimum Drag
    Axisymmetric Vehicles in Incompressible Fluids)
    Parsons, J. S.; Goodson, R. E.

85. 72M50104   1 page   FORTRAN V Program
TRW Vortex-Lattice (N-Surface) Subsonic Aerodynamics
(Aerodynamic Calculations for Single or Multiple Lifting Surface
Configurations by Improved Vortex-Lattice Method)
Romere
National Aeronautics and Space Administration, Lyndon B. Johnson
Space Center, Houston, Texas.

86. 71M51203   1 page   FORTRAN IV Program
Subsonic Unsteady Aerodynamic
(Steady and Unsteady Aerodynamic Coefficients·for Subsonic Lifting
Surfaces)
Harrison
National Aeronautics and Space Administration, Marshall Space Flight
Center, Huntsville, Ala., Jan. 1972.

87. 74M10002   1 page   CDC FORTRAN Program 1,293 cards
Modified Multhopp Mean Camber Program -- Mean Camber Surface Required to
Support Set of Loadings on Composite Wing in Subsonic Compressible Flow
National Aeronautics and Space Administration, Langley Research Center,
Langley Station, Va., Price: Program $275.00/ Documentation $15.50.

88. 73M10132   2 pages   FORTRAN IV Program 6,594 cards
An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail
Configurations in Subsonic and Supersonic Flow
(Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and
Supersonic Flow)
Aerophysics Research Corp., Bellevue, Wash., Price: Program $600.00/
Documentation $27.50.

89. 73M10131   1 page   FORTRAN IV Program 960 cards
General Lifting-Line Jet Flap FORTRAN Program for Estimating Subsonic
Aerodynamic Characteristics
(Estimation of Subsonic Aerodynamic Characteristics of Jet-Flapped Wings)
Northrop Corporate Labs., Hawthorne, Calif., Price: Program $250.00/
Documentation $4.00.

90. 74A25062   ONERA, TP No. 1247   14 pages
In French
Aerodynamic Problems of the Short Takeoff and Landing Aircraft
Ceresuela, R.
(L'Aeronautique et L'Astronautique, No. 41, 1973, pp. 43-56).

91. 74A21893   4 pages   In Russian
An Optimization Method for a Generalized Class of Functionals and Its
Application to the Problem of Determining the Shape of a Body Exhibiting
Maximum Aerodynamic Efficiency
Bunimovich, A. I.; Dubinskii, A. V.
Moskovskii Universitet, Vestnik, Seriia I - Matematika, Mekhanika, Vol.
28, Nov.-Dec. 1973, pp. 87-90.

92. 74A21104   11 pages
The High Subsonic Flow Around a Two-Dimensional Aerofoil with a Trailing
Edge Control Surface
Nixon, D.
Aeronautical Quarterly, Vol. 24, Nov. 1973, pp. 273-283.

93. 74A20765   AIAA Paper 74-106   28 pages
Aeroelastic Loads Predictions Using Finite Element Aerodynamics
Rowan, J. C.; Burns, T. A.
American Institute of Aeronautics and Astronautics, Aerospace Sciences
Meeting, 12th, Washington, D. C., Jan. 30-Feb. 1, 1974.

94. 74A20280   7 pages
Subsonic Potential Aerodynamics for Complex Configurations - A General
Theory
Morino, L.; Kuo, C.-C.
AIAA Journal, Vol. 12, Feb. 1974, pp. 919-197.

95. 74A19684   36 pages
In French
Flexible Lifting Surfaces -- In Steady Inviscid Compressible Flow
Vaussy, P.
(Association Technique Maritime et Aeronautique, Session, 73rd, Paris,
France, May 14-18, 1973), Association Technique Maritime et
Aeronautique, Bulletin, No. 73, 1973, pp. 361-394, Discussion, p. 395.

96. 74A18878   AIAA Paper 74-107   14 pages
A Finite Element Method for Potential Aerodynamics Around Complex
Configurations
Chen, L.-T.; Suciu, E. O.; Morino, L.
American Institute of Aeronautics and Astronautics, Aerospace Sciences
Meeting, 12th, Washington, D. C., Jan. 30-Feb. 1, 1974.

97. 74A18820   AIAA Paper 74-72   8 pages
Odin - Optimal Design Integration System for Synthesis of Aerospace
Vehicles
Rau, T. R.; Decker, J. P.
American Institute of Aeronautics and Astronautics, Aerospace Sciences
Meeting, 12th, Washington, D. C., Jan. 30-Feb. 1, 1974.

98. 74A18681   13 pages
In Russian
Calculation of the Aerodynamic Characteristics of a Wing System Moving
at Subsonic Speed Near Land or Smooth Water Surface
Ermolenko, S. D.; Khrapovitskii, V. G.
Samoletostroenie I Tekhnika Vozdushnogo Flota, No. 32, 1973, pp. 3-15.

464

99. 74A15965   9 pages
Exact Method of Designing Airfoils with Given Velocity Distribution in
Incompressible Flow
Strand, T.
Journal of Aircraft, Vol. 10, Nov. 1973, pp. 651-659.

100. 74A15747   22 pages
Note on the Aerodynamic Theory of Oscillating T-Tails. I - Theory of
Wings Oscillating in Yaw and Sideslip
Ichikawa, T.; Isogai, K.
Japan Society for Aeronautical and Space Sciences, Transactions, Vol.
16, No. 33, 1973, pp. 173-194.

101. 74A15709   7 pages
In Russian
A Problem of Designing the Optimal External Contours of an Aircraft
Osipov, V. A.; Tereshchenko, A. M.
Avaiatsionnaia Teknika, Vol. 16, No. 3, 1973, pp. 11-17.

102. 73A40427   5 pages
Simplified Aerodynamic Theory of Oscillating Thin Surfaces in Subsonic
Flow
Jones, W. P.; Moore, J. A.
AIAA Journal, Vol. 11, Sept. 1973, pp. 1305-1309.

103. 73A37846   7 pages
In Russian
Integral Equation in the Theory of Lifting Surfaces
Poliakhov, N. N.
Leningradskii Universitet, Vestnik, Matematika, Mekhanika, Astronomiia,
Apr. 1973, pp. 115-121.

104. 73A37545   15 pages
In Romanian
New Contributions to the Iterative Method for Aerodynamic Calculations
of Wings in Subsonic Flows
Patraulea, N. N.
Studii Si Cercetari De Mecanica Aplicata, Vol. 31, No. 1, 1973, pp. 15-29
15-29.

105. 73A36394   5 pages
A Finite-Element Method for Calculating Aerodynamic Coefficients of a
Subsonic Airplane
Hua, H. M.
Journal of Aircraft, Vol. 10, July 1973, pp. 422-426.

106. 73A31746   4 pages
     Remarks on Vortex-Lattice Methods
     (Optimal Grid Arrangement in Vortex Lattice Method of Lifting Surface
     Aerodynamic Analysis, Comparing Numerical with Kernel Function Results
     for Simple Wing Planforms)
     Hough, G. R.
     Journal of Aircraft, Vol. 10, May 1973, pp. 314-317.

107. 73A27732   5 pages
     In German
     FS-28 - A Contribution to a Possible Development Trend in Light-
     Aircraft Design
     (Light Motorized Glider-Type Aircraft Design, Development and Flight
     Testing, Discussing Aerodynamic Configuration, Structural Design and
     Performance Characteristics)
     Deutscher Aerokurier, Vol. 17, Mar. 1973, pp. 152-156.

108. 73A26256   440 pages
     In Russian
     Design and Stability of Airplanes and Helicopters
     (Russian Book on Airplane and Helicopter Design and Stability Covering
     Selection of Wing/Rotor/Configuration and Power Plant, Subsystem Design,
     Strength, Reliability, Lifetime, Etc.)
     Voskoboinik, M. S.; Lagosiuk, G. S.; Milen'Kii, Iu. D.; Mirtov, K. C.;
     Osokin, D. P.; Skripka, M. L.; Ushakov, V. S., Chernenko, Zh. S.
     Moscow, Izdatel'Stvo Transport, 1972.

109. 73A24915   4 pages
     Symmetrical Airfoils Optimized for Small Flap Deflection
     Wortmann, F. X.
     Aero-Revue, Mar. 1973, pp. 147-150.

110. 73A23468   4 pages
     Estimation of Aerodynamics for Slender Bodies Alone and with Lifting
     Surfaces at Alpha's from 0 Deg to 90 Deg
     (Aerodynamic Forces and Moments Estimation for Slender Bodies of
     Circular and Noncircular Cross Section without and with Lifting
     Surfaces at 0-90 Degree Angles of Attack)
     Jorgensen, L. H.
     AIAA Journal, Vol. 11, Mar. 1973, pp. 409-412.

111. 73A21611   10 pages
     In Russian
     Discrete Vortex Scheme of a Wing of Finite Span
     Vorob'ev, N. F.
     Akademiia Nauk, SSSR, Sibirskoe Ctdelente, Izvestiia, Seriia
     Tekhnicheskikh Nauk, Oct. 1972, pp. 59-68.

466

112. 73A18510   10 pages
     Pressure Airships - A Review
     (Airships Design, Constructional and Operational Characteristics,
     Discussing Aerodynamics, Flight Control, Performance and Trim)
     Hecks, K.
     Aeronautical Journal, Vol. 76, Nov. 1972, pp. 647-656.

113. 73A11657   39 pages
     In German
     Further Development and Employment of the Subsonic Panel Method
     (Three-Dimensional Potential Flow Past Arbitrarily Shaped Aerodynamic
     Configurations, Using Hess-Smith Numerical Method)
     Kraus, W.
     Deutsche Gesellschaft Fuer Luft-Und Raumfahrt, Jahrestagung, 5th,
     Berlin, West Germany, Oct. 4-6, 1972.

114. 73A10048   1 page
     Simplification of the Wing-Body Interference Problem
     Graham, R. E.; McDowell, J. L.
     Journal of Aircraft, Vol. 9, Oct. 1972, p. 752.

115. 72A44298   18 pages
     In German
     Computation of the Potential-Theoretical Flow Around Wing-Fuselage
     Combinations and a Comparison with Measurements
     Koerner, H.
     Zeitschrift Fuer Flugwissenschaften, Vol. 20, Sept. 1972, pp. 351-368.

116. 72A41150   9 pages
     Experimental Investigations of Separated Flows on Wing-Body Combinations
     with Very Slender Wings at Free-Stream Mach Numbers from 0.5 to 2.2
     Stahl, W.; Hartmann, K.; Schneider, W.
     International Council of the Aeronautical Sciences, Congress, 8th,
     Amsterdam, Netherlands, Aug. 28-Sept. 2, 1972.

117. 72A41138   9 pages
     Potential Flow Calculations to Support Two-Dimensional Wind Tunnel Tests
     on High-Lift Devices
     Labrujere, Th. E.; Schipholt, G. J.; De Vries, O.
     International Council of the Aeronautical Sciences, Congress, 8th,
     Amsterdam, Netherlands, Aug. 28-Sept. 2, 1972.

118. 72A34060   AIAA Paper 72-682
     Analytic Prediction of Dynamic Stall Characteristics
     (Aerodynamic Stall Characteristic Prediction from Static Experimental
     Data for Airfoils, Noting Boundary Layer Effects)
     Ericsson, L. E.; Reding, J. P.
     American Institute of Aeronautics and Astronautics, Fluid and Plasma
     Dynamics Conference, 5th, Boston, Mass., June 26-28, 1972.

119. 72A32250   138 pages
Handbook of Airfoil Sections for Light Aircraft
(Book on Airfoil Section Designs for Light Aircraft Covering Wind
Tunnel Studies of Lift Drag Ratio as Function of Angle of Attacks
Rice, M. S.
Milwaukee, Vis., Aviation Publications, $3.95, 1971.

120. 72A31401   22 pages
Vortex-Lattice Method for Calculating Aerodynamic Coefficients of a
Subsonic Airplane
(Vortex-Lattice Method for Subsonic Aircraft Aerodynamic Coefficients
Calculation, Verifying Results with Airbus Lifting Surface Wing Tunnel
Test Data)
Hua, H. M.
Astronautical Society of the Republic of China, Transactions, Nov. 1,
1971, pp. 1-22.

121. 72A29132   7 pages
In Russian
Invariant Methods of Determining the Lift Coefficient of Various
Aerodynamic Profiles from the Flow Spectrum
(Aerodynamic Profiles Lift Coefficient Determination by Empirical
Formula Based on Potential Flow Lines Obtained by Conformal Mapping)
Suprun, V. M.
Samoletostroenie I Tekhnika Vozdushnogo Flota, No. 25, 1971, pp. 8-14.

122. 72A25595   SAE Paper 720337   12 pages
Consideration of Application of Currently Available Transport-Category
Aerodynamic Technology in the Optimization of General Aviation
Propeller-Driven Twin Design.
(Transport Aircraft Aerodynamic Design Technology Application to General
Aviation Propeller Driven Twin Engine Aircraft, Discussing Wing Loading
and Aspect Ratio Optimization)
Raisbeck, J. D.
Society of Automotive Engineers, National Business Aircraft Meeting,
Wichita, Kan., Mar. 15-17, 1972.

123. 72A18958   AIAA Paper 72-188   16 pages
Review and Evaluation of a Three-Dimensional Lifting Potential Flow
Computational Method for Arbitrary Configurations
(Subsonic Three Dimensional Potential Flow Computational Method Lifting
Aerodynamic Configurations Analysis and Design)
Rubbert, P. E.; Saaris, G. R.
American Institute of Aeronautics and Astronautics, Aerospace Sciences
Meeting, 10th, San Diego, Calif., Jan. 17-19, 1972.

124. 72A17194   11 pages
In French
Research and Tests on Laminar Airfoils
(Laminar Flow Airfoils for Gliders, Optimizing Profiles for Favorable
Velocity and Pressure Distribution)
De Lagarde, B.; De Loof, J. P.
L'Aeronautique et L'Astronautique, No. 32, 1971, pp. 29-39.

125. 72A16109   5 pages
Refinement of the Nonplanar Aspects of the Subsonic Doublet-Lattice
Lifting Surface Method
(Subsonic Doublet-Lattice Lifting Surface Method. Nonplanar Aspects
Refinement, Using Wing-Tail Configurations)
Rodden, W. P.; Giesing, J. P.; Kalman, T. P.
Journal of Aircraft, Vol. 9, Jan. 1972, pp. 69-73.

126. 72A12723   66 pages
Experimental Studies of Aerodynamic Coefficients of a Wing-Fuselage
Combination and Comparison with the Results of Linear and Nonlinear
Theories at Subsonic Speeds
(Wing-Fuselage Combination Aerodynamic Coefficients, Comparing Experi-
mental Data with Subsonic Linear and Nonlinear Theoretical Results)
Herpfer, E.; Heynatz, J. T.
Deutsche Gesellschaft Fuer Luft-Und Raumfahrt, Jahrestagung, 4th,
Baden-Baden, West Germany, Oct. 11-13, 1971.

127. 71A43312   14 pages
Lifting Line Theory of a Wing in Uniform Shear Flow
(Minimum Drag and Lifting Line Characteristics of Large Aspect Ratio
Wing in Univorm Shear Flow with Velocity Variations Along Span)
Morita, K.
JSME, Bulletin, Vol. 14, pp. 550-563.

128. 71A39543   3 pages
Equations of an Aircraft's Form
(Computer Aided Aircraft Design, Analysis and Production, Discussing
Numerical Master Geometry Program Developed by British Aircraft
Corporation)
New Scientist and Science Journal, Vol. 51, pp. 410-412.

129. 71A24012   332 pages
In Russian
Preliminary Design of an Aircraft
(Soviet Book on Aircraft Preliminary Design Specifications as Function
of Performance, Aerodynamic and Structural Parameters. Discussing
Tradeoffs in Operational Requirements for Specific Configurations)
Diachanko, A. A.; Fadeev, N. N.; Goroshchenko, B. T.
Moscow, Izdatel'stvo Mashincstroenie.

130. 71A18511   AIAA Paper 71-50   19 pages
Aerodynamics of Finned Missiles at High Angle of Attack
(Finned Missiles Aerodynamics at High Angle of Attack, Examining
Body Vortex Wake Region Interaction with Fins)
Nicolaides, J. D.; Oberkampf, W. L.
American Inst. of Aeronautics and Astronautics, Aerospace Sciences
Meeting, 9th, New York, N. Y., Jan. 25-27, 1971.

131. 71A13737   4 pages
The Lift of a Slender Combination of a Fuselage of Rectangular Cross-
Section with a High Wing
(Lift of Slender Aircraft with Rectangular Cross Section Fuselage and
High Wing)
Andrews, R. D.
AERONAUTICAL JOURNAL, Vol. 74, pp. 903-906.

132. 74N16700   NASA-TM-X-62325   91 pages
Equation Solving Program for Aerodynamic Lifting Surface Theory
Medan, R. T.; Lemmer, O. J.

133. 74N16699   NASA-TM-X-62323   67 pages
Boundary Condition Program for Aerodynamic Lifting Surface Theory --
Using FORTRAN IV
Medan, R. T.; Ray, K. S.

134. Not in File   13 pages
Sting Free Drag Measurements on Ellipsoidal Cylinders at Transition
Reynolds Numbers
Judd, M.; Vlajinac, M.; Covert, E. E.
Journal of Fluid Mechanics, Vol. 48, pp. 353-365, July 1970.

135. 74N14710   11 pages
Technical Evaluation Report -- Application of Aerodynamic Drag Research
to Design of Aircraft
Butler, S. F. J.
In AGARD Aerodyn. Drag.

136. 72A12723   66 pages
In German
Experimental Studies of Aerodynamic Coefficients of a Wing-Fuselage
Combination and Comparison with the Results of Linear and Nonlinear
Theories at Subsonic Speeds
(Wing-Fuselage Combination Aerodynamic Coefficients, Comparing Experi-
mental Data with Subsonic Linear and Nonlinear Theoretical Results)
Herpfer, E.; Heynatz, J. T.
Deutsche Gesellschaft Fuer Luft-Und Raumfahrt, Jahrestagung, 4th,
Baden-Baden, West Germany, Oct. 11-13, 1971.

470

137. 73N27209   NASA-TT-F-14962   34 pages
Calculation of Potential Flow Around Profiles with Suction and Blowing
(Integral Equations for Calculating Incompressible Potential Flows Aroun
Around Profiles with Suction and Blowing)
Jacob, K.
Washington NASA Transl. into English from Ing.-Arch., Berlin, Vol. 32,
No. 1, 1963, pp. 51-65.

138. 73N27045   AFFDL-TR-72-132   592 pages
Optimal Design Integrations of Military Flight Vehicles (ODIN/MFV)
(Development of Digital Computing System for Synthesis and Optimization
of Military Flight Vehicle Preliminary Designs) Final Report, May 1971 -
Sept. 1972
Hague, D. S.; Glatt, C. R.
Aerophysics Research Corp., Bellevue, Wash.

139. 73N25043   ARC-R/M-3279 ARC-22503   46 pages
On the Design of Wing-Body Combinations of Low Zero-Lift Drag Rise at
Transonic Speeds
(Optimum Design of Wing-Body Combinations for Zero-Lift Drag Rise at
Transonic Speeds)
Lord, W. T.
Ministry of Aviation, London, England
In ARC Aerodyn. Res. Progr., Including Turbine, Nozzle, Flutter, and
Instrumentation Studies, Vol. 2, pp. 1381-1426.

140. 73N24049   57 pages
Parametric and Optimization Techniques for Airplane Design Synthesis
(Parametric and Optimization Techniques for Aircraft Design Synthesis
to Show Principal Lines of Data Flow for Component Development)
Wallace, R. E.
In AGARD Aircraft Performance Prediction Methods and Optimization.

141. 73N24042   AGARD-LS-56   345 pages
In English and Partly in French
Aircraft Performance Prediction Methods and Optimization
(Development) and Application of Aircraft Performance Prediction Methods
for Subsonic and Supersonic Transport and Fighter Aircraft)
Williams, J. A/ED.
Advisory Group for Aerospace Research and Development, Paris, France,
Avail. NTIS HC $19.25.

142. 73N21916   157 pages
The Design of a Vertical Takeoff and Landing Aircraft for the General
Aviation Market
(Design and Development of Vertical Takeoff Aircraft Configuration for
Use with Air Transportation Services Between Major Population Centers)
Harding, J. C.
Ph.D. Thesis, Dartmouth Coll., Hanover, N. H., Avail Univ. Microfilms
Order No. 72-23515.

143. 73N21899   NASA-CR-112231   32 pages
A Parametric Study of Planform and Aeroelastic Effects on Method for
Computing the Aerodynamic Influence Coefficient Matrix of Nonplanar
Wing-Body-Tail Configurations
(Aerodynamic Influence Coefficient Matrix for Nonplanar Wing-Body-Tail
Configurations)
Roskam, J.

144. 73N15997   222 pages
An Optimal Configuration Design of Lifting Surface Type Structures Under
Dynamic Constraints
(Optimized Design of Supersonic Aircraft Wing Based on Linear
Combination of Weight of Wing and Aerodynamic Drag Minimization)
Miura, H.
Ph.D. Thesis, Case Western Reserve Univ., Cleveland, Ohio, Avail. Univ.
Microfilms Order No. 72-18717.

145. 73N14004   41 pages
In Italian
Theory of Subsonic Lifting Surface (Fixed Mode), Some Considerations
and Propositions for Improving the Method of Numerical Solution
(Improving Method of Numerical Calculation of Aerodynamic Coefficients
for Subsonic Lifting Surface)
Polito, L.
Pisa Univ., Italy, Faculty of Engineering.

146. 72N32007   12 pages
Calculation of the Aerodynamic Characteristics of Lifting Systems
Composed of Rectangular Wings
(Calculating Aerodynamic Characteristics of Lifting Systems Composed of
Rectangular Wings Arranged One Behind Another)
Joint Publications Research Service, Arlington, Va.
In Its Rept. from the Higher Educational Inst., Aviation Tech., pp. 15-
26.

147. 72N29016   404 pages
Preliminary Design of an Aircraft
(Development of Handbook of Basic Principles of Aircraft Design Based
on Technical Specifications and Calculation of Aerodynamic
Characteristics)
Goroschchenko, B. T.; Dyachenko, A. A.; Fadeev, N. N.
Transl. into English from "Eskiznoe Proektircvanie Samoleta", 1970,
pp. 1-332, Translation of No. 129.

148. 72N29000   326 pages   Unclassified Document
Aerodynamics
(Application of Aerodynamic Data to Design of Passenger Aircraft with
Emphasis on Laws of Gas Motion Flow and Boundary Layer Theory)
Mkhitaryan, A. M.
Transl. into English from the Publ., "Aerodinamika", 1970, pp. 175-428.

472

149. 72N23042   AFOSR-72-0369TR   67 pages
Theoretical Studies on the Aerodynamics of Slat Airfoil Combinations
(Aerodynamic Characteristics of Leading Edge Slats Plus Main Airfoil
Combinations), Final Report
Liebeck, R. H.

150. 72N18037   AEDC-TR-71-186   68 pages
Calculation of Forces on Aircraft Stores Located in Distrubed Flow
Fields for Application in Store Separation Prediciton
(Aerodynamic Characteristics of Bomb in Steady, Incompressible,
Potential Flow Based on Model), Final Report, 1 Apr. 1970, - 30 June 1971
MacDermott, W. N.; Johnson, P. W.

151. 72N11017   AFOSR-71-1079TR   53 pages
On the Aerodynamic Forces of Oscillating Two-Dimensional Lifting Surfaces
(Aerodynamic Lift Characteristics of Oscillating Two-Dimensional Airfoil
Subjected to Sinusoidal Gust), Final Report
Yates, J. E.; Houbolt, J. C.

152. 71N37597   46 pages
Calculation of Potential Flow About Arbitrary Three-Dimensional Lifting
Bodies
(Computer Program Development for Potential Flow Calculation About
Lifting Bodies), Final Report, Dec. 1969 - Oct. 1970
Hess, J. L.

153. 71N33016   AFFDL-TR-71-26-VOL-1   208 pages
Stol High Lift Design Study, Volume 1 - State of the Art Review of
(Test Data Reduction and Prediciton Techniques for High Lift Aerodynamic
and Propulsion System Configurations for Short Takeoff Aircraft Design -
Bibliographies)
May, F.; Widdison, C. A.

154. 71N29335   22 pages
Calculation Methods for Unsteady Airforces of Tandem Surfaces and
T-Tails in Subsonic Flow
(Numerical Analysis of Aerodynamic Loads and Coefficients for Tandem
and T-Tail Surfaces Harmonically Oscillating in Subsonic Flow)
Davis, D. E.
In AGARD Symp. on Unsteady Aerodynamics for Aeroelastic Analyses of
Interfering Surfaces, Part 1, 1971.

155. 71N21973   NASA-TN-D-6243   11 pages
Charts for Predicting the Subsonic Vortex-Lift Characteristics of
Arrow, Delta, and Diamond Wings
(Predicting Aerodynamic Characteristics of Arrow, Delta, and Diamond
Wing Platforms Using Prandtl-Glauert Transformation)
Polhamus, E. C.

156. 71N20115   80 pages
Calculation of the Three-Dimensional Potential Flow Around Lifting
Non-Planar Wings and a Wing-Bodies Using a Surface Distribution of
Quadrilateral Vortex-Rings
(Numerical Calculation of Steady Three Dimensional Potential Flow
Around Lifting Nonplanar Aerodynamic Configurations Based on Surface
Distribution of Quadrilateral Vortex-Rings)
Maskew, B.

157. 71N13402   ONERA-NT-163   34 pages
In French
Precise Calculation of Unsteady Aerodynamic Pressures in Subsonic Flow
(Transient Pressures and Aerodynamic Coefficients of Rectangular Wings
in Subsonic Flow Using Linear Equations)
Salaun, P.
Office National D-etudes et de Recherches Aerospatiales, Paris, France.

158. 74A18897   3 pages
In German
Investigations Concerning Wing-Fuselage Interference in the Case of
Subsonic Velocity
Koerner, H.; Ahmed, S.R.; Mueller, R.
Dfvlr-Nachrichten, Dec. 1973.

159. 74A17270   18 pages
In German
Nonlinear Airfoil Theory with Allowance for Ground Effects --- for
Aerodynamic Interference Problems Solution
Hummel, D.
Zeitschrift Fuer Flugwissenschaften, Vol. 21, Dec. 1973, p. 425-442.

160. 74A17180   DGLR Paper   33 pages
In German
The Effect of Wing Planform Modifications on the Aerodynamic Performances
of Fighter Aircraft
Staudacher, W.
Oesterreichische Gesellschaft Fuer Weltraumforschung Und Flugkoerpertechnik
and Deutsche Gesellschaft Fuer Luft-Und Raumfahrt, Gemeinsame Jahrestagung,
6th, Innsbruck, Austria, Sept. 24-28, 1973.

161. 74A11606   SAE Paper 730876   9 pages
New Airfoil Sections for General Aviation Aircraft --- Cruising and Flap
Development Tests
Wentz, W. H., Jr.
Society of Automotive Engineers, National Aerospace Engineering and
Manufacturing Meeting, Los Angeles, Calif., Oct. 16-18, 1973.

162. 73A43028   11 pages
In German
The Panel Method for the Calculation of the Pressure Distribution on
Missiles in the Subsonic Range
Kraus, W.; Sacher, P.
Zeitschrift Fuer Flugwissenschaften, Vol. 21, Sept. 1973, p.301-311.

163. 73A41192   13 pages
The Aerodynamic Development of the Wing of the A 300B
Mcrae, D.M.
Aeronautical Journal, Vol. 77, July 1973, p. 367-379.

164. 73A38007   4 pages
Prediction of the Lift and Moment on a Slender Cylinder-Segment Wing-Body
Combination
Crowell, K.R.; Crowe, C.T.
Aeronautical Journal, Vol. 77, June 1973, p. 295-298.

165. 73A34676   SAE Paper 730318   25 pages
Applications of Advanced Aerodynamic Technology to Light Aircraft
Crane, H.L.; McGhee, R.J.; Kohlman, D.L.
Society of Automotive Engineers, Business Aircraft Meeting Wichita,
Kan., Apr. 3-6, 1973.

166. 73A32819   49 pages
In French
Calculation of the Characteristics of Tail Fins in the Vortical Field of
a Wing
Yermia, M.
Association Aeronautique et Astronautique de France, Colloque D'Aero-
dynamique Appliquee, 9th, Saint-Cyr-L'Ecole, Yvelines and Paris, France,
Nov. 8-10, 1972.

167. 73A25490   AIAA Paper 73-353   10 pages
Application of Computer-Aided Aircraft Design in a Multidisciplinary
Environment
Fulton, R.E.; Sobieszczanski, J.; Storaasli, O.; Landrum, E.J.;
Loendorf, D.
AIAA, ASME, and SAE, Structures, Structural Dynamics, and Materials
Conference, 14th, Williamsburg, Va., Mar. 20-22, 1973.

168. 73A23856   32 pages
Transonic Airfoils - Recent Developments in Theory, Experiment, and
Design
Nieuwland, G.Y.; Spee, B.M.
In Annual Review of Fluid Mechanics. Volume 5. (A73-23851 10-12)
Palo Alto, Calif., Annual Reviews, Inc., 1973, p. 119-150.

169. 73A1419   2 pages
Lift of Wing-Body Combination
Yang, H. T.
AIAA Journal, Vol. 10, Nov. 1972, p. 1535, 1536.

170. 74N21645   AD-775538 MDC-J5831   277 pages
Analytical Studies of Two-Element Airfoil Systems
James, R.M.
Interim Report, Feb. 1971 - Dec. 1973.

171. 74N21635   NASA-TN-D-7579   15 pages
On the Use of Thick-Airfoil Theory to Design Airfoil Families In Which
Thickness and Lift are Varied Independently
Barger, R.L.

172. 74N20694   AD-774430   91 pages
Addition of an Arbitrary Body Analysis Capability to the Boeing TEA
236 Finite Element Computer Program
Westphal, J. L.
M.S. Thesis
Air Force Inst. of Tech., Wright-Patterson AFB, Ohio. (School of
Engineering)

173. 74N18654   AGARD-R-614   20 pages
Interfering Lifting Surfaces In Unsteady Subsonic Flow Comparison
Between Theory and Experiment
Becker, J.
Presented at 37th AGARD Structures and Mater. Panel Meeting, The Hague,
7-12 Apr. 1973.

174. 74N17707   41 pages
In German; English Summary
Reciprocal Influence of a Body of Finite Length and a Wing at Mid-Wing
Position at Subsonic Speed
Gregoriou, G.
Avail. Ntis HC $5.25; Bundeswehramt, Bonn 30 DM

175. 74N13674   24 pages
Reynolds Number Effects at Low Speeds on the Maximum Lift of
Two-Dimensional Aerofoil Sections Equipped with Mechanical High Lift
Devices
Thain, J.A.
In Natl. Res. Council of Can. Quart. Bull. of the Div. of Mech. Eng.
and the Natl. Aeron. Estab. p. 1-24 (see N74-13673 04-34)

476

176. 74N11821   NASA-TN-D-7428   71 pages
Low Speed Aerodynamic Characteristics of a 17 Percent Thick Airfoil
Section Designed for General Aviation Applications
McGhee, R.J.; Beasley, W.D.

177. 74N11815   NASA-TT-F-15183   25 pages
Calculation of Flows Around Zero Thickness Wings with Evolutive Vortex
Sheets
Rehbach, C.
Transl. into English from Rech. Aerosp. (France), No. 2,
Mar. - Apr. 1972, p. 53-61.

178. 74N10019   NASA-CR-2344   97 pages
The Effects of Leading-Edge Serrations on Reducing Flow Unsteadiness
About Airfoils, an Experimental and Analytical Investigation Final
Report
Schwind, R.G.; Allen, H.J.

179. 73N26000   NASA-TT-F-14959   42 pages
Airfoil Profiles in a Critical Reynolds Number Region
(Force Measurements and Pressure Distributions on Three Gottinger
Airfoil Profiles Druing Transition From Laminar To Turbulent Boundary
Layer Flow)
Kraemer, K.
Transl. into English from Soderdruck Aus der Z. Forsch. Auf Dem Gebiete
des Ingenieurwesens' ' (West Germany), V. 27, No. 2, 1961, p. 33-46.

180. 73N25002   ARC-R/M-3180   19 pages
Observations of the Flow Over a Two Dimensional 4 Percent Thick Aerofoil
At Transonic Speeds
(Wind Tunnel Tests to Determine Pressure Distributions for Four Percent
Thick, Circular ARC, Biconvex Airfoil at Transonic Speeds)
Henshall, R.D.; Cash, R. F.
In ARC Aerodyn. Res., Including Heating, Airfoils, and Boundary Layer
Studies, Vol. I, p. 63-81 (see N73-24999 16-01).

181. 73N24040   AD-757813   81 pages
An Exact Method of Designing Airfoils with Given Velocity Distribution in
Incompressible Flow An Extension of the Lighthill and Arlinger Methods
(Application of Conformal Mapping Procedures for Designing Airfoil
Shapes with High Design Lift Coefficients)   Final Report
Strand, T.
15 Jun. - 15 Dec. 1972.

182. 73N24000   ARC-R/M-3238   40 pages
      The Pressure Distribution on Two-Dimensional Wings Near the Ground
      (Numerical Analysis of Pressure Distribution in Incompressible Flow
      on Two-Dimensional Airfoils Near Ground)
      Bagley, J.A.
      In ARC Res. Progr. on Aerodyn. Heating, Airfoils, Wings, and Aircraft
      During 1960, Vol. 1, p. 79-118 (see N73-23995 15-01).

183. 73N22977   NASA-CR-112297   233 pages
      An Analytical Study for the Design of Advanced Rotor Airfoils
      (Design and Evaluation of Two Airfoils for Helicopter Rotors for
      Reduction of Rotor Power Requirements)
      Kemp, L.D.

184. 73N21914   AD-755480 MDC-J5679-01   166 pages
      Calculation of Potential Flow About Arbitrary Three-Dimensional
      Lifting Bodies
      (Development of Method for Calculating Potential Flow about Arbitrary
      Lifting Three-Dimensional Bodies with Emphasis on Bound Vorticity and
      Application of Kutta Condition)   Final Technical Report
      Hess, J.L.

185. 73N21907   NASA-TN-D-7183   41 pages
      Low-Speed Wind Tunnel Investigation of A Semispan Stol Jet Transport
      Wing Body with an Upper Surface Blown Jet Flap
      (Wind Tunnel Tests to Determine Static Longitudinal Aerodynamic
      Characteristics of Jet Transport Wing-Body with Upper Surface Blown Jet
      Flap for Lift Augmentation)
      Phelps, A.E., III; Letko, W.; Henderson, R.L.

186. 73N21054   7 pages
      Wake Characteristics of a Two-Dimensional Symmetric Aerofoil
      (Generation of Aerodynamic Noise by Turbulent Wake Behind Rotary
      Wing Airfoil and Relationship to Drag and Lift Coefficients)
      Kavrak, I.
      In AGARD Aerodyn. Rotary Wings (see N73-21031 12-02)

187. 73N20995   198 pages
      An Analysis of the Design of Airfoil Sections for Low Reynolds Numbers
      (Design of Airfoil Sections for Low Reynolds Numbers Based on Requirement
      to Achieve Transition Upstream of Major Adverse Pressure Gradient)
      Miley, S.J.
      Ph.D. Thesis
      Mississippi State Univ., State College. Avail Univ. Microfilms
      Order No. 72-20272.

188. 73N16283   AD-751075 MDC-J5713   63 pages
A New Family of Airfoils Based on the Jet-Flap Principle
(Air Foils Based On Utilization of Jet-Flap Principle)
Bauer, A.B.
Technical Report, Apr. 1971-Apr. 1972.

189. 73N15992   AD-751045   116 pages
Circulation Control By Steady and Pulsed Blowing for a Cambered
Elliptical Airfoil
(Short Takeoff Aircraft Lift Augmentation and Prevention of Airflow
Separation on Cambered Ellipitical Airfoil Section Using Circulation
Control)
Walters, R.E.; Myer, D.P.; Holt, D.J.

190. 73N15050   DLR-FB-72-63   42 pages
In German; English Summary
Theoretical Parameter Studies of Wing-Fuselage Combinations
(Prediction Analysis Method to Determine Influence of Geometry
Parameters on Aerodynamic Characteristics of Body-Wing Configuration)
Koerner, H.
Deutsche Forschungs- und Versuchsanstalt Fuer Luft- und Raumfahrt,
Brunswick (West Germany). (Abteilung Fuer Theoretische Aerodynamik.)
Avail. Ntis HC $4.25; Dfvlr. Porz, West Ger. 11 DM.

191. 73N15010   16 pages
The Effect of Leading Edge Geometry on High Speed Stalling
(Aerodynamic Configurations of Swept Wings to Improve Lift Performance
at Stall in Higher Range of Subsonic Speeds)
Moss, G.F.; Haines, A.B.; Jordon, R.
In AGARD Fluid Dyn. of Aircraft Stalling (see N73-14998 06-02).

192. 73N15009   12 pages
A Simplified Mathematical Model for the Analysis of Multielement Airfoils
Near Stall
(Development of Procedure for Determining Characteristics of High Lift
Systems Where Viscous Effects Dominate)
Bhateley, I.C.; Bradley, R.G.
In AGARD Fluid Dyn. of Aircraft Stalling (see N73-14998 06-02).

193. 73N15008   12 pages
The Low Speed Stalling of Wings With High Lift Devices
(Analysis of Aerodynamic Stall Characteristics of Wing Sections With
High Lift Devices in Two-Dimensional Flow)
Foster, D.N.
In AGARD Fluid Dyn. of Aircraft Stalling (see N73-14998 06-02).

194. 73N15007   27 pages
Aerodynamics in High Lift Airfoil Systems
(Analysis of Aerodynamic Processes Occurring in Flow Past Unpowered
Multi-Element Airfoils in High Lift Attitude)
Smith, A.M.O.
In AGARD Fluid Dyn. of Aircraft Stalling (see N73-14998 06-02).

195. 73N14998   AGARD-CP-102   342 pages
Partly in English and Partly in French
Fluid Dynamics of Aircraft Stalling
(Proceedings of Conference on Fluid Dynamics of Aircraft Stalling to
Include Stall and Post-Stall Aerodynamic Characteristics of Various
Military Aircraft)
Advisory Group for Aerospace Research and Development, Paris (France)
Avail. Ntis HC $19.25
Presented at Fluid Dyn. Panel Specialists Meeting, Lisbon, 25-28 Apr. 1972.

196. 73N14051   AD-749726   FTD-HT-23-181-72   36 pages
Application of the Wing Impulse Theory to the Determination of Propeller
Slip Stream Influence on Wing Aerodynamic Characteristics
(Wing Impulse Theory Applied to Determination of Propeller Slipstream
Influence on Wing Aerodynamic Characteristics, Using Airfoil of Finite
Span)
Kopylov, G.N.
Transl. into English from Tr. Vysshee Aviationnoe Uchilishche
Grazhdanskii Aviatsii (USSR), No. 24, 1965, p. 24-43.

197. 73N14043   AD-749485   AFFDL-TR-72-96-PT-2   86 pages
Development of Theoretical Method for Two-Dimensional Multi-Element
Airfoil Analysis and Design. Part 2 Leading-Edge Slat Design Method
(Computer Program for Designing Leading Edge Slats for Producing
Specified Pressure Distribution on Main Airfoil)
McGregor, O.W.; McWhirter, J.W.
Final Report, 24 May 1971 - 12 Jun. 1972.

198. 73N10242   AD-740124 MCD-J5264-VOL-2   310 pages
Investigation of Aerodynamic Analysis Problems in Transonic Maneuvering.
Volume 2 Airfoil Analysis Computer Program
(Development of Computer Program for Analyzing Mono-Element and Multi-
Element Airfoils at Subsonic Speed with Attached Air Flow - Vol. 2)
Gentry, A.E.
Final Report, Jun. 1970 - Aug. 1971.

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."
—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546