Numerical Optimization Techniques for Bound Circulation Distribution for Minimum Induced Drag of Nonplanar Wings: Computer Program Documentation

John M. Kuhlman and Tzuchun Jeffrey Ku

GRANT NSG-1357
AUGUST 1981
Numerical Optimization Techniques for Bound Circulation Distribution for Minimum Induced Drag of Nonplanar Wings: Computer Program Documentation

John M. Kuhlman and Tzuchun Jeffrey Ku
*Old Dominion University Research Foundation*
*Norfolk, Virginia*

Prepared for
Langley Research Center
under Grant NSG-1357

National Aeronautics and Space Administration
Scientific and Technical Information Branch
1981
A two-dimensional advanced panel far-field potential flow model of the undistorted, interacting wakes of multiple lifting surfaces has been developed which allows the determination of the spanwise bound circulation distribution required for minimum induced drag. This model has been implemented in a FORTRAN computer program, the use of which is documented in this report.

The nonplanar wakes are broken up into variable sized, flat panels, as chosen by the user. The wake vortex sheet strength is assumed to vary linearly over each of these panels, resulting in a quadratic variation of bound circulation. Panels are infinite in the streamwise direction. The theory is briefly summarized herein; sample results are given for multiple, nonplanar, lifting surfaces, and the use of the computer program is detailed in the appendixes.

INTRODUCTION

With the current resurgence of interest in utilization of unconventional aircraft concepts for future transport aircraft to provide reductions in drag, increases in fuel efficiency, and lower operating costs, there is a need for accurate estimations of the induced drag for nonplanar configurations. Examples of these novel configurations include wings fitted with end plates or winglets (ref. 1), the tandem wing (ref. 2), and the joined wing (ref. 3). The current far-field theoretical model allows very accurate calculation of induced drag for multiple nonplanar aerodynamic surfaces, allowing investigation of the
drag-reduction potential of such nonplanar aircraft concepts. Further, the bound circulation and wake vortex strength distributions necessary to achieve this minimum induced drag are computed. The bound circulation output may then be used to determine the aerodynamic surface camber shapes required to achieve this minimum drag.

The theoretical wake model has been described in detail in reference 4. It assumes the vortex sheet strength to vary piecewise linearly on a number of flat wake panels. Wake rollup is neglected. Analytical expressions are developed for the normal velocities induced by each wake panel at any point on the wake using the Biot-Savart Law (ref. 5). The wake vortex strength is integrated spanwise to compute bound circulation, and the product of local bound circulation with the total induced normal velocity is analytically integrated to obtain the induced drag. To determine the wake vortex distribution required for minimum drag, two theoretical methods are used: Munk's criterion (ref. 6), and a direct optimization technique. The Munk criterion technique is computationally more efficient, since only the induced velocity expressions are utilized. This technique is similar to the theory developed in reference 7. The direct optimization technique is necessary for determination of induced drag for relative optimum configurations which might have additional constraints on bending moment or pitching moment. Analytical expressions are developed for the derivatives of $C_D$, $C_L$, $C_m$, and $C_R$ with respect to the unknown values of the wake vortex sheet strengths at the corners of each vortex sheet panel, as described in reference 8. This wake model using the direct optimization technique has been implemented in a vortex lattice wing design computer code (ref. 9), as described in reference 8. Comparisons between results of the original design code (ref. 9) and modified code with the current wake model (ref. 8) are given in references 8 and 10.

This report details the computer program which was written to implement the theoretical wake model of reference 4. The theory is briefly summarized. Use of the program and sample input and output data are given in the appendixes: the code is briefly described (Appendix A), input data preparation is explained (Appendix B), output data is described (Appendix C), sample input and output data are given (Appendix D), and a listing of the computer program is given (Appendix E).
SYMBOLS

\( A, A_{ij} \)
- matrix of influence coefficients in induced drag
  [eq. (11)]

\( A_{ij}, A^{2}_{ij}, A^{3}_{ij}, A^{4}_{ij} \)
- integrals appearing in normal wash expression
  [eq. (3)]

\( B \)
- constant appearing in integrals in Appendix A

\( b \)
- wing span

\( c \)
- constant appearing in integrals in Appendix A

\( C_{p} \)
- pressure coefficient

\( \Delta C_{p} \)
- difference in pressure coefficient

\( C_{B} \)
- wing root bending moment coefficient

\( C_{D} \)
- induced drag coefficient

\( C_{D,ij} \)
- induced drag coefficient on wake panel \( i \) due to induced
  velocity of panel \( j \) and its image [eq. (6)]

\( C_{l} \)
- lift coefficient

\( C_{m} \)
- pitching moment coefficient

\( d \)
- constant appearing in integrals in Appendix A

\( G_{i}, \bar{G}_{i}, \ddot{G}_{i} \)
- variables containing unknown wake vortex sheet strengths,
  appearing in drag coefficient equation [eq. (6)]

\( h \)
- vertical separation of diamond wing roots (figs. 6, 7)

\( h_{ij} \)
- distance between influenced point on panel \( i \) and influencing
  point on panel \( j \) (fig. 1)

\( h'_{ij} \)
- distance between influenced point on panel \( i \) and influencing
  point on image of panel \( j \) (fig. 1)

\( I_{1,ij}, I_{2,ij}, I_{3,ij}, I_{4,ij}, I_{5,ij}, I_{6,ij} \)
- influence coefficient integrals appearing in drag coefficient
  equation [eq. (6)]
k  induced drag efficiency factor, defined as ratio of planar wing induced drag to that of nonplanar configuration

\&  vertical fence height (fig. 2)

n  integer appearing in integrals in Appendix A

N  number of wake panels on one-half of total configuration

R  constant appearing in integrals in Appendix A

\( R_{ij} \)  projection of distance \( h_{ij} \) onto the plane of influenced panel \( i \) (fig. 1)

\( R'_{ij} \)  projection of distance \( h'_{ij} \) onto the plane of influenced panel \( i \) (fig. 1)

\( S_{\text{ref}} \)  reference wing area

\( s \)  local wake panel coordinate

\( s \)  wake panel semiwidth

T  constant appearing in integrals in Appendix A

U  free-stream velocity

\( w_n \)  normal wash velocity

\( w_{n,j} \)  normal wash velocity induced at point \( s = s_i \) on panel \( i \) due to panel \( j \) and its image [eq. (2)]

\( w_0 \)  constant appearing in Munk's criterion normal wash velocity expressions [eqs. (7) and (8)]

X  streamwise coordinate

Y  spanwise coordinate

Z  vertical coordinate, positive down

\( \gamma \)  wake trailing vortex sheet strength

\( \Gamma \)  bound circulation

\( \Gamma \)  average bound circulation

\( \Gamma_0 \)  bound circulation at outboard edge of wake panel [eq. (4)]

\( \eta \)  nondimensional spanwise coordinate

\( \lambda \)  Lagrange multiplier in equations (10) and (12)

\( \sigma_{ij} \)  angle between y-axis and orientation of \( h_{ij} \) (fig. 1)

\( \phi \)  dihedral angle
THEORETICAL DEVELOPMENT

The wakes are broken into N finite, flat panels. The panels are numbered sequentially from tip to root of each planform. On each panel, the wake vortex sheet strength is assumed to vary linearly, as

$$\gamma_j = \frac{\gamma_{j+1} + \gamma_j}{2} + \frac{s_j}{s_{j+1}} \cdot \frac{\gamma_{j+1} - \gamma_j}{2}$$

(1)

The $\gamma_j$ value equals the vortex sheet strength at the junction between segments $j$ and $j-1$ (see the inset on fig. 1). Figure 1 also shows the wake geometry notation used in the current theory for a wing-winglet configuration.

The induced velocity normal to point $\delta_i$ on wake segment $i$, due to wake segment $j$ and its image, may be written, using the Biot-Savart Law, as

$$w_{n,j} = \frac{\gamma_{j+1} + \gamma_j}{2} \frac{1}{2\pi} \left\{ \int_{-s_j}^{s_j} \left( \frac{R_{ij}}{h_{ij}} - \frac{R_{ij}'}{h_{ij}'} \right) d\delta_j \right\}$$

$$+ \frac{\gamma_{j+1} - \gamma_j}{2} \frac{1}{2\pi} \left\{ \int_{-s_j}^{s_j} \left( \frac{\delta_j R_{ij}}{h_{ij}^2} - \frac{\delta_j R_{ij}'}{h_{ij}^2} \right) d\delta_j \right\}$$

(2)

The $R_{ij}$ and $h_{ij}$ are distances as shown in figure 1. This expression is evaluated (ref. 4) as
\[ w_{n,j} = \frac{\gamma_{j+1} + \gamma_j}{2} (A_{1ij} + A_{2ij}) + \frac{\gamma_{j+1} - \gamma_j}{2} (A_{3ij} + A_{4ij}) \]

(3)

where the \( A_{ij} \), etc., are determined by the wake geometry. Next, the \( \gamma \) distribution is integrated spanwise beginning at the tip of the current planform to obtain the bound circulation as

\[ \Gamma(\Delta_i) = \Gamma_o(-s_i) + \frac{s_i}{4} \left( \gamma_{i+1} + \gamma_i \right) + \left( \frac{\gamma_{i+1} + \gamma_i}{2} \right) \Delta_i \]

(4)

where \( \Gamma_o(-s_i) \) is the value of \( \Gamma \) at \( \Delta_i = -s_i \), which is a known linear function of the \( \gamma \) values.

The drag induced on segment \( i \) by segment \( j \) and its image is, in coefficient form,

\[ C_{D,ij} = \frac{1}{s_{\text{ref}}} \int_{-s_i}^{+s_i} \left( \frac{\gamma(\Delta_i)}{U} \right) \left( \frac{w_{n,i}(\Delta_i)}{U} \right) d\Delta_i \]

(5)

This equation has been evaluated analytically as described in reference 4 using the MACSYMA symbolic manipulation language (ref. 11). The result is an expression which is quadratic in the unknown \( \gamma_j \) values:

\[ C_{D,ij} = \frac{1}{s_{\text{ref}}} \left( C_{ij} I_{1,i,j} + C_{ij} I_{2,i,j} + C_{ij} I_{3,i,j} + C_{ij} I_{4,i,j} + C_{ij} I_{5,i,j} + C_{ij} I_{6,i,j} \right) \]

(6)

where the \( C_{ij} \), \( \bar{C}_{ij} \), \( \hat{C}_{ij} \), etc., are linear functions of the \( \gamma_j \)s (ref. 4). The \( I_{1,i,j} \) through \( I_{6,i,j} \) are again determined by wake geometry, being integrals of combinations of the \( A_{1ij} \) through \( A_{4ij} \) times \( \Delta_i^n \), for \( n = 0, 1, 2 \), as given in reference 4.
The Munk optimization procedure uses (ref. 6)

\[ \frac{w_n}{\cos \phi} = w_o = \text{constant} \quad (7) \]

which for the assumed wake model is written as

\[
\cos \phi_j = \frac{1}{w_o} \sum_{j=1}^{N} \left\{ \left( \frac{\gamma_j + 1}{2} \right) \left[ A_{1ij} + A_{2ij} \right] + \left( \frac{\gamma_j + 1}{2} \right) \left[ A_{3ij} + A_{4ij} \right] \right\} \quad (8)
\]

This yields \( N \) equations for \( N + 1 \) unknowns, the \( N \) \( \gamma_j \) values and \( w_o \).

The system is completed by specifying a \( C_L \) value, since (ref. 4)

\[
C_L = \frac{8}{S_{\text{ref}}} \left\{ \frac{1}{3} \sum_{j=1}^{N} \left( \cos \phi_j s_j \frac{\gamma_j + 1}{w_o} \left( \frac{2\gamma_j}{w_o} \right) \right) + \sum_{j=1}^{N} \left( \cos \phi_j s_j \frac{r_o (s_j)}{w_o} \right) \right\} \frac{w_o}{U} \quad (9)
\]

The direct optimization procedure extremizes (minimizes) the function

\[
C_D + \lambda \left( \sum_{j=1}^{N} C_{L,j} \cdot \frac{\gamma_j}{U} - C_L \right) \quad (10)
\]

where \( \lambda \) is a Lagrange multiplier and \( C_{L,j} \) is an analytical expression for the derivative of equation (9) above with respect to \((\gamma_j/U)\). Similarly, expressions for derivatives of \( C_D \) with respect to \((\gamma_j/U)\) have been developed, as reported in reference 4. The current method, which yields identical results, is to write the induced drag in matrix form as

\[
C_D = \left( \begin{array}{c} \gamma \end{array} \right)^T A \left( \begin{array}{c} \gamma \end{array} \right) \quad (11)
\]

The optimal \( \gamma_j/U \) values are then determined by \( N+1 \) equations given by
The $A$ matrix, as given in reference 8, is in terms of the $I_{1, i, j}$ through $I_{6, i, j}$ from equation (6). More details of the theory may be found in references 4 and 8.

SAMPLE RESULTS

Convergence studies for the present method have been given in reference 4. In general, the advanced panel method has been shown to be on the order of four times more accurate than a discrete vortex wake model having the same number of singularity unknowns, both for isolated planar wings and some limited, isolated nonplanar examples from references 12 and 13. In reference 8, results are given for multiple planform configurations from references 14 and 15. In these previous studies, the current wake model yielded induced drag values within 1 percent of the exact results for from 25 to 50 unknown wake strength values. The two optimization methods yield essentially identical results, except for $\gamma$ values near a wing tip. Cosine wake spacing greatly improves accuracy for a fixed number of unknowns. The reader is referred to references 4 and 8 for details of these studies.

In this section some additional solutions will be presented to illustrate the utility of the present theory. Figure 2 displays results for the present theory compared with that of references 14 and 15 for the planform sketched in the figure. The inboard 50 percent of the wing is flat, with a constant 30-degree dihedral outboard of the flat portion. In addition, a vertical fence of variable height $\ell$ is positioned at the dihedral break span location. This is termed configuration 5 in reference 14. Shown in figure 2 are values of the
induced drag efficiency parameter, $k$, defined as the ratio of the induced drag for a planar wing of equal span divided by the $C_D$ for the nonplanar configuration. The two theories agree favorably. For the present method, the fence has been oriented at $\phi = 89.7$ degrees to avoid numerical difficulties, as mentioned in reference 4. Figure 3 presents induced drag results for vee wings, compared with an exact solution from reference 16. In figure 4, similar results are presented for a diamond wake shape, again compared with exact results from reference 16. This last wake shape is of interest for the joined wing concept of reference 3. The present theory essentially duplicates these exact results. In figure 5 the bound circulation distributions from the present theory are shown for vee and diamond wings having $\phi = 30^\circ$. For minimum drag, both wings of the diamond shaped wake carry the same lift and have the same $\Gamma$ distributions. The vee wing has relatively a smaller fraction of its lift developed inboard than does the diamond wing having the same dihedral.

The results described above illustrate the capabilities of the computer program to accurately duplicate known exact solutions. As examples of more complicated wake geometries, figures 6 and 7 display the computed induced drag efficiency factors for a series of diamond wings (ref. 16), with the addition of end plates and winglets, respectively. These configurations can improve the induced drag efficiency factors for the concept of reference 3.

CONCLUSIONS

An advanced panel Trefftz plane wake model has been developed which allows accurate computation of the induced drag, bound circulation distribution, and wake vortex strength for nonplanar multiple planform configurations. The computer program which has been written to implement this theoretical method has been documented herein in the appendixes, including a listing of the code and user input instructions. A brief outline of the theory and some sample results have been given. These results reproduce accepted exact solutions for vee and diamond wings.
APPENDIX A

DESCRIPTION OF COMPUTER PROGRAM AND LIMITATIONS

This Appendix briefly describes the organization of the computer program written to implement the theory outlined earlier in this report, which has been described in some detail in references 4 and 8. Some limitations of this computer program are also discussed.

This program has been written in FORTRAN IV and is currently operational on a Cyber 173 computer at NASA/LaRC. This computer uses approximately 15 decimal digits in all computations. Some modifications to the code will be necessary if it is to be used on a computer system which uses a significantly different number of decimal digits. For example, the tolerances in subroutines SNTAN and LOGS may have to be varied. Further, double precision arithmetic will be required for all calculations for machines using eight significant figures. This would entail an IMPLICIT DOUBLE PRECISION (A-H, O-Z) statement in the main program and all subroutines, as well as use of double precision on all special functions: DCOS, DSTN, DLOG, DATAN, DATAN2, DSQRT, DABS, DMIN1, and DMAX1. Further, some of the variable names may need to be changed to be consistent with the implicit double precision statements.

The computer program consists of a main program, DNWASH, which reads the input data, performs the initial wake geometry computations, sets up the optimal induced drag matrix, and computes the final induced drag and normal wash and bound circulation distributions. This program calls nine subroutines: CCAL and CONCAL, which compute wake geometry constants; SNTAN and LOGS, which compute integrals appearing in the expression for \( C_D \), as detailed in the appendix to reference 4; WCAL, which computes an element of the optimization matrix for the Munk's criterion procedure; DRACAL and OPTCAL, which compute elements of the direct optimization matrix; GAMCAL, which computes the bound circulation terms; and SIMEQ, a linear equation solver. A listing of the complete computer program is given in Appendix E of this report, and an example input and output are given in Appendix D for one of the configurations discussed earlier in this report.
The known limitations of the computer program are now briefly described. First, the user-specified local dihedral angles may not anywhere equal 90 degrees. As discussed in reference 4, this value of $\phi$ may be approached ($\phi \approx 89.5^\circ$) to approximate the wake geometry for a vertical end plate or pylon. (Further examples of wake geometries with nearly vertical surfaces have been given in figures 2 and 6.) Second, the total number of wake panels for all planforms is currently limited to a maximum of 50. Based on results from reference 4 for isolated planar and nonplanar planforms, this should provide an induced drag solution accuracy comparable to that obtainable from 200 to 250 discrete vortex unknowns; that is, better than one percent accuracy. Third, the maximum number of individual planforms possible is currently 10, while the maximum number for which runs have been attempted to date is only 3. Fourth, the code currently does both a Munk criterion optimization and a direct optimization only for a single nonplanar or planar planform. Solutions for multiple interacting surfaces are computed using only the Munk criterion solution. (It is to be noted that the design code described in reference 8 does have the multiple planform capability using the direct optimization technique.) Fifth, based upon previous experience (ref. 4), it is recommended that a cosine spacing of the wake panels on all planforms be used.

Next, for configurations with multiple planforms, either the wakes must not cross one another, or any such wake crossings must occur at the edges of wake panels. This can be accomplished by specifying the wake crossing point as a common wake breakpoint on both planforms. (See Appendix B for a description of preparation of an input deck and definitions of the input data.) If wake crossings occur in the midrange of any wake panel, a message "80 ENTERED" is printed on the output file. For such cases, a midrange singularity occurs in the inverse tangent integrals evaluated in SNTAN. The code attempts to fit a pair of quadratics, one on either side of the singularity, to the inverse tangent portion of these integrands. The accuracy of this procedure is unknown, and any results so obtained are likewise of unknown accuracy. Further, for wake shapes comprised of continuously varying curved surfaces, it is possible for this problem of a midrange singularity to occur for multiple planforms even when the wakes do not themselves cross. Instead, all that is required to cause the apparent
singularity is for the projection of the plane containing one wake panel to intersect another wake panel away from that panel's edges. Again, this problem can be avoided by defining such points to be wake breakpoints on the second planform. It is believed that this apparent singularity is only due to the way in which the computer program is structured, where for example the above-mentioned inverse tangent integrals always occur in pairs, but each integral is evaluated individually. This has been alluded to in reference 4 (p. 16), where it is remarked that integrals of the form

\[ \int_{-s}^{s} \frac{\delta^n}{|R+T\delta|} \tan^{-1} \left( \frac{c+2B\delta}{|R+T\delta|} \right) \, d\delta \]

for $0 \leq n \leq 4$, become infinite for $R = T = 0$. However, since what actually must be evaluated is an integral of the form

\[ \int_{-s}^{s} \frac{\delta^n}{|R+T\delta|} \left\{ \tan^{-1} \left( \frac{c+2B\delta}{|R+T\delta|} \right) - \tan^{-1} \left( \frac{d+2B\delta}{|R+T\delta|} \right) \right\} \, d\delta \]

these two integrals are equal to the sum of the finite parts of the individual integrals, which have the form

\[ -\int_{-s}^{s} \frac{\delta^n}{c+2B\delta} \, d\delta \]

However, there is currently no logic in the code to automatically replace the original integrand with the simpler, finite part in the vicinity of a wake crossing point.

Finally, the optimum wake vortex sheet strengths and bound circulations for a single planform for the Munk criterion solution differ slightly from those computed by the direct optimization solution technique. Usually these differences are confined to the tip region of a planform. This is believed to be due to the inaccuracy of the piecewise linearly varying functional form of the wake vortex sheet strength in the vicinity of the tip, where the actual wake sheet strength should be infinite. Comparisons between the two solution techniques for a planar isolated wing have been given in reference 4. These slight differences in the $\gamma$ and $\Gamma$ distributions for the two
solution techniques do not appreciably affect the computed induced drag efficiency factors, but do lead to inaccuracies in the normal wash computations, especially for a nearly vertical surface, near a wing tip. Use of cosine wake panel spacing, as recommended above, will minimize this problem.
APPENDIX B

INPUT DATA PREPARATION

In this appendix the information necessary to prepare an input deck to use the computer program listed in Appendix E is given. A sample input deck, as well as the resultant output, are given in Appendix D.

The first four input cards specify control integers and integers which define the number of lifting surfaces and distribution of wake vortex panels. These cards are all in a 5I5 format. The specific information needed on each card is as follows:

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Variable</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NLLINE</td>
<td>(1-5)</td>
<td>Total number of lifting surfaces for current configuration; NLLINE ≤ 10.</td>
</tr>
</tbody>
</table>

For each of the NLLINE surfaces specified above, cards 2 through 5 must be specified, as follows:

2         ICNTRL  (1-5)  A control integer which determines the wake panel spacing on the current lifting surface, as follows:

A.  ICNTRL = 3; general input. User must specify NTOT (card 2, below), followed by values of (YHH(I), ZHH(I), PPP(I), I = 1, NTOT), the wake panel corner points and dihedral angles for the current surface. The YHH, ZHH, PPP values are specified in a 6F10.0 format. Cards 3 to 5 described below are not needed when ICNTRL = 3.

B.  ICNTRL = 6; circular arc wake. User must input NTOT (card 2, below), followed by one card giving the values of BET, THET, and BOT in 6F10.0 format. BET equals the ratio of the maximum vertical extent of the surface to the semispan. A value of BET = 0. corresponds to a flat surface, while BET = 1.0 corresponds to a semicircular arc. (See reference 4 for results for this type of surface.) THET is equal to the value, in degrees, for the angle subtended by one-half of the circular arc wake, while BOT equals the desired
<table>
<thead>
<tr>
<th>Card Number</th>
<th>Variable</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>wake semispan. Cards 3 to 5 described below are not needed when ICNTRL = 6.</td>
</tr>
<tr>
<td>C.</td>
<td>ICNTRL = 7; equal wake panel spacing. The computer program will automatically panel each flat portion of the wake between adjacent wake breakpoints (YY, Z) (described below; card 5), with equally sized wake panels. Wake panel size may vary on different flat portions of the wake. When ICNTRL = 7, card 2 must be followed by cards 3 to 5 described below.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICNTRL = 8; cosine wake panel spacing. The computer program will automatically panel each flat portion of the wake between adjacent wake breakpoints (YY, Z), with cosine-spaced wake panels. If the flat wake portion ends at the configuration centerline or lifting surface's junction with another surface, a quarter-circle distribution is used; otherwise a semicircle distribution is generated. This is generally the recommended value of ICNTRL for maximum accuracy (see Appendix A). When ICNTRL = 8, card 2 must be followed by cards 3 to 5 described below.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 2 | NTOT | (6-10) | Total number of wake vortex panels on current aerodynamic surface. Note that NTOT ≤ 50. Further, the grand total of the sum of all NTOT values for all NLINE surfaces must not exceed 50. |

For values of ICNTL = 7 or 8, cards 3 to 5 must be specified as follows:

| 3 | NBRK | (1-5) | Total number of wake breakpoints for the current surface. Note that NBRK equals the number of changes in wing dihedral, plus two; or the number of flat portions of the surface, plus one. |

| 4 | LSEG(1), I = 1, (1-25) ... , (NBRK-1) | User-specified numbers of wake panels on each of the (NBRK-1) flat portions of the current aerodynamic surface, beginning at the root. |
The following card is in a 6F10.0 format:

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Variable</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>YY(I), Z(I), PP(I), I = 1, ..., NBRK</td>
<td>(1-60)</td>
<td>Values of the Y, Z (in appropriate units), and ( \phi ) (in degrees) for breakpoints of the current aerodynamic surface, beginning at the root. Note that the left half of the assumed-symmetrical planform is input, so that Y becomes negative going root-to-tip, while Z is negative up (see fig. 1). Note also that the PP(I) value is the dihedral value, in degrees, inboard of breakpoint I; the root value of PP is therefore not needed.</td>
</tr>
</tbody>
</table>

After all NLLINE sets of geometry data have been input, the following cards are needed:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>CLDES</td>
<td>(1-10)</td>
<td>Desired lift coefficient, in F10.0 format.</td>
</tr>
<tr>
<td>6</td>
<td>SREF</td>
<td>(11-20)</td>
<td>Total configuration reference area (in appropriate units), in F10.0 format.</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>(5)</td>
<td>zero (0)</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>(10)</td>
<td>zero (0)</td>
</tr>
</tbody>
</table>

Card 7 signifies the end of input data for one configuration. Further configurations may follow card 7 beginning again with card 1. At the end of all configuration data for any one run, a final blank card must be included to signify the end of that run:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(5)</th>
<th>zero (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that when ICNTL equals 7 or 8 the breakpoint data specified includes YY, Z, and PP, which in effect overspecifies the wake geometry. This has not proved to be a problem, except that for more complicated configurations the entire F10.0 data field for each YY, Z, or PP value should contain significant figures for optimum accuracy of the input geometry.
APPENDIX C
OUTPUT DATA DESCRIPTION

The computer program prints out information of two general types: first, geometry data, both as input data and the calculated wake panel geometry, are printed. This information is followed by the minimum drag solution information, which includes the wake vortex sheet strengths, optimum bound circulation, induced drag coefficient, and induced drag efficiency factor. For a single planform, this solution information is printed for the Munk's criterion solution, followed by the same output for the direct optimization technique solution, while for configurations with more than one planform, only the Munk's criterion solution is computed and printed. In this Appendix, the output information for a configuration is described in the order in which it printed.

Geometry Data

For each lifting surface of the configuration the values of the wake breakpoints, from root to tip (see Appendix B), are listed. This is followed by the peripheral length of that surface. Next, the individual wake vortex panel corner points, dihedral angles (in radians), and panel semiwidths are listed for the entire configuration. Finally, reference quantities for the configuration are listed. In detail, the geometry data listed is as follows:

(YY(I), I = 1, ..., NBRK) Y coordinates of the breakpoints of each lifting surface, as input by the user, ordered from root to tip. (See figure 1 for positive coordinate directions.)
(Z(I), I = 1, ..., NBRK) Z coordinates of the breakpoints of each lifting surface, as input by the user, ordered from root to tip.
(PP(I), I = 1, ..., NBRK) Dihedral angles, $\phi$, just inboard of breakpoint I, in radians.
DTOT Total peripheral length of each lifting surface.
The above data is followed by the wake panel corner points and semiwidths actually used in the Trefftz plane calculations. First, the program lists whether equal or cosine spaced wake panels have been generated. This is followed by a table containing the following information:

<table>
<thead>
<tr>
<th>I; I = 1, ..., (NTOTT + 1)</th>
<th>Individual wake panel number, numbered from the tip of the first lifting surface to the root, followed by the tip-to-root numbering of wake panels on successive surfaces. NTOTT is the total number of wake panels for the entire configuration; NTOTT ≤ 50.</th>
</tr>
</thead>
<tbody>
<tr>
<td>YHH(I)</td>
<td>Y coordinate of outboard, or tipmost, corner of wake panel I.</td>
</tr>
<tr>
<td>ZHH(I)</td>
<td>Z coordinate of outboard corner of wake panel I.</td>
</tr>
<tr>
<td>PPP(I)</td>
<td>Dihedral angle, in radians, of wake panel I.</td>
</tr>
<tr>
<td>SNN(I)</td>
<td>Semiwidth of wake panel I.</td>
</tr>
</tbody>
</table>

Finally, the following reference quantities are listed:

<table>
<thead>
<tr>
<th>TOL2</th>
<th>The tolerance utilized in subroutine LOGS. Generally, the value of the tolerance utilized in SNTAN, TOL, will have the same value, unless changed by the user. The value of TOL2 should be small compared to the smallest value of SNN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLDES</td>
<td>The desired lift coefficient value.</td>
</tr>
<tr>
<td>SREF</td>
<td>The configuration reference area.</td>
</tr>
<tr>
<td>BSAVE</td>
<td>The configuration reference span, taken as twice the maximum absolute value of YHH.</td>
</tr>
<tr>
<td>ARAT</td>
<td>The configuration aspect ratio, defined as (BSAVE)²/SREF.</td>
</tr>
</tbody>
</table>

Solution Data

The output data for the minimum drag solution for the Munk's criterion solution consists, first, of a table of the following:
I

Individual wake panel number, numbered tip-to-root, as described above, for each lifting surface of the configuration.

BGAM(I)

Bound circulation value, $\Gamma/U$, at the outboard, or tipmost, corner of a wake panel I for minimum induced drag at a specified lift coefficient.

CDRAG(I)

Nondimensional bound circulation value for minimum induced drag, $\Gamma/\Gamma_0$, at the outboard corner of wake panel I. Note that for a wake consisting of a portion of a circular arc, CDRAG values are nondimensionalized by the $\Gamma_0$ value at the root of the planform.

AOPT(I, NTOTT + 1)

Wake vortex sheet strengths for minimum induced drag, $\gamma/U$, at the outboard corner of wake panel I.

GAM(I)

Nondimensional wake vortex sheet strength for minimum induced drag, $\gamma/\gamma_0$, at the outboard corner of wake panel I.

ETA

Nondimensional spanwise coordinate of outboard corner point of wake panel I, at which the above values are computed.

It is after this information, during the computation of the induced drag, that it is possible that a message "80 ENTERED" may be printed, to indicate a problem with wakes crossing one another, as discussed in Appendix A. This is followed by the induced drag coefficient, induced drag efficiency factor, and computed normal wash velocities as follows:

CD

Induced drag coefficient computed using optimum value vortex strengths, for Munk's criterion solution.

DIDEAL

Induced drag coefficient for a planar wing having the same projected span as the current configuration, evaluated at the same lift coefficient value.

WDBU

The ratio of the constant, $w_0$, appearing in the general statement of Munk's criterion [eq. (7)], divided by $U$.

DEFF

Induced drag efficiency factor, $k$, for the configuration, defined as the ratio of the induced drag for the planar wing divided by the computed induced drag for the configuration.
I  Wake panel number.

WDOWN  Computed induced normal velocity at the midpoint of wake panel I.

WOP  Induced normal velocity divided by the cosine of the dihedral angle, evaluated at the midpoint of wake panel I.

CDAPP  An approximate value of induced drag coefficient, evaluated by assuming $\Gamma$ and $\gamma$ are constant on each wake panel.

For single planforms, all of the above output, with the exception of the initial geometry data, is repeated for a second solution achieved using the direct optimization procedure for the same configuration. A sample output, as well as the input data deck, appear in Appendix D.
APPENDIX D
EXAMPLE OF INPUT AND OUTPUT DATA

Sample input data and output data are presented for one of the configurations of figure 6 of this report, where $d/h = 1.0$, $h/b = 0.355$. Input data and a sketch of the input wake shape appear on page 22, while the output data begins on page 23.
Input Data and Sketch of Wake for
Diamond Wing with End Plates;
d/h = 1.0, h/b = 0.355
Output Data for Diamond Wing with End Slates; d/h = 1.0, h/b = 0.355

GENERAL INPUT GEOMETRY

<table>
<thead>
<tr>
<th>SEGMT NO</th>
<th>Y</th>
<th>Z</th>
<th>PHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.499926</td>
<td>-0.169025</td>
<td>1.562070</td>
</tr>
<tr>
<td>2</td>
<td>-0.499658</td>
<td>-0.138363</td>
<td>1.562070</td>
</tr>
<tr>
<td>3</td>
<td>-0.499225</td>
<td>-0.088750</td>
<td>1.562070</td>
</tr>
<tr>
<td>4</td>
<td>-0.498793</td>
<td>-0.039137</td>
<td>1.562070</td>
</tr>
<tr>
<td>5</td>
<td>-0.498525</td>
<td>-0.008475</td>
<td>1.562070</td>
</tr>
<tr>
<td>6</td>
<td>-0.495383</td>
<td>-0.001093</td>
<td>-0.342102</td>
</tr>
<tr>
<td>7</td>
<td>-0.483185</td>
<td>-0.005436</td>
<td>-0.342102</td>
</tr>
<tr>
<td>8</td>
<td>-0.450089</td>
<td>-0.014017</td>
<td>-0.342102</td>
</tr>
<tr>
<td>9</td>
<td>-0.423689</td>
<td>-0.026623</td>
<td>-0.342102</td>
</tr>
<tr>
<td>10</td>
<td>-0.377557</td>
<td>-0.042944</td>
<td>-0.342102</td>
</tr>
<tr>
<td>11</td>
<td>-0.322720</td>
<td>-0.062578</td>
<td>-0.342102</td>
</tr>
<tr>
<td>12</td>
<td>-0.259637</td>
<td>-0.085042</td>
<td>-0.342102</td>
</tr>
<tr>
<td>13</td>
<td>-0.190161</td>
<td>-0.109783</td>
<td>-0.342102</td>
</tr>
<tr>
<td>14</td>
<td>-0.116002</td>
<td>-0.136191</td>
<td>-0.342102</td>
</tr>
<tr>
<td>15</td>
<td>-0.038987</td>
<td>-0.163161</td>
<td>-0.342102</td>
</tr>
<tr>
<td>16</td>
<td>-0.499926</td>
<td>-0.169025</td>
<td>-1.562070</td>
</tr>
<tr>
<td>17</td>
<td>-0.499658</td>
<td>-0.138363</td>
<td>-1.562070</td>
</tr>
<tr>
<td>18</td>
<td>-0.499225</td>
<td>-0.088750</td>
<td>-1.562070</td>
</tr>
<tr>
<td>19</td>
<td>-0.498793</td>
<td>-0.039137</td>
<td>-1.562070</td>
</tr>
<tr>
<td>20</td>
<td>-0.498525</td>
<td>-0.008475</td>
<td>-1.562070</td>
</tr>
<tr>
<td>21</td>
<td>-0.495383</td>
<td>-0.001093</td>
<td>-0.342102</td>
</tr>
<tr>
<td>22</td>
<td>-0.483185</td>
<td>-0.005436</td>
<td>-0.342102</td>
</tr>
<tr>
<td>23</td>
<td>-0.450089</td>
<td>-0.014017</td>
<td>-0.342102</td>
</tr>
<tr>
<td>24</td>
<td>-0.423689</td>
<td>-0.026623</td>
<td>-0.342102</td>
</tr>
<tr>
<td>25</td>
<td>-0.377557</td>
<td>-0.042944</td>
<td>-0.342102</td>
</tr>
<tr>
<td>26</td>
<td>-0.322720</td>
<td>-0.062578</td>
<td>-0.342102</td>
</tr>
<tr>
<td>27</td>
<td>-0.259637</td>
<td>-0.085042</td>
<td>-0.342102</td>
</tr>
<tr>
<td>28</td>
<td>-0.190161</td>
<td>-0.109783</td>
<td>-0.342102</td>
</tr>
<tr>
<td>29</td>
<td>-0.116002</td>
<td>-0.136191</td>
<td>-0.342102</td>
</tr>
</tbody>
</table>
Output Data for Diamond Wing with End Plates;
\( \frac{d}{h} = 1.0, \; \frac{h}{b} = 0.355 \) (continued)

<table>
<thead>
<tr>
<th>I</th>
<th>( S_N(I) )</th>
<th>.163616</th>
<th>.342102</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.00848</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>.02219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>.02743</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.02219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.00848</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.00326</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.00969</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>.01589</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>.02169</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>.02696</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>.03157</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>.03540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>.03835</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>.04037</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>.04139</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>.00848</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>.02219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>.02743</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>.02219</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>.00848</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>.00326</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>.00969</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>.01589</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>.02169</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>.02696</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>.03157</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>.03540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>.03835</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>.04037</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>.04139</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( TOL \; IN \; S_{NTAN} = \; .26057E-05 \)

\( DESIGN \; LIFT \; COEF. = \; .50000 \)

\( WING \; REFERENCE \; AREA = \; 1.00000 \)

\( REF \; WING \; SPAN = \; 1.00000 \)

\( ASPECT \; RATIO = \; 1.00000 \)
OPTIMUM LOADING USING MUNKS CRITERION

<table>
<thead>
<tr>
<th>SEGMT</th>
<th>BOUND CIRC</th>
<th>BGAM/AVE</th>
<th>SHED STRH.</th>
<th>GAM/AVE</th>
<th>ETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.</td>
<td>0.</td>
<td>0.30907E+01</td>
<td>0.31321E+02</td>
<td>1.00000E+01</td>
</tr>
<tr>
<td>2</td>
<td>33090E-01</td>
<td>29612E+00</td>
<td>81359E+00</td>
<td>82448E+01</td>
<td>99970E+00</td>
</tr>
<tr>
<td>3</td>
<td>62488E-01</td>
<td>55920E+00</td>
<td>51136E+00</td>
<td>51820E+01</td>
<td>99893E+00</td>
</tr>
<tr>
<td>4</td>
<td>87560E-01</td>
<td>78357E+00</td>
<td>40281E+00</td>
<td>40820E+01</td>
<td>99797E+00</td>
</tr>
<tr>
<td>5</td>
<td>10456E+00</td>
<td>93567E+00</td>
<td>36320E+00</td>
<td>36806E+01</td>
<td>99720E+00</td>
</tr>
<tr>
<td>6</td>
<td>10955E+00</td>
<td>98035E+00</td>
<td>22603E+00</td>
<td>22905E+01</td>
<td>99690E+00</td>
</tr>
<tr>
<td>7</td>
<td>11040E+00</td>
<td>98795E+00</td>
<td>34701E-01</td>
<td>35165E+00</td>
<td>98463E+00</td>
</tr>
<tr>
<td>8</td>
<td>11089F+00</td>
<td>99232E+00</td>
<td>15637E+00</td>
<td>15847E+00</td>
<td>94811E+00</td>
</tr>
<tr>
<td>9</td>
<td>11156E+00</td>
<td>99831E+00</td>
<td>26488E+00</td>
<td>26842E+00</td>
<td>88825E+00</td>
</tr>
<tr>
<td>10</td>
<td>11322E+00</td>
<td>10132E+01</td>
<td>50322E+00</td>
<td>50996E+00</td>
<td>80651E+00</td>
</tr>
<tr>
<td>11</td>
<td>11658E+00</td>
<td>10432E+01</td>
<td>74120E+00</td>
<td>75113E+00</td>
<td>70492E+00</td>
</tr>
<tr>
<td>12</td>
<td>12160E+00</td>
<td>10882E+01</td>
<td>84958E+00</td>
<td>86096E+00</td>
<td>58596E+00</td>
</tr>
<tr>
<td>13</td>
<td>12748E+00</td>
<td>11408E+01</td>
<td>81092E+00</td>
<td>82178E+00</td>
<td>45258E+00</td>
</tr>
<tr>
<td>14</td>
<td>13314E+00</td>
<td>11915E+01</td>
<td>66558E+00</td>
<td>67449E+00</td>
<td>30806E+00</td>
</tr>
<tr>
<td>15</td>
<td>13762E+00</td>
<td>12315E+01</td>
<td>44410E+00</td>
<td>45004E+00</td>
<td>15595E+00</td>
</tr>
<tr>
<td>16</td>
<td>0.</td>
<td>0.</td>
<td>0.30907E+01</td>
<td>0.31321E+02</td>
<td>1.00000E+01</td>
</tr>
<tr>
<td>17</td>
<td>33090E-01</td>
<td>29612E+00</td>
<td>81359E+00</td>
<td>82448E+01</td>
<td>99970E+00</td>
</tr>
<tr>
<td>18</td>
<td>62488E-01</td>
<td>55920E+00</td>
<td>51136E+00</td>
<td>51820E+01</td>
<td>99893E+00</td>
</tr>
<tr>
<td>19</td>
<td>87560E-01</td>
<td>78357E+00</td>
<td>40281E+00</td>
<td>40820E+01</td>
<td>99797E+00</td>
</tr>
<tr>
<td>20</td>
<td>10456E+00</td>
<td>93567E+00</td>
<td>36320E+00</td>
<td>36806E+01</td>
<td>99720E+00</td>
</tr>
<tr>
<td>21</td>
<td>10955E+00</td>
<td>98035E+00</td>
<td>22603E+00</td>
<td>22905E+01</td>
<td>99690E+00</td>
</tr>
<tr>
<td>22</td>
<td>11040E+00</td>
<td>98795E+00</td>
<td>34701E-01</td>
<td>35165E+00</td>
<td>98463E+00</td>
</tr>
<tr>
<td>23</td>
<td>11089F+00</td>
<td>99232E+00</td>
<td>15637E+00</td>
<td>15847E+00</td>
<td>94811E+00</td>
</tr>
<tr>
<td>24</td>
<td>11156E+00</td>
<td>99831E+00</td>
<td>26488E+00</td>
<td>26842E+00</td>
<td>88825E+00</td>
</tr>
<tr>
<td>25</td>
<td>11322E+00</td>
<td>10132E+01</td>
<td>50322E+00</td>
<td>50996E+00</td>
<td>80651E+00</td>
</tr>
<tr>
<td>26</td>
<td>11658E+00</td>
<td>10432E+01</td>
<td>74120E+00</td>
<td>75113E+00</td>
<td>70492E+00</td>
</tr>
<tr>
<td>27</td>
<td>12160E+00</td>
<td>10882E+01</td>
<td>84958E+00</td>
<td>86096E+00</td>
<td>58596E+00</td>
</tr>
<tr>
<td>28</td>
<td>12748E+00</td>
<td>11408E+01</td>
<td>81092E+00</td>
<td>82178E+00</td>
<td>45258E+00</td>
</tr>
<tr>
<td>29</td>
<td>13314E+00</td>
<td>11915E+01</td>
<td>66558E+00</td>
<td>67449E+00</td>
<td>30806E+00</td>
</tr>
<tr>
<td>30</td>
<td>13762E+00</td>
<td>12315E+01</td>
<td>44410E+00</td>
<td>45004E+00</td>
<td>15595E+00</td>
</tr>
<tr>
<td>31</td>
<td>13946E+00</td>
<td>12480E+01</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED

Output Data for Diamond Wing with End Plates;
\text{d/h} = 1.0, \text{h/b} = 0.355 \text{ (continued)}
Output Data for Diamond Wing with End Plates;
\[ \frac{d}{h} = 1.0, \quad \frac{h}{b} = 0.355 \] (continued)

<table>
<thead>
<tr>
<th>I</th>
<th>Downwash</th>
<th>( \frac{W}{\cos(\phi)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>2</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>3</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>4</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>5</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>6</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>7</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>8</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>9</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>10</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>11</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>12</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>13</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>14</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
</tbody>
</table>

CD calculated using subcritical and optimal loads using Munk CRIT = .47846E-01

CD for flat wing = .79577E-01

Ratio of zero dihedral downwash/\( \mu \) = .38317E+00

Induced drag efficiency for wings of equal span = .166319E+01
Output Data for Diamond Wing with End Plates;
\[ d/h = 1.0, \ h/b = 0.355 \] (concluded)

<table>
<thead>
<tr>
<th></th>
<th>.90242E-01</th>
<th>.95793E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>16</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>17</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>18</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>19</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>20</td>
<td>.83594E-03</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>21</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>22</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>23</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>24</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>25</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>26</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>27</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>28</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>29</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
<tr>
<td>30</td>
<td>.90242E-01</td>
<td>.95793E-01</td>
</tr>
</tbody>
</table>

APPROX CD USING SOLVED BOUND CIRCULATIONS AND WASHES AT SEG MIDPOINTS = .47909E-01
APPENDIX E

COMPUTER PROGRAM LISTING

This program has been written in FORTRAN IV language for the CDC series 6000 computer system with NOS1.3 operating system. Minor modifications may be necessary to achieve successful execution on other computers, as discussed in Appendix A. The following table is an index to the computer program listing:

<table>
<thead>
<tr>
<th>Name</th>
<th>Letter Designation</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM DNWASH</td>
<td>A</td>
<td>29</td>
</tr>
<tr>
<td>SUBROUTINE GAMCAL</td>
<td>B</td>
<td>47</td>
</tr>
<tr>
<td>SUBROUTINE WCAL</td>
<td>C</td>
<td>48</td>
</tr>
<tr>
<td>SUBROUTINE CCAL</td>
<td>D</td>
<td>50</td>
</tr>
<tr>
<td>SUBROUTINE CONCAL</td>
<td>E</td>
<td>51</td>
</tr>
<tr>
<td>SUBROUTINE SNTAN</td>
<td>F</td>
<td>52</td>
</tr>
<tr>
<td>SUBROUTINE LOGS</td>
<td>G</td>
<td>60</td>
</tr>
<tr>
<td>SUBROUTINE DRACAL</td>
<td>H</td>
<td>63</td>
</tr>
<tr>
<td>SUBROUTINE SIMEQ</td>
<td>I</td>
<td>65</td>
</tr>
</tbody>
</table>

The permanent file name of this program at NASA/Langley Research Center is DRG, stored under user number 496125E.
PROGRAM DNWASH

1

PROGRAM DNWASH(INPUT, OUTPUT, TAPE5, TAPE6, OUTPUT)

DIMENSION GAM5(5), GAM6(5), YHH5(5), ZH5(5), PPP(5)

DIMENSION DNW5(5)

DIMENSION AOPT(5,2)

DIMENSION AINT(6)

DIMENSION CDRA5(5)

DIMENSION AS1(5,5), PS1(5,1), PIVOT5(5)

DIMENSION YYY(10), Z(10), P(10), LSE0(10), DTME5(10)

DIMENSION PEPF(10), H(10), GSUM(10)

DIMENSION CLP(5), SEQ(10), DTO10

COMMON /FILL/ TOL, TOL2, TOL3

COMMON /SEG/ SNS(5)

COMMON /DIOPT/ T1(53,53), T2(53,53), T3(53,53), T4(53,53), T5(53,53)

1, T6(53,53)

COMMON /FEN/ N5PT10, MNL1

PI = 4. * ATAN1.1)

DTP = PI / 100

C PROGRAM WRITTEN TO IMPLEMENT TIEFFFtz PLAN~ DRAG


C

20

C PROGRAM WRITTEN BY DR. JOHN K. KUHLMAN, DEPT MECHANICAL

C ENGINEERING AND MECHANICS, OLD DOMINION UNIVERSITY, NOR-

C FOLK, VA 23504, UNDFR NASA LANGLEY GRANT NSG-1357.

C

25

C THEORY ASSUMES A TWO DIMENSIONAL ADVANCED PANEL MODEL OF

C THE UNDISTORTED, INTERACTING WAKES OF MULTIPLE LIFTING

C SURFACES.

C

30

C WAKE VORTEX SHEET STRENGTHS ARE ASSUMED TO VARY IN A

C PIECEWISE LINEAR FASHION. ANALYTICAL EXPRESSIONS FOR

C INDUCED NORMAL VELOCITY, ROUND CIRCULATION, INDUCED DRAG,

C AND LIFT ARE DEVELOPED IN CR-3154 IN TERMS OF THE ASSUMED

C WAKE MODEL.

C

35

C THESE EXPRESSIONS ARE USED TO OBTAIN MINIMUM DRAG WAKE VORTEX

C SHEET STRENGTHS. ROUND CIRCULATION DISTRIBUTIONS, AND CO

C VALUES FOR MINIMUM DRAG AT A GIVEN LIFT, FOR NONPLANAR MULTIPLE

C INTERACTING LIFTING SURFACES, USING BOTH MUNK'S CRITERION AND

C A DIRECT OPTIMIZATION PROCEDURE.

C

40

C IC=5 FOR GENERAL INPUT OF NON-PLANAR WING CASES
PROGRAM DUNKASH

C THFN MUST INPUT TRIPLES OF YNN, ZNN, PNP FOR EACH CORNER POINT
C OF EACH SEGMENT, STARTING AT WING TIP
45
C ICTRL=6 FOR CIRCULAR ARC WING, CONE, TR 139
C ICTRL=7 FOR WING OF ARBITRARY PHI AND ETA-EQUAL SPACING
C ICTRL=8 FOR WING OF ARBITRARY PHI AND ETA-COSINE SPACING
C
C NNOT=NUMBER OF SEGMENTS ON SEMISPAN, BOTH FOR EQUAL AND COSINE SPA
C
50
C READ AN INTEGER VARIABLE NLINE =RENT LIFTING LINES NEEDED TO MAKE
C NNOTT = NUMBER OF SEGMENTS ON NLINE
C NS = START NUMBER ON EACH NLINE

C
55
1 READ (5,131) NALLINE
IF (NALLINE.GT.0) GO TO 129
MNOTT = 0
LSTART = 1
DTOT = 0.0
60 WRITE (6,130)
WRITE (6,135)
2 CONTINUE
3 READ (5,131) ICTRL,NNOTT
IF (ICCTRL.GT.0) GO TO 1
MNOTT = ICTRL+1
NSPT(LSTART) = NS
NNOTT = NNOTT+NOTT
70 ISTOP = 1
IIHAG = I

C
75
C CALCULATE DUNKASH AT SEGMENT MIDPOINTS
C
3 CONTINUE
80 WRITE (6,133)
DO 11 I=1,NNOTT
WDOWN = 0.
DO 10 J=1,NNOTT
S = SNN(J)
10 CONTINUE
11 CONTINUE
129
GO TO 129
GO TO 1
85 CALL CCAL (I,J,YHM,ZHM,PPPH,SA,UA,DU,FF,GG,EE,AA,AK,PP,TT,UU,WW) A 85
CALL CONCAL (AA,RAH,FF,GG,SA,AR,CA,DA,FA,GA,CA,CL,CH,CN,CO,CP,1) A 86
IF (R*EQ.O.)  GO TO 4 A 87
P = 2.*c(ATHAN(C/ARS(PP)) - ATAN(D/ARS(RR)) )/ARS(RR)) A 88
GO TO 5 A 89

90 4 CONTINUE A 90
P = 2.(/FF-2.*S)-2.(/FF+2.*S) A 91
5 CONTINUE A 92
IF (UU.EQ.O.)  GO TO 6 A 93
Q = 2.*(ATAN((AJ+2.*S)/ARS(UU)) - ATAN((AJ-2.*S)/ARS(UU)) )/ARS(UU) A 94
GO TO 7 A 95

6 CONTINUE A 96
Q = 2.(/AJ-AJ-2.)/(AJ+2.*S) A 97
7 CONTINUE A 98
Z1 = (S*S+FF*S+GG)/(S*S+FF*S*GG) A 99
Z2 = (S*S+AJ*S+AK)/(S*S+AJ*S+AK) A 100
A11J = (1.*S+AI3J*S+AK)*LOG(Z2)/(Z*PI) A 101
A31J = (CL*P*A31J*S+CO*ALOG(Z2))/(Z*PI) A 102
CALI CONCAL (DD,EE,AA,AK,SA,AR,CA,DA,FA,GA,CA,CL,CH,CN,CO,CP,2) A 103
A21J = -(Q*S+EE*AI3J*LOG(Z2))/Z*PI) A 104
105
A41J = -(CL*Q*A*EE*S*CO*ALOG(Z2))/(Z*PI) A 105
GO R K =1,NLIE A 106
KR = K+1 A 107
KCHK = MPS(KK)-1 A 108
IF (I,F,KCHK)  GO TO 9 A 109
8 CONTINUE A 110
WDOWN = WDOWN+.S*(GAM(J+1)*GAM(J)) * (A11J+A21J)*.S*(GAM(J+1)-GAM(J)) A 111
A31J*A41J) A 111
GO TO 10 A 112
9 CONTINUE A 113
10 CONTINUE A 115
C
C DOWNWASH AT WING IS 5 WASH AT MINUS INFINITY A 116
C
C WDOWN = WDOWN/.P A 117
C WOP = WDOWN/COS(PP(I)) A 118
C WTRI = (.134*I+1)*WDOWN,WOP A 119
C DOWNWASH = WDOWN A 120
C
11 CONTINUE A 121
125 IF (ISTOP,6F.*7)  GO TO 124 A 122
ISTOP = ISTOP+1 A 123
120
PROGRAM DNWASH

GO TO 14

C C GENERAL GEOMETRY CALCULATIONS
C

130 CONTINUE
READ (5,136) YHH(I),ZZH(I),PPP(I),NS,NTOT
HOT = ABS(YHH(NS)+SNH(NS)*COS(PPP(NS)))
H(START) = 2.*HOT

135 DTOT = 0.
DO 13 I=1,NTOT
DTOT = DTOT+SNH(I)
13 CONTINUE
DTOT = 2.*DTOT

WRITE (6,133) DTOT
DTOF(START) = DTOT
DTOF = DTOF+DTOT

GO TO 19

140 CONTINUE

C C GEOMETRY FOR CIRCULAR ARC AIRFOIL
C

145 CONTINUE
NTOT = NTOT+1
READ (5,136) HFT,THFT,HOT
THFT = THFT*DFT
H(START) = 2.*HOT
D = HFT*HOT
R = HOT/SIN(THFT)

150 DTHTA = THFT/FLOAT(NTOT)
DO 16 I=1,NTOT
YT = -R*SIN(THFT-FLOAT(I-1)*DTHTA)
ZT = -R*COS(THFT-FLOAT(I-1)*DTHTA)

155 VT(I) = -R*SIN(THFT-FLOAT(I)*DTHTA)
II = NS-I+1
PPP(I) = ATAN((ZT-ZT1)/(YT-YT1))
SNH(I) = 5*SQRT((YT-YT1)**2+(ZT-ZT1)**2)
YHH(I) = 5*(YT+YT1)

160 ZHH(I) = 5*(ZT+ZT1)

16 CONTINUE

DO 17 I=NS,NTOT

17 CONTINUE

PROGRAM DNWASH

170 CONTINUE
DTOT = DTOT + SHN(I)
175 CONTINUE
IF (NS, NE, 1) GO TO 18
UT01 = UT01
18 CONTINUE
GO TO 29

C 29 CONTINUE
C GEOMETRY CALCS FOR WAKE MADE OF STRAIGHT SEGMTS(CONCTD)
C
185 READ (5, 137) NBRK
NBR = NBRK - 1
READ (5, 137) (LSF(I), I = 1, NBR)
READ (5, 136) (YY(I), I = 1, NBRK)
DO 21 I = 1, NBRK
WRITE (6, 132) YY(I), 7(I), PP(I), I = 1, NBRK
DO 22 I = 1, NBR
21 WRITE (6, 132) YY(I), 7(I), PP(I)
22 PERIF(I) = SQRT((7(I) - 7(I-1)**2 + (YY(I) - YY(I-1))**2)
195 DT0T = 0
DO 23 I = 1, NBR
23 DT0T = DT0T + PERIF(I)
WRITE (6, 133) DT0T
190 DO 24 I = 1, NBRK
WRITE (6, 133) DT0T
200 IF (NS, NE, 1) GO TO 24
DO 25 I = 1, NBRK
25 PP(I) = NTH*PP(I)
K = LSTART/I, 2*ARS(NBRK)
IF (LCTRL.EQ.,F) GO TO 30
DO 26 I = 1, NBRK
SEG(I) = 0.5*PERIF(I)/LSF(I)
210 CONTINUE
PROGRAM DNWASH

\[ S = \text{SEQ}(\text{NRK}) \]
\[ \text{SN}(\text{NS}) = S \]
\[ \text{PP}(\text{NS}) = \text{PP}(\text{NRK}) \]
\[ \text{YH} = \text{Y}(\text{NRK}) * S * \text{COS}(\text{PP}(\text{NS})) \]
\[ \text{ZH} = \text{Z}(\text{NRK}) * S * \text{SIN}(\text{PP}(\text{NS})) \]
\[ \text{N} = \text{NS} \]
\[ \text{DO 20 J} = 1, \text{NRK} \]
\[ \text{NSEG} = \text{N} \text{LSEG}(\text{NRK} - J) - 1 \]
\[ \text{II} = \text{N} + 1 \]
\[ \text{LL} = \text{NRK} - J \]
\[ \text{LH} = \text{LL} + 1 \]
\[ \text{DO 27 I} = \text{II}, \text{NSEG} \]
\[ \text{SN} = \text{SE0}(\text{LH}) \]
\[ \text{YH} = \text{Y}(\text{I}) * 2 * \text{SE0}(\text{LH}) * \text{COS}(\text{PP}(\text{N})) \]
\[ \text{ZH} = \text{Z}(\text{I}) * 2 * \text{SE0}(\text{LH}) * \text{SIN}(\text{PP}(\text{N})) \]
\[ \text{PP} = \text{PP}(\text{I}) \]
\[ \text{LL} = \text{NRK} - J \]
\[ \text{DO 20 I} = 1, \text{NRK} \]
\[ \text{CONTINUE} \]

\[ \text{GO TO 42} \]

C
C COSINE SPACING CALCULATIONS
C
C CONTINUE

\[ \text{IF } (\text{YH}(\text{I}) < 0) \text{ GO TO 32} \]
\[ \text{DO 31 I} = 1, \text{NRK} \]
\[ \text{OBTTHF}(\text{I}) = \text{PI} / (\text{FLOAT}(\text{LSFG}(\text{NRK} - I))) \]
\[ \text{GO TO 35} \]
\[ \text{DO 34 I} = 1, \text{NRK} \]
\[ \text{IF } (\text{I} > 0) \text{ GO TO 33} \]

C
C THE VARIATION OF THETA IS NAMED ARRAY OBTTHF(I) IN ICNTRL=TH
C
\[ \text{OBTTHF}(\text{I}) = \text{PI} / (\text{FLOAT}(\text{LSFG}(\text{NRK} - I))) \]
\[ \text{GO TO 34} \]

\[ \text{CONTINUE} \]

\[ \text{IF } (\text{NRK} > 0) \text{ GO TO 35} \]

\[ \text{CONTINUE} \]

\[ \text{CONTINUE} \]

\[ \text{CONTINUE} \]

\[ \text{CONTINUE} \]

\[ \text{CONTINUE} \]
PROGRAM DNASH

SNH(NS) = 0.5*PEIF(1)*(1.-COS(DTETE(1)))
GO TO 36

255

35 FM = 0.5*PEIF(NHR)
SNH(NS) = 0.5*RH*(1.-COS(DTETE(1)))
36 PPP(NS) = PP(NHRK)
NN = 0
DO 40 I = 1, NPP
NN = NN + LSF(NHRK - I)
IF (I, EQ, NPP, AND, YY(1), EQ, 0.) GO TO 37
FM = 0.5*PEIF(NHRK - I)
GO TO 38

260

37 CONTINUE

265

NN = PEIF(NHRK - I)

39 CONTINUE

LL = LSF(NHRK - I)
DO 39 J = 1, LL
IF (I, EQ, 1, AND, J, EQ, 1) GO TO 39
NN = NN + J
KM = NN + J
PPP(KM) = PP(NHRK + 1 - I)
SNH(KM) = 0.5*RH*(COS(FLOAT(J)*DTETE(1)) - COS(FLOAT(J-1)*DTETE(1))

275

SNH(KM) = ARS(SNH(KM))

39 CONTINUE

40 CONTINUE

YHH(NS) = YY(NHRK) + SNH(NS)*COS(PPP(NS))
ZHH(NS) = Z(NHRK) + SNH(NS)*SIN(PPP(NS))

280

NSEG = NOT0
DO 41 I = 2, NSEG
I = NS - 1 + I
YHH(I) = YHH(I-1) + SNH(I-1)*COS(PPP(I-1)) + SNH(I)*COS(PPP(I))
ZHH(I) = ZHH(I-1) + SNH(I-1)*COS(PPP(I-1)) + SNH(I)*SIN(PPP(I))

285

41 CONTINUE

42 CONTINUE
IF (LSTART.EQ., NLLLINF) GO TO 43
LSTART = 1
LSTART = 1
GO TO 2

290

43 CONTINUE
IF (LCNTRL.EQ, 6) WRITE (6, 139)
IF (LCNTRL.EQ, 7) WRITE (6, 140)
IF (LCNTRL.EQ, 8) WRITE (6, 141)
LN = NLLLINF + 1
NSPT(I) = NTOTT+1
WRITE (*,147)
WRITE (*,148) (1,YHM(I),I=N1M(I),I=1,NTOTT)
WRITE (*,144)
WRITE (*,145) (1,YHM(I),I=1,NTOTT)

C READ, WRITE CL SREF, ETC
C
READ (5,136) CLDES, SREF
BSAVE = h(1)

IF (NLINF.EQ.0,1) GO TO 45
DO 44 I=1,NLINF
RTFM = P(I)
44 BSAA V = AMAX1(BSAVE,RTFM)

C CONTINUE
ARAT = BSAVE*BSAVE/SREF
SMIN = SREH(1)
Do 46 J=2,NTOTT
STEM = SREH(I)
46 SMIN = AMIN1(SMIN,STEM)

IF (IDRAG.LT.0,7) GO TO 52
WRITE (6,146) CLDES, SREF, BSAVE, ARAT
WRITE (*,149)
IDRAG = IDRAG+1
C
SET UP ALL A FOR MINK CRITERION OPTIMIZATION
DO 49 I=1,NTOTT
DO 48 J=1,NTOTT
48 CALL WCAL (I,J,NTOTT,YHM, ZHM, PPP, AOPT(I,J))
AOPT(I,J) = AOPT(I,J)*2./SREF

C CONTINUE
49 CONTINUE
TL = NTOTT+1
DO 50 I=1,NTOTT
SIN(I,1) = COS(PPP(I))
50 CONTINUE
PROGRAM DNWASH

AOPT(I,J) = COS(PXP(I,J))

50 CONTINUE

DO 51 I=1,NTOTT
DO 51 J=1,NTOTT
ASIM(I,J) = AOPT(I,J)

51 CONTINUE

IL1 = NTOTT

C ABOVE IL1 IS TEMPORARY SET L4 NTOTT

GO TO 77

52 CONTINUE

C SET UP ALL A THROUGH F FOR ALL I,J TO DO DIRECT OPTIMIZATION

IF (M(N,L,N,F,I,J)) GO TO 7

NTOTT2 = NTOTT+2

DO 53 I=1,NTOTT2
DO 53 J=1,NTOTT2

T1(I,J) = 0.
T2(I,J) = 0.
T3(I,J) = 0.
T4(I,J) = 0.
T5(I,J) = 0.

53 CONTINUE

DO 54 I=1,NTOTT
I1 = I+1
DO 54 J=1,NTOTT
J1 = J+1

CALL PRACAL(I,J,YHH,THH,PPP,AINT)

T1(I1,J1) = AINT(1)
T2(I1,J1) = AINT(2)
T3(I1,J1) = AINT(3)
T4(I1,J1) = AINT(4)
T5(I1,J1) = AINT(5)
T6(I1,J1) = AINT(6)

54 CONTINUE

C DIRECT OPTIMIZATION MDS

55 CONTINUE

DO 56 I=1,NTOTT
I1 = I+1
DO 56 J=1,NTOTT

56 CONTINUE


```
J1 = J1 + 1
AOPT1(J,J) = T3(I1+J1)-T4(I1+J1)-T5(I1+J1)+T6(I1+J1)+T3(I1+J1)-T4
AOPT(I,J) = AOPT(I,J)+1.5*SNH(I)*T(I1+J1)-T2(T1(I1+J1))+1.5*SNH(I)*T(I1+J1-1)+T2(I1+J1-1)

385 IF (I1,F0.1) GO TO 55
AOPT(I,J) = AOPT(I,J)+1.5*SNH(I-1)*T(I1-I1-J1+1)-T2(T1(I1-I1-J1+1),J1-1)+T1(I1-I1-J1-1)
55 CONTINUE
AOPT(I,J) = AOPT(I,J)*0.25
56 CONTINUE
DO 64 J=1,NNTOTT
I1 = I1+1
IF (I1,F0.1) GO TO 60
SNCON = 0.5*SNH(I)+SNH(I-1)
TEMP = 0.
DO 59 J=1,NNTOTT
J1 = J1 + 1
TEMP = 1.5*SNH(I-1)*T(I1+J1-I1-J1+1)-T2(T1(I1-I1-J1+1),J1-1)+T1(I1-I1-J1-1)
IF (I1,EQ.II) TEMP = TEMP
IS = I1+1
DO 57 NP=IS,NNTOTT
JP = JP+1
IP1 = IP1+1
TEMP = TEMPP+SNCON*(T1(IP1,J1)-T2(IP1,J1)+T1(IP1,J1-1)+T2(IP1,J1-1))
57 CONTINUE
58 CONTINUE
AOPT(I,J) = AOPT(I,J)-TEMP
AOPT(I,J) = AOPT(I,J)/SREF
59 CONTINUE
GO TO 63
60 CONTINUE
SNCON = 1.5*SNH(I)
DO 62 J=1,NNTOTT
J1 = J1 + 1
TEMP = 0.
DO 61 NP=J1,NNTOTT
IP1 = IP1+1
TEMP = TEMPP+SNCON*(T1(IP1,J1)-T2(IP1,J1)+T1(IP1,J1-1)+T2(IP1,J1-1))
61 CONTINUE
AOPT(I,J) = AOPT(I,J)-TEMP
AOPT(I,J) = AOPT(I,J)/SREF
62 CONTINUE
```
63 CONTINUE
64 CONTINUE
DO 65 J=1,NTOTT
DO 65 J=1,NTOTT
DO 65 J=1,NTOTT
425 T1(I,J) = AOPT(J,J)
DO 66 I=1,NTOTT
DO 66 J=1,NTOTT
ANPT(I,J) = T1 * (ANPT(I,J) + T1(I,J))
66 CONTINUE
IL = NTOTT+1
II = IL
II = I+1
DO 67 I=1,II
ANPT(I,IL2) = 0.
67 CONTINUE
ANPT(IL1,IL2) = CLDFS
ANPT(IL1,1) = P*COS(PPP(1))*SNN(1)**2/(3.*SREF)
DO 68 I=2,NTOTT
ANPT(IL1,1) = (2*COS(PPP(1))*SNN(1)**2+COS(PPP(I-1))*SNN(I-1)**2)*
14./(3.*SREF)
69 CONTINUE
SUMX = 0.
DO 69 I=2,NTOTT
SUMX = SUMX*COS(PPP(1))*SNN(1)
69 CONTINUE
ANPT(IL1,1) = ANPT(IL1,1) + (4./SREF)*SNN(1)*SUMX
DO 72 I=2,NTOTT
SUMX = COS(PPP(I))*SNN(I)*SNN(I-1)
II = I+1
IF (II<0,HTOTT) GO TO 71
DO 70 J=1,NTOTT
SUMX = SUMX*COS(PPP(J))*SNN(J)*SNN(I-1)
70 CONTINUE
71 CONTINUE
ANPT(IL1,1) = ANPT(IL1,1) + (4./SREF)*SUMX
72 CONTINUE
C
DO 73 I=1,NTOTT
ANPT(IL1,1) = T1 * ANPT(IL1,1)
73 CONTINUE
DO 74 I=1,NTOTT
ANPT(IL1,1) = ANPT(IL1,1)
74 CONTINUE
   DO 75 I=1,IL1
   465
75 CONTINUE
   CLP(I) = AOPT(I,IL1+1)
   470
76 CONTINUE
   WRITE (*,147)
   475
   WRITE (*,148)
   CALL SIMQ (ASIM*IL1,RSIM1,DET,IPIVOT,51,ISCALE)
   480
   IL1 = NTOIT+1
   SET IL1 BACK TO NTOIT+1
   IF (IPRGT.NE.3) GO TO A1
   CLCHK = 0.
   DO 79 I=1,NTOIT
   AOPT(I,IL1) = RSIM(I,1)
   CLCHK = CLCHK+CLP(I)*AOPT(I,IL1)
   485
   CONTINUE
   WRITE (*,150) CLCHK
   490
   WRITE (*,152) I,AOPT(I,IL2)
   P1 CONTINUE
   WRITE (*,151)
   495
   C
   C CALCULATE ROWN CIRCULATIONS AND
   C CALCULATE CINN CHIN DOWNWASH DIVIDED BY FRESTREAM U
   C
   NTOIT = NTOIT+1
   SUMGAM = 0.
   AOPT(NTOIT,IL1) = 0.
   500
   IF (IPRGT.NE.3) GO TO A3
   DO 92 I=1,IL1
   AOPT(I,1L) = AOPT(I,IL2)
   AOPT(I,IL1) = 0.
   505
   B3 CONTINUE
DO 84 I = I + 1, NTO1
CALL GAMCAL (I, IL, SNN, AOPT, RGAM0)
RGAM(I) = RGAM0

84 CONTINUE
IF (NLLINE, FO, 1) GO TO 90
NLI = NLLINE - 1
NI = Ni + 1
JF = NSPT(I) - 1

*SUM(I) = RGAM(JF) * AOPT(JF, IL) * SNN(JE) *

85 CONTINUE
DO 89 I = I + 1, NLI
JF = NSPT(I) - 1
YT = YH(JF) * SNN(JF) * COS(PPP(JF))
IF (ARS(YT, 1, T, 0) > 0.0001) GO TO 89
JF = NSPT(I) - 1
DO 86 J = I, JF
JS = J
IF (YT < YH(J)) GO TO 87

86 CONTINUE
A7 CONTINUE
DO 94 J = JS, JF
RGAM(J) = RGAM(J) * SUM(I)

88 CONTINUE

90 CONTINUE
SUMGAM = 0.
DO 94 I = I + 1, NTO1
NI = Ni + 1
JCHR = Ni + 1
JF = NSPT(JCHR) - 1
IF (I, FO, JCH) GO TO 92
SUMGAM = SUMGAM + COS(PPP(I)) * (SNN(I) ** 2 * (AOPT(I, 1, IL) + 2. * AOPT(I, 1))

91 CONTINUE
SUMGAM = SUMGAM + COS(PPP(I)) * (SNN(I) ** 2 * (AOGA(I) / 3. * SNN(I) * RGAM(I)))
GO TO 93

92 CONTINUE
SUMGAM = SUMGAM + COS(PPP(I)) * (SNN(I) ** 2 * (AOP1, IL) / 3. * SNN(I) * RG)

94 CONTINUE

A4
94 CONTINUE
IF (MLINE,ME,1) GO TO 95
JF = NSPT(2) - 1
95 CONTINUE

550

555

560

96 AOPT(I,1L) = AOPT(I,1L)*WDRU
97 CONTINUE

565

C HESCALF ROUND SHEET STRENGTHS TO CALCULATE AVG NON-DIM VALUES
C 566
C AOPT(I,1L) = OPT SHEET STRENGTHS
C RGAM(I) = OPT ROUND CIFC VALUES
C CDRAG(I) = OPT NON-DIM ROUND CIRC
C GAM(I) = OPT NON-DIM SHEET VALUES
C
570

SUBGAM = 0.
SURAM = 0.
NS = NSPT(2) - 1
DO 100 J = 1,NS
TFNP = AOPT(I,J,1L)
DO 99 J = 1,MLINE
JJ = J - 1
JFMK = NSPT(JJ) - 1
IF (I,IF,JFMK) GO TO 98
TEMP = 0.
98 CONTINUE
99 CONTINUE

580

585

SUBGAM = SURAM*SN1(I) + (TFNP*AOPT(I,1L))
SUBGAM = SUBGAM/2.*SN1(I) + (RGAM(I) + (SN1(I)/12.)*(4.*TFNP*R*AOPT(I,1L))

100 CONTINUE
SUBGAM = SUBGAM/WDRU
SUBGAM = SUBGAM/UT0(I,1)
NS = NSPT(2) - 2
PROGRAM DNASH
590 NS1 = NS1+1
   DO 103 I=I+1,NS
   IF (ICTHI .EQ. 0) GO TO 101
   CDCP(I+1) = RGAM(I+1)/SUBGAM
   GO TO 102
101 CONTINUE
   CDCP(I+1) = RGAM(I+1)/RGAM(NS1)
102 CONTINUE
103 CONTINUE
   JL = NSPT(I)+1
   CDCP(NTOT1) = RGAM(JL)*AOPT(JL)*SNN(JL)
600 CDCP(NTOT1) = CDCP(NTOT1)/SUBGAM
   DO 104 J=1,NTOTT
   GAM(J) = AOPT(J+1)/SUBGAM
104 CONTINUE
   CDCP(I) = 0.
605 GAM(NTOT1) = 0.
   IF (MILLER+1) GO TO 109
   DO 109 I=2,MINLINE
610 JS = NSPT(I)
   II = I+1
   JF = NSPT(II)+1
   SUBGAM = 0.
   DO 108 J=JS,JF
620 TEMP = AOPT(J+1)*IL
   IF (JF+1) GO TO 105
   TEMP = 0.
615 TEMP = TEMP+SUBGAM*(RGAM(J)*(SNN(J)*SNN(J)/12.)*(4.*TEMP+B*AOPT(J)*IL))
105 CONTINUE
621 SUBGAM = SUBGAM+TEMP
   IF (I+1) GO TO 101
   DO 107 J=JS,JF
622 CDCP(J) = RGAM(J)/SUBGAM
107 CONTINUE
106 CONTINUE
623 CONTINUE
105 CONTINUE
   DO 110 I=1,NTOTT
624 IF (I+1) GO TO 110
625 FTA = 2.*((TH(1)+SNI(I)*COS(PPP(I)))/SAVE
   GO TO 111
630 110 CONTINUE
CINDUCFD DRAG CALC USING ANALYTICAL INT OF WN TIMES ROUND CIRC
C
COM = 0.
NLL = NLL+1
DO 119 J=1,NNTT
DO 113 K=2,NLL
ICHK = NSPT(K)-1
IF (ICHK,J,NLL) GO TO 114
113 CONTINUE
XI = RGAM(I)*.25*SNN(I)*AOPT(I+1,IL)+3.*AOPT(I,IL)
YI = .5*(AOPT(I+1,IL)+AOPT(I,IL))
ZI = .5*(AOPT(I+1,IL)-AOPT(I,IL))
GO TO 115
114 CONTINUE
XI = RGAM(I)*.75*SNN(I)*AOPT(I,IL)
YI = .5*AOPT(I,IL)
ZI = -YI
115 CONTINUE
DO 119 J=1,NNTT
DO 116 K=2,NLL
JCHK = NSPT(K)-1
IF (J,F0,JCHK) GO TO 117
116 CONTINUE
YJ = .5*(AOPT(J+1,IL)+AOPT(J,IL))
ZJ = .5*(AOPT(J+1,IL)-AOPT(J,IL))
GO TO 119
117 CONTINUE
YJ = .5*AOPT(J,IL)
/ZJ = -YJ
118 CONTINUE
CALL DRFCAL (I,J,YHH,YHC,PP,APT)
CDI = XI*YH*APT(1)+XI*YH*APT(2)+YI*YH*APT(3)+YI*YH*APT(4)+ZI*Y
1*APT(5)+ZI*YH*APT(6)
CO = CDI-CDI
119 CONTINUE
CII = CDI*2.*SHL
IF (IDRAG,EQ,3) GO TO 120
PROGRAM ONWASH

WRITE (*,154) CD
GO TO 121

120 CONTINUE
WRITE (*,153) CD
121 CONTINUE
DINAL = .5*REF*CLS**2/(PI*(HSAVE)**2)
IF (INPAG,EQ,3) GO TO 122
WRITE (*,155) DINAL,WDHAL

122 CONTINUE
DEFF = DINAL/CU
WRITE (*,156) DEFF
ISTOP = [ISTOP+1]
DO 123 I=1,NTRI

123 GAM(I) = A0PT(I,II)
GO TO 3

124 CONTINUE
CDAPP = 0.

DO 125 J=1,NTRI
DO 126 J=1,1,1
JG = J+1
CQHIL = NSFJ(JG) - 1
IF (I,F0,JCQHIL) GO TO 126

125 CONTINUE
CDAPP = [CDAPP+2.*SNH(I)*ONWSH(I)*(GAM(I)+.25*A0PT(I,IL)+3.*A0P(I,II))*SNH(I)]
GO TO 127

126 CONTINUE
CDAPP = CDAPP+2.*SNH(I)*ONWSH(I)*(GAM(I)+.75*A0PT(I,IL)*SNH(I))

127 CONTINUE

128 CONTINUE
CDAPP = 4.*CDAPP/SHF
WRITE (*,158) CDAPP

IF (INPAG,EQ,3) GO TO 2
GO TO 41

129 CONTINUE

130 FORMAT(1H1)
131 FORMAT(515)
132 FORMAT(25X,3F15.5)
133 FORMAT(//33X,1H1,7X,8H00K,11X,10H0W/COS(PHI)/)
134 FORMAT(30X,15.2F15.5)
135 FORMAT(//30X,22HF0NDR1 INPUT GEOMETRY/)
715 136 FORMAT(F10.0) 715
137 FORMAT(10I5) 715
138 FORMAT(30X,13HTOTAL PLANFORM PERIPHERAL LENGTH=,F15.5/) 715
139 FORMAT(/30X,17HCIRCULAR ARC WING/) 715
140 FORMAT(30X,21HEQUAL SEGMENT SPACING/) 715
720 141 FORMAT(10X,22HCOSINES SEGMENT SPACING/) 720
142 FORMAT(/24X,FHSFGPT NO.X,1MY+15X,1HZ+13X,3HPI/) 721
143 FORMAT(25X,15.6/) 722
144 FORMAT(27X,1HCHSWHE(1)) 723
145 FORMAT(25X,15,F12.5/) 724
725 146 FORMAT(/// 25X,20HDESIGN LIFT COLF. = ,F10.5//,25X,22HWING REFEREN 725
ICE AREA = ,F10.5//,25X,16HHREF WING SPAN = ,F10.5//,25X, 726
15HSHAPFCT RATIO = ,F10.5/) 727
147 FORMAT/// 25X,10F10.3/) 728
148 FORMAT/// 25X,54HDIRECT OPTIMIZATION USING ANALYTICAL EXPRESSION F 729
10 FOR CI./// 730
149 FORMAT/// 25X,34HDIRECT OPTIMIZATION USING MUNKS CRITERION//) 731
150 FORMAT/// 23X,4HLIFT COEFF CALCULATED FROM CLP AND SOLVED GAMS. 732
1E13.5/) 733
735 151 FORMAT/// 22X,5H5FGMT,3X,10HBOUND CIRC,2X,8HHAAM/AVE,4X,10HSHED STR 734
1H,5X,7IGAM/AVE,4X,1H ICFA/) 735
152 FORMAT/// 20X,15,5F13.5/) 736
153 FORMAT/// 25X,17DCALCULATED USING SUB DRACAL AND LOADS FROM DIR 737
1ECT OPTIMIZATION = ,F15.5/) 738
740 154 FORMAT/// 25X,4H4DCALCULATED USING SUB DRACAL AND OPTIM LOADS US 739
1ING MUNK (K=15.5/) 740
155 FORMAT(25X,19H4CD FOR FLAT WING = ,E15.5/25X,3HSHAPE RATIO OF ZERO DIHED 741
1IPAL DOWNHST = ,E15.5/) 742
156 FORMAT/// 25X,4HINDUCED DRAG EFFICIENCY FOR WINGS OF EQUAL SPAN = 743
1E15.6/) 744
745 157 FORMAT/// 25X,14HTOL IN SHTAN = ,E15.5/) 745
158 FORMAT/// 25X,24HPAPPROX CD USING SOLVED ROUND/25X,4HCIRCULATIONS AN 746
1D WASHES AT SFG MIDPOINTS = ,E13.5/) 747
END
SUBROUTINE GAMCAL (IL, SN, AOPT, HGAM)

CALCULATE BOUND CIRCULATION AT LEFT END PT OF SEGMENT ILHGAM

AOPT LAST COLUMN CONTAINS ARRAY OF OPTIMIZED SHEET STRENGTHS

COMMON /FEN/ NSPT(10), NLIN
DIMENSION SN(1), AOPT(5,2,1)
HGAM = 0.
GO TO 10

K = NLIN - K + 1

KCHK = NSPT(KK)

IF (I, KCHK) GO TO 4

IF (I, K, KCHK) GO TO 2

1 CONTINUE

2 CONTINUE

KST = KCHK
HGAM = AOPT(KST, IL) * SNN(KST) * AOPT(1, IL) * SNN(1 - 1)

KCHK = KCHK + 1

IF (I, KCHK) GO TO 5

I = I - 1

IP = KCHK + 1

GO TO 5

3 CONTINUE

GO TO 5

4 HGAM = 0.

5 CONTINUE

RETURN

END
SUBROUTINE WCAL

1. CALCULATE MATRIX COEFFICIENT FOR DRAG OPTIMIZATION USING MUNK CRIT
2. FINDS COEF OF ITH DOWNWASH DUE TO JTH SHEET STRENGTH
3. IF, FINDS COEF MULTIPLYING STRENGTH J IN EQUATION I

DIMENSION YHN(1), ZHN(1), PPP(1)
COMMON /SEG/ SNN(5)
COMMON /FEN/ NSPT(10), NLINIT
INTERFER P
PI = 4. * ATAN (1.)
AAAA = 0.
ICNTRL = 6
P = 1
K = J

1. CONTINUE
2. CALL CCAL (PK, YHN, ZHN, PPP, SNN(K), AA, BH, DD, FF, GG, EE, AJ, AK, RR, TT, UU)
3. CONTINUE
4. CALL CCAL (AA, BH, FF, GG, SNN(K), AH, CJ, D, F, G, HJ, BK, RL, RM, BN, BR, BP, 1)
5. CONTINUE
6. IF (PPI = 0.) GO TO 2
7. EXP = 2. * (ATAN2 (C, AHS (RR)) - ATAN2 (U, ABS (HR))) / (AH5 (HR))
8. GO TO 3
9. CONTINUE
10. EXP = 2. / U - 2. / C
11. CONTINUE
12. RLOG = ALOG (F/G)
14. A2PK = (A2PK, SNN(K) + BO*RLG) / (2. * PI*SNN(K))
15. CALL CCAL (D1, F, A, AK, SNN(K), A, C, D, F, G, HJ, BK, RL, RM, BN, BR, BP, 2)
16. IF (UHI == 0.) GO TO 4
17. EXP = 2. * (ATAN2 (C, AHS (UU)) - ATAN2 (U, ABS (UU))) / (ABS (UU))
18. GO TO 5
19. CONTINUE
20. EXP == 2. / U - 2. / C
21. CONTINUE
22. RLOG = ALOG (F/G)
23. A2PK = - (A2PK, SNN(K) + BO*RLG) / (2. * PI)
24. A4PK = - (A4PK, SNN(K) + BO*RLG) / (2. * PI*SNN(K))
25. IF (ICNTRL == 7) GO TO 7
26. A1PK = - (A1PK, A2PK) * 0.50 - (A3PK + A4PK) * 0.50
SURROUNTE WCAL

ICNTRL = 2
DO 6 L=1,NLLINE
JCHK = NSPT(L)
IF (K,FR,JCHK) GO TO H
6 CONTINUE
K = J+1
GO TO J
CONTINUE
7 CONTINUE
PPAA = PAA + 0.5*(A1PK+A2PK+A3PK+A4PK)
A CONTINUE
RETURN
END

C 43
C 44
C 45
C 46
C 47
C 48
C 49
C 50
C 51
C 52
C 53
C 54
SURROUTINE CCAL

SUBROUTINE CCAL (I,J,YHH,ZZH,PPP,S,AA,HH,DD,FF,GG,EE,JJ,KK,RR,TT,WW)

C CALCULATES GEOMETRICAL CONSTANTS NEEDED IN EVALUATION OF INTEGRALS
C FOR VARYING I AND J VALUES

REAL JJ,KK
DIMENSION YHH(I), ZZH(J), PPP(I)

DYIJ = YHH(I) - YHH(J)
DZIJ = ZZH(I) - ZZH(J)
COI = COS(PPP(I))
SIJ = SIN(PPP(I))
COJ = COS(PPP(J))
SIJ = SIN(PPP(J))
AA = DYIJ*COI*DZIJ*SIJ
MH = -COS(PPP(J) - PPP(I))
FF = -2*(DYIJ*COJ*DZIJ*SIJ)

GG = DYIJ*DYIJ + DZIJ*DZIJ

DD = DYIJ*COI*DZIJ*SIJ
FF = COS(PPP(J) + PPP(I))

JJ = 2*(DYIJ*COJ - DZIJ*SIJ)
KK = DYIJ*DYIJ + DZIJ*DZIJ
PR = 2*(DYIJ*SIJ - DZIJ*COJ)
TT = 2*SIN(PPP(J) - PPP(I))
IU = 2*(DYIJ*SIJ + DZIJ*COJ)

WW = 2*SIN(PPP(J) + PPP(I))

RETURN

END
SUBROUTINE CONCAL

REAL J,K,L,M,N
A = AA-0.5*RR*FF
B = 1.0-HH*RR
C = FF+2.0*S
D = FF-2.0*S
F = S*S+FF+GG
G = S*S-FF+GG
J = 2.0*(AA+FF)
L = 0.5*(AA*FF-AA*FF-2.0*HH*GG)
M = 0.5*(-FF-6.0*AA*HH+4.0*FF*HH*BB)
N = 2.0*(HH*HH-1.0)*RR
O = 0.5*(AA-FF*HH)
P = 0.5*(1.0-2.0*HH*HH)
RETURN
END
SUBROUTINE SNTAN

SUBROUTINE SNTAN (S, C, RR, TT, RTAN, RS1AN, RS2TAN, RS3TAN, RS4TAN)

EVALUATES INTEGRALS OF THE FORM S**N*ATAN((C+2*RR*S)/(PR*S*TT))
ALL DIVIDED BY (RR*S*TT)
WITH RESPECT TO S BETWEEN LIMITS OF -S AND S FOR N=0,1,2,3.

ATAN PART OF INTEGRAND APPROXIMATED AS A QUADRATIC IN S WHICH IS
FORCED THROUGH ATAN VALUES AT -S, 0, AND S.

A,C ARE CALCULATED IN SUBROUTINE CONCAL
RR,TT ARE CALCULATED IN SUBROUTINE CCAL

RESULTS ARE RTAN,RS1AN,RS2TAN,RS3TAN
APPROXIMATE INTEGRAL EVALUATED USING MACSYMA PROGRAM OF MIT PROJ.
EVALUATION OF INTEGRALS FOR TT=0, BEGIN AT AT LABLE 10
SINGULAR INTEGRALS EVALUATED AT APPROXIMATE ENDPOINTS, -S/2 AWAY
MIDRANGE SINGULARITIES EXCLUDED, ATAN PART OF INT APPROX-
IMATED AS 2 QUADRATICS

DIMENSION AA(3,3), AA(3), IPIVOT(3)
COMMON /TELL/, TOL, TOL2

RRR = RP
SSS = S
CCC = C
RTAN = 0.
RS2TAN = 0.
RS3TAN = 0.
RS4TAN = 0.

IF (APS(TT,LT,IE-05) AND ABS(RR),LT,IE-05) GO TO 7
IF (TT,EQ,0.0,AND,RP,EQ,0.0) GO TO 7
IF (TT,GT,0.0) GO TO 4

FIRST, CHECK FOR MIDRANGE SINGULARITIES, EXCLUDING ANY FOUND

SZERO = -RR/TT
IF (APS(AHS(SZERO)-S),LT,IE-03,AND,ABS(SZERO),LE,5) GO TO 3
IF (SZERO,LT,0.0,AND,SZERO,GT,-5) GO TO 16
IF (SZERO,GT,0.0,AND,SZERO,LT,5) GO TO 16
CONTINUE
SUBROUTINE SNTAN

C = ATAN2((CCC, ABS(RR)))
CL = S*(ATAN2((CCC+2*RR*RR, ABS(RR+TT*TT)))+ATAN2((CCC-2, BB*BB, ABS)))
C1 = C1/(S*S)
C2 = (ATAN2((CCC+2, RR*RR, ABS(RR+TT*TT)))/S-C-S-C)
C INTEGRAND NOW IS C1*S*S+C2*S*C*S*S*T/(RR-TT*TT)
CLOGR = ALOG(ABS((RR+TT*TT)/(RR-TT*TT))
CON0 = (C*TT*TT-C2*RR*TT+CC1*RR)/TT*TT)
CON1 = (C2*TT-C1*RR)/TT*TT)
RTAN = COND*CLOGR+S*CON1)
CON2 = (C*TT*TT-C2*RR*TT+CC1*RR*RT)/(TT*TT)
CON3 = (C1*TT*TT-C2*RR*TT+CC1*RR*RT)/(TT*TT)
RSTAN = 2*S*CON2-CON3*CLOGR+2*C1*TT*TT*TT+3)/3*TT*TT)
CON4 = (C1*TT*TT-C2*RR*TT+CC1*RR*RT)/(TT*TT)
CON5 = 4*C2*TT*TT*3-1*2*C1*RR*TT*2
CON5 = CON5/(17*TT*TT)
CON6 = (-C*PR*TT*TT-C2*RR*RR*TT-C1*RR*RT)/(TT*TT)
RSTAN = CON4*CLOGR+S*CON6)
CON9 = 20*(C*TT*TT-C2*RR*TT+CC1*RR*RT)/(60*TT*TT)
CON9 = (C1*TT*TT-C2*RR*TT+CC1*RR*RT)/(TT*TT)
CON10 = C1*TT*TT-C2*RR*TT+CC1*RR*RT)
CON10 = CON10/(17*TT*TT)
PS3TAN = 2*S*CON9*S*3-2*S*CON9*CON10+CLOGR+24*C1*TT*TT*TT*5/(60*TT*TT)
PS3TAN = 2*S*CON9*S*3-2*S*CON9*CON10+CLOGR+24*C1*TT*TT*TT*5/(60*TT*TT)
CON0 = (C1*TT*TT-C2*RR*TT+CC1*RR*RT+6)/(TT*TT)
CON8 = (C2*TT-C1*RR)/(5*TT*TT)
CONC = (-C*PR*TT*TT-C2*RR*TT+CC1*RR*RT)/(3*TT*TT)
COND = CONC*RR*RT/(TT*TT)
R54TAN = CONA*CLOGR+2*S*5+CONC*2*S*5+COND*2*S)
IF (RR+TT, 0) GO TO 2
R54TAN = -R54TAN
R54TAN = -R54TAN
R54TAN = -R54TAN
R54TAN = -R54TAN
2 CONTINUE
3 CONTINUE
S = SAWAY
SUBROUTINE SNTAN

GO TO 1

C FOR CASE OF RR NOT ZERO, TT=0.0
C
4 CONTINUE

RR = ABS(RR)

ALNUM = (2*RR)**2*S**2+4*C*B*S+RR**2*C**2

ALDEN = (2*RR)**2+S**2-4*C*B*S+RR**2+C**2

IF (ALNUM, EQ, 0.0, OR, ALDEN, EQ, 0.0) GO TO 5

GO TO 6

5 CONTINUE

S = S-10L

ALNUM = (2*RR)**2*S**2+4*C*B*S+RR**2+C**2

ALDEN = (2*RR)**2+S**2-4*C*B*S+RR**2+C**2

6 RATLN = ALOG(ALNUM/ALDEN)

TNDIF = ATAN2((C+2.*RR*S),RR)-ATAN2((C-2.*BB*S),RR)

TNSUM = ATAN2((C+2.*BB*S),RR)+ATAN2((C-2.*BB*S),RR)

PTAN = -((C*R/RH)*RATLN+0.5+C*TNDIF/DTL)

PTAN = PTAN/RH+S/TT

RSTAN = (S**2/(S**2+S**2+4*C*B**2+S**2+RR**2+C**2))/2+C*TNDIF/DTL

GO TO 13

F FOR CASE OF RR=TT=0.0, IF I=J

TOP = C+2.*HH*S

BOT = C-2.*RR*S

IF (C*RQ, 0.0, AND, BH, EQ, 0.0) GO TO 13

SBAN = -C/(2.*HH)

SBADAR = ABS(SBAN)

IF (SBADAR, LT, 5) GO TO 8

5

GO TO 1

6

GO TO 13

13
PROGRAM SUANUT

120 CONTINUE
WRITE (*,25)
SUL = SAIN-TOL
SLI = SALT-TOL
CLOGR1 = ALOG(TOP/(C+2.*RH*TLL))
CLOGR2 = ALOG((C+2.*RH*TLL)/BOT)
CLOGR = CLOGR1+CLOGR2
RTAN = -(S/PF)*CLOGR
RSTAN = -(S/PP*(S-LL)*S)=CLOGR*(C/(S-RH**2))
125/PH*

130 CONTINUE
IF (TOP.LT.E-9) GO TO 12
IF (TOP.LE.n.n) GO TO 11
IF (TOP.LT.0.0) GO TO 14

135 CLOGR = ALG(Q/A,.PUT)
RTPN = -(S/PP)*(CLOGR-S/PP)
RSTAN = -(C/C*C)**S+3/(3.*BB)**2)*CLOGR-S/PP
140 CONTINUE
IF (TOP.GT.E-9) GO TO 10
IF (TOP.LE.0.0) GO TO 14
IF (TOP.LT.0.0) GO TO 14

145 CLOGR = ALOG(TOP/BOT)
PTAN = -(1/HR)*CLOGR
RTAN = RTAN/2
RSTAN = -(S/PP)*CLOGR-S/PP
150 CONTINUE
IF (TOP.LT.E-9) GO TO 10
IF (TOP.LE.0.0) GO TO 14
IF (TOP.LT.0.0) GO TO 14

155 CLOGR = ALOG(TOP/BOT)
PTAN = -(1/HR)*CLOGR
RTAN = RTAN/2
GO TO 13

160 CONTINUE
STOP = S-TOL
STOP = C+2.*BB*SAWAY
CLOGR = ALOG(TOP/BOT)
PTAN = -(1/HR)*CLOGR
RTAN = RTAN/2
STOP = S-TOL
STOP = C+2.*BB*SAWAY
CLOGR = ALOG(TOP/BOT)
PTAN = -(1/HR)*CLOGR
RTAN = RTAN/2
GO TO 13
SURROUTINE SNTAN

RSSTAN = (1.25*C/RR**2) * CLOGR - S/HH
RS2TAN = -((C*2/(HH)**3)) * CLOGR + C*S/2.*BB**2
RS3TAN = (C*3/(1.6*HH**4)) * CLOGR - S**3/3.*HH - S*C**2/(4.*BB**3)
RS4TAN = -(C*4/(32.*HH**5)) * CLOGR + 2.*C*S**3/(17.*BB**2) + 2.*S*C**3/116*HH**6

GO TO 13

13 CONTINUE

WRITE STATEMENTS GO HERE IF NEEDED
GO TO 15

WRITE (6,26)
CONTINUE
GO TO 74

WRITE (6,27)
SUL = SZERO-TOL
SLL = SZERO+TOL
SMID1 = S=0.5*ABS(S-SLL)
SMID2 = S=0.5*ABS(S-SUL)
ANG1 = ATAN2((C+2.*RR*S),ABS(RR+TT*S1))
ANG2 = ATAN2((C+2.*RR*S),ABS(RR+TT*SMID1))
ANG3P = ATAN2((C+2.*RR*S),ABS(RR+TT*SLL))
ANG3 = ATAN2((C+2.*RR*S),ABS(RR+TT*SMID2))
ANG5 = ATAN2((C+2.*RR*S),ABS(RR+TT*SUL))

IF (I.EQ.2) GO TO 17

AA1(1,1) = SUL*SUL
AA1(1,2) = SUL
AA1(1,3) = AA1(2,3)=AA1(3,3)=1.
AA1(2,1) = SMID2*SMID2
AA1(2,2) = SMID2
AA1(2,3) = SMID2
AA1(3,1) = S8S
AA1(3,2) = -S
AA1(3,3) = ANG5
AA(2) = ANG4
AA(3) = ANG3
AA(1) = PPGS

CLOGR = ALOG(ABS((RR+TT*SUL)/(RR+TT*S)))
SUAROUTINE SUAROUTINE

SUSE = SMID2
DFLS = SUL * S
DELS2 = SUL**2 - S**2
DELS3 = SUL**3 + S**3
DELS4 = SUL**4 - S**4
DELS5 = SUL**5 + S**5
DELS6 = SUL**6 - S**6
GO TO 16

17 CONTINUE

220

AA1(1,1) = S**7
AA1(1,2) = S
AA1(1,3) = AA1(2,3) = AA1(3,3) = 1.
AA1(2,1) = SMID1 * SMID1
AA1(2,2) = SMID1
AA1(3,1) = SLL * SLL
AA1(3,2) = SLL
AA1(3,3) = ANG1
AA(1) = ANG1
AA(2) = ANG2
AA(3) = ANG3

230

CLOGP = ALOGP(ABS((RR*TT*S)/(RR*TT*SLL)))
SUSE = SMID1
DFLS = S - SLL
DELS2 = S*S - SLL
DELS3 = S**3 - SLL**3
DELS4 = S**4 - SLL**4
DELS5 = S**5 - SLL**5
DELS6 = S**6 - SLL**6
GO TO 16

10 CONTINUE

CALL SIMF0 (AA1,3,AA1,DET,IPIVOT,3,ISCALE)

C1 = AA(1)
C2 = AA(2)
C = AA(3)
CON0 = (C**TT*T1 - C2*RH*T1 + C1*RR*RH) / (TT**3)
CON1 = (C2*TT - C1*WH) / (TT*TT)
CON11 = 0.5*C1 / TT
CON2 = (C*TT*T1 - C2*RH*T1 + C1*WH*KR) / (TT**3)
CON3 = (C*RP*TT - C2*RH*TT + C1*RR*TT) / (TT**3)
CON21 = C2 / (2*TT*TT)
CON4 = (C*RP*TT - C2*RH*TT + C1*RR*TT) / (TT**3)
CON5 = 4.0*(C2*TT**3 - 4.0*C1*RP*TT**2)
CON5 = DBS / (2*TT**3)
CON6 = (-C*RP*TT + C2*RH*TT - C1*RR*TT) / (TT**3)

F 211
F 212
F 213
F 214
F 215
F 216
F 217
F 218
F 219
F 220
F 221
F 222
F 223
F 224
F 225
F 226
F 227
F 228
F 229
F 230
F 231
F 232
F 233
F 234
F 235
F 236
F 237
F 238
F 239
F 240
F 241
F 242
F 243
F 244
F 245
F 246
F 247
F 248
F 249
F 250
F 251
F 252
CON31 = (C*TT*3 - C2*RR*TT**2 + C1*RR**2*TT) / (2*TT**4) F 253
CON32 = C1 / (4*TT) F 254
CON41 = (C2*TT - C1*RR) / (4*TT**2) F 255
CON42 = (-C2*RR**2 + C2*RR**2*TT - C1*RR**3 + C1*RR**2*TT) / (2*TT**4) F 256
CON10 = C10 / (TT**4) F 257
CON10 = CON10 / (TT**4) F 258
CON1 = (C2*TT - C1*RR) / (4*TT**2) F 259
CON2 = (C2*TT - C1*RR) / (4*TT**2) F 260
CON3 = (C2*TT - C1*RR) / (4*TT**2) F 261
CON4 = (C2*TT - C1*RR) / (4*TT**2) F 262
CON5 = (C2*TT - C1*RR) / (4*TT**2) F 263
CON6 = CON6 / (TT**2) F 264
CON7 = CON7 / (TT**2) F 265
CON8 = CON8 / (TT**2) F 266
CON9 = CON9 / (TT**2) F 267
CON10 = CON10 / (TT**2) F 268
CON11 = CON11 / (TT**2) F 269
CON12 = CON12 / (TT**2) F 270
CON13 = CON13 / (TT**2) F 271
CON14 = CON14 / (TT**2) F 272
CON15 = CON15 / (TT**2) F 273
CON16 = CON16 / (TT**2) F 274
CON17 = CON17 / (TT**2) F 275
CON18 = CON18 / (TT**2) F 276
CON19 = CON19 / (TT**2) F 277
CON20 = CON20 / (TT**2) F 278
CON21 = CON21 / (TT**2) F 279
CON22 = CON22 / (TT**2) F 280
CON23 = CON23 / (TT**2) F 281
CON24 = CON24 / (TT**2) F 282
CON25 = CON25 / (TT**2) F 283
CON26 = CON26 / (TT**2) F 284
CON27 = CON27 / (TT**2) F 285
CON28 = CON28 / (TT**2) F 286
CON29 = CON29 / (TT**2) F 287
CON30 = CON30 / (TT**2) F 288
CON31 = CON31 / (TT**2) F 289
CON32 = CON32 / (TT**2) F 290
CON33 = CON33 / (TT**2) F 291
CON34 = CON34 / (TT**2) F 292
CON35 = CON35 / (TT**2) F 293
CON36 = CON36 / (TT**2) F 294
SUBROUTINE SNTAN

295 21 CONTINUE
22 CONTINUE
23 CONTINUE
   GO TO 13
24 CONTINUE
300   RR = PPR
       S = SS
       C = CCC
       RETURN

305 25 FORMAT(30X,11H200 ENTERED)
26 FORMAT(30X,43HONE OF THE ENDPONTS HAS A NEGATIVE LOG ARG)
27 FORMAT(30X,10H200 ENTERED)
    END
SUBROUTINE LOGS (S+D,RELN,RESLN,RES2LN,RES3LN)

CALCULATES INTEGRALS OF FORM S**N*ALOG(S*S+E*S+D) WITH RESPECT TO S OVER LIMITS OF -S TO S FOR N=0,1,2,3.

S = PANEL SEGMENT HALFWIDTH
E,D ARE CALCULATED IN SUBROUTINE CONCAL
INTEGRAL RESULTS ARE RELN,RESLN,RES2LN,RES3LN
EVALUATION OF INTEGRALS PERFORMED USING MACSYMA ALGEBRAIC MANIPULATION PROGRAM OF MIT PROJECT MAC
IF I=J INTEGRAL EVALUATED AT APPROXIMATE ENDPOINTS, S=SAWAY

REAL LATB+LANB+L1+L2+L3+L4
COMMON /TELL/ TOL+TOL2
RELN = 0.
RESLN = 0.
RES2LN = 0.
RES3LN = 0.
SS = S
A = S+S+E*S+D
R = S+S-E*S+D
A = ABS(A)
R = ABS(R)
AA = ABS(A)
BB = ABS(B)
IF (AA,LE,0.00000000001) GO TO 6
IF (BB,LE,0.00000000001) GO TO 6

DISC = E*F-10
DISQ = SORT(ABS(DISC))
DIS = E*F-2.8
DIS3 = E**3-3,7*10
DIS4 = (E*F-4.2)*E*F-3.5
DIS5 = E*F-5.6*E*F-4.3*E*F**2
LATH = ALOG(A*B)
LADH = ALOG(A/B)

IF (AA,LE,0.00000000001) S = SAWAY
IF (BB,LE,0.00000000001) S = SAWAY
PE = S*LATB*0.5*F*LADH
SUBROUTINE LOGS

ESP = F+2*S
ESM = F-2*S
IF (DISC) 43,3+2
2 CONTINUE
L1 = F-DIS5+2*S
L2 = E+DIS5+2*S
L3 = F-DIS5-2*S
L4 = F+DIS5-2*S
50 DIFFLN = ALOG(L)*L/L3/L4)
RFNL = 0.5*LADD*S**2+(0.25*E*DISC/DISQ)*DIFFLN
RESLN = RFNL+0.25*LADD*E*S
55 RES2LN = (S**3/3)*LATR-DIS4/(6*DISQ)*DIFFLN-LADD*(DIS3/6)-4*S**3/9
1=6*DIS5/S
RES3LN = 0.25*S**4*LADD+DIS5/(8*DISQ)*DIFFLN-LADD*(DIS4/8)+E*S**3
1/6+0.5*S*DIS3
GO TO 2
3 CLOGRT = ALOG(ESP/ESM)
RELN = S*LALT-4*S*F*CLOGRT
RESLN = 0.5*S**2*LADD-0.5*DIS*CLOGRT+E*S
RES2LN = (S**3*LALT+DIS3*CLOGRT*(DIS4)((1/ESP)-(1/ESM))-2*S*DIS)/
1=4*(S**3)/3
60 RES3LN = 0.25*S**4*(LADD-DIS4+0.25*CLOGRT=25*DIS5*(1/ESP)-1/ESM)*E
1/6+0.5*S**DIS3
GO TO 5
4 TNRAT = ATAN2(ESP,DISQ)-ATAN2(ESM,DISQ)
PENL = PE-4*S-(DISC/DISQ)*TNRAT
RESLN = 0.5*(S**2-0.5*DIS)*LADD+0.5*E*DISC/DISQ*TNRAT+E*S
RES2LN = S**3/3*LALT*(DIS3/6)*LADD-(DIS4/3*DISQ)*TNRAT-4*S**3/9-
12*S*DIS3/3
RES3LN = (0.25*S**4-DIS4/8)*LADD+0.25*DIS5/DISQ*TNRAT+E*S**3/6+S*
DIS3/2
75 5 CONTINUE
GO TO 7
6 CONTINUE
SAWAY = S-TOL
A = SAWAY*SAWAY+E*SAWAY+D
80 T = SAWAY*SAWAY-F*SAWAY+D
A = ABS(A)
R = ABS(R)
GO TO 1
7 CONTINUE

G 43
G 44
G 45
G 46
G 47
G 48
G 49
G 50
G 51
G 52
G 53
G 54
G 55
G 56
G 57
G 58
G 59
G 60
G 61
G 62
G 63
G 64
G 65
G 66
G 67
G 68
G 69
G 70
G 71
G 72
G 73
G 74
G 75
G 76
G 77
G 78
G 79
G 80
G 81
G 82
G 83
G 84
SUBROUTINE LOGS

85 S = SS
RETURN
END

G 85
G 86
G 87
SUBROUTINE DRACAL (I,J,YHH,ZHH,PPP,AINT)

SUBROUTINE DRACAL

TREFFT PLANT DRAG ANALYSIS ASSUMES PIECEWISE LINEARLY VARYING SHEAR VORTICITY SHEET STRENGTH

CALCULATE INTEGRALS A THROUGH F FOR DRAG COEF CALCULATION

CALLS SUBROUTINES LOGS,SNTAN,CCAL,CONCAL

DIMENSION AINT(6)
DIMENSION YHH(1), ZHH(1), PPP(1)
COMMON /SEG/ SNN(51)
PI = 4.*ATPNFL.
S = SIN(J)
CALL CCAL (I,J,YHH,LHH,PPP,S,AA,RR,DD,FF,GG,EF,AF,AK,RR,TT,WW,WW)
CALL CONCAL (AA,HH,FF,GG,SS,AB,CC,DD,GG,CC,CL,CM,CM,CO,CP,1)
S = SNN(J)
CALL LOGS (S,CJ,F,HELN,RESLN,RES2LN,RES3LN)
CALL SNTAN (S,CJ,TT,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)

AAAAA = A*RTAN+B*RSTAN+BE*RELN/4
BBRHRH = 2.*(TT*RTAN+CM*RSTAN+CN*RS2TAN)+CO*RELN+CP*RESLN
CCCCCC = A*RSTAN+R*RS2TAN+RR*RESLN/4
DDDDDD = 2.*(TT*RSTAN+CM*RS2TAN+CN*RS3TAN)+CO*RESLN+CP*RES2LN
EEEFFE = 2.*(TT*RS2TAN+CM*RS3TAN+CN*RS4TAN)+CO*RES2LN+CP*RES3LN
CALL LOGS (S,CJ,FF,RELN,RESLN,RES2LN,RES3LN)
CALL SNTAN (S,C,J,T,TT,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)

AAAAAA = A*RTAN+B*RSTAN+BB*RELN/4
BBBRRH = 2.*(TT*RTAN+CM*RSTAN+CN*RS2TAN)+CO*RELN+CP*RESLN
CCCCCC = A*RSTAN+B*RS2TAN+BR*RELN/4
DDDDDD = 2.*(TT*RSTAN+CM*RS2TAN+CN*RS3TAN)+CO*RESLN+CP*RES2LN
EEEFFF = EEEFFF-2.*(TT*RS2TAN+CM*RS3TAN+CN*RS4TAN)-CO*RES2LN+CP*RE
153LN
S = SNN(J)
CALL CONCAL (DD,EF,AF,AK,SS,A,B,CC,DD,GG,CC,CL,CM,CM,CO,CP,2)
S = SNN(J)
CALL LOGS (S,CJ,F,RELN,RESLN,RES2LN,RES3LN)
CALL SNTAN (S,CJ,TT,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)
SUBROUTINE DRACAL

AAAAAA = AAAAAA * ARRTAN * RTSTAN * E * RELN / 4
BBBBBB = BBBBBB * (CL * RTAN + CM * RTAN + CN * RTSTAN) - CO * RELN - CP * RESLN

CCCCCC = CCCCCC * ARRTAN * RS2TAN * E * RESLN / 4

DDDDDD = DDDDDD * (CL * RTAN + CM * RTSTAN + CN * RTSTAN) - CO * RESLN - CP * RES2

FFFEFF = EEEEEE * ARSTAN * RTAN * EE * RES2LN / 4

FFFFFFFF = FFFFFF * (CL * RS2TAN + CM * RS3TAN + CN * RS4TAN) - CO * RES2LN - CP * RE

1S3LN
CALL LOGS (S, CN, RTAN, RESLN, RES2LN, RES3LN)

CRI TAN (S, PI, EE, (1/2) * RTAN, RS2TAN, RTSTAN, RS4TAN)

AAAAAA = AAAAAA / PI

HRP = BBBBBB / (2 * PI * SK) + (2 * S / PI) * (BB - EE)

CCCCCC = CCCCCC / PI

DDDDDD = DDDDDD / (2 * PI * SK)

FFFEFF = EEEEEE / (2 * PI * SK)

FFFFFFFF = FFFFFF / (4 * PI * S * SK + (BB - EE) * S * S / (3 * PI))

AINT (1) = AAAAAA

AINT (2) = BBBBBB

AINT (3) = CCCCCC

AINT (4) = DDDDDD

AINT (5) = FFFFFF

AINT (6) = FFFFFF

RETURN

END
SUBROUTINE SINEQ (A, N, M, DETERM, IPIVOT, NMAX, ISCALE)

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

*** DOCUMENT DATE 08-01-68 SUBROUTINE REVISED 08-01-68 **********

DIMENSION IPIVOT(N), A(NMAX,N), B(NMAX,M)
EQUIVALENCE (IROW, JROW), (ICOLUMN, JCOLUMN), (AMAX, T, SWAP)

INITIALIZATION

ISCALE = 0
R1 = 1.0
R2 = 1.0/100
DETERM = 1.0

DO 2 J = 1, N
2 IPIVOT(J) = 0

DO 3 R I = 1, N
3 IF (IPIVOT(I) + 1) 3, 7, 3

DO 6 K = 1, N
6 IF (IPIVOT(K) + 1) 4, 6, 39

IF (ABS(A(AMAX)) - ABS(A(I,K))) 5, 6, 6
5 IF (ABS(A(AMAX)) - ABS(A(I,K))) 5, 6, 6

ICOLUMN = K
AMAX = A(I,K)
CONTINUE

CONTINUE
IF (AMAX) 9, 8, 9

IF (DETERM) 9, 8, 9
ISCALE = 0
GO TO 39

IPIVOT(ICOLUMN) = IPIVOT(ICOLUMN) + 1

INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL

IF (IROW = ICOLUMN) 10, 14, 10

DETERM = -DETERM
DO 11 L = 1, N
SWAP = A(IROW, L)
A(IROW,L) = A(ICOLUM,L)
11 A(ICOLUM,L) = SWAP
IF (M) 14,14,12
12 DO 13 L=1,M
SWAP = H(IROW,L)
13 A(IROW,L) = A(ICOLUM,L)
14 PIVOT = A(ICOLUM,ICOLUM)
IF (PIVOT) 15,8,15
C SCALE THE DETERMINANT
C
15 PIVOT = PIVOT
IF (ARSDETERM-R1) 18,16,16
16 DETERM = DETERM/R1
ISCALE = ISCALE+1
IF (ARSDETERM-R1) 21,17,17
17 DETERM = DETERM/R1
ISCALE = ISCALE+1
GO TO 21
18 IF (ARSDETERM-R2) 19,19,21
19 DETERM = DETERM/R1
ISCALE = ISCALE-1
IF (ARSDETERM-R2) 20,20,21
20 DETERM = DETERM/R1
ISCALE = ISCALE-1
GO TO 21
21 IF (ARSPIVOT-R1) 24,22,22
22 PIVOT = PIVOT/R1
ISCALE = ISCALE+1
IF (ARSPIVOT-R1) 27,23,23
23 PIVOT = PIVOT/R1
ISCALE = ISCALE+1
GO TO 27
24 IF (ARSPIVOT-R2) 25,25,27
25 PIVOT = PIVOT/R1
ISCALE = ISCALE-1
IF (ARSPIVOT-R2) 26,26,27
26 PIVOT = PIVOT/R1
ISCALE = ISCALE-1
GO TO 27
27 DETERM = DETERM*PIVOT
C DIVIDE PIVOT ROW BY PIVOT ELEMENT
SUBROUTINE SIHFQ

C
DO 29 L=1,N
IF (IPIVOT(L)-1) 29,29,39
28 A(ICOLM,L) = A(ICOLM,L)/PIVOT
29 CONTINUE

C
DO 31 L=1,M
R(ICOLM,L) = R(ICOLM,L)/PIVOT

C
REDUCE NON-PIVOT ROWS
C
DO 38 L=1,N
IF (L-ICOLM) 33,38,33
33 T = A(L,ICOLM)
DO 35 L=1,N
IF (IPIVOT(L)-1) 34,35,39
34 A(L,L) = A(L,L)-A(ICOLM,L)*T
35 CONTINUE
IF (L) 39,78,78
36 DO 37 L=1,M
37 B(L,L) = B(L,L)-R(ICOLM,L)*T
38 CONTINUE
39 RETURN
END
REFERENCES


Figure 1. Trefftz plane geometry used in the present method.
Figure 2. Induced drag efficiency for nonplanar wing with vertical fences of variable size.
Figure 3. Induced drag efficiency for a series of vee wings.
Figure 4. Induced drag efficiency for a series of diamond wings.
Figure 5. Comparison of bound circulation distributions for vee wing and diamond wing, $\phi = 30^\circ$, using present theory.
Figure 6. Induced drag efficiency for diamond wing fitted with end plates; $N = 30$ cosine segment spacing, and $\phi = 89.5^\circ$ on end plates for all results.
Figure 7. Induced drag efficiency for diamond wing fitted with winglets. Winglets perpendicular to wings, N = 30, and cosine segment spacing for all results.
**Title and Subtitle**
Numerical Optimization Techniques for Bound Circulation Distribution for Minimum Induced Drag of Nonplanar Wings: Computer Program Documentation

**Performing Organization Name and Address**
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508

**Abstract**
A two-dimensional advanced panel far-field potential flow model of the undistorted, interacting wakes of multiple lifting surfaces has been developed which allows the determination of the spanwise bound circulation distribution required for minimum induced drag. This model has been implemented in a FORTRAN computer program, the use of which is documented in this report.

The nonplanar wakes are broken up into variable sized, flat panels, as chosen by the user. The wake vortex sheet strength is assumed to vary linearly over each of these panels, resulting in a quadratic variation of bound circulation. Panels are infinite in the streamwise direction. The theory is briefly summarized herein; sample results are given for multiple, nonplanar, lifting surfaces, and the use of the computer program is detailed in the appendixes.

**Key Words**
- Subsonic Aerodynamics
- Induced Drag Minimization
- Nonplanar Configurations
- FORTRAN Computer Program
- Users Manual

**Distribution Statement**
FEDD Distribution

**Available:** NASA's Industrial Applications Centers

**Security Classification**
Unclassified