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
SUMMARY

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)]
- \(A_1\), \(A_2\), \(A_3\), \(A_4\): 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
- **ΔC\(_p\)**: 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\(_i,\tilde{G}_i,\hat{G}_i\)**: 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)
- \(I_1\), \(I_2\), \(I_3\), \(I_4\), \(I_5\), \(I_6\): influence coefficient integrals appearing in drag coefficient equation [eq. (6)]
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>induced drag efficiency factor, defined as ratio of planar wing induced drag to that of nonplanar configuration</td>
</tr>
<tr>
<td>$l$</td>
<td>vertical fence height (fig. 2)</td>
</tr>
<tr>
<td>$n$</td>
<td>integer appearing in integrals in Appendix A</td>
</tr>
<tr>
<td>$N$</td>
<td>number of wake panels on one-half of total configuration</td>
</tr>
<tr>
<td>$R$</td>
<td>constant appearing in integrals in Appendix A</td>
</tr>
<tr>
<td>$R_{ij}$</td>
<td>projection of distance $h_{ij}$ onto the plane of influenced panel $i$ (fig. 1)</td>
</tr>
<tr>
<td>$R'_{ij}$</td>
<td>projection of distance $h'_{ij}$ onto the plane of influenced panel $i$ (fig. 1)</td>
</tr>
<tr>
<td>$S_{ref}$</td>
<td>reference wing area</td>
</tr>
<tr>
<td>$s$</td>
<td>local wake panel coordinate</td>
</tr>
<tr>
<td>$s$</td>
<td>wake panel semiwidth</td>
</tr>
<tr>
<td>$T$</td>
<td>constant appearing in integrals in Appendix A</td>
</tr>
<tr>
<td>$U$</td>
<td>free-stream velocity</td>
</tr>
<tr>
<td>$W_n$</td>
<td>normal wash velocity</td>
</tr>
<tr>
<td>$W_{n,j}$</td>
<td>normal wash velocity induced at point $s = s_j$ on panel $i$ due to panel $j$ and its image [eq. (2)]</td>
</tr>
<tr>
<td>$w_o$</td>
<td>constant appearing in Munk's criterion normal wash velocity expressions [eqs. (7) and (8)]</td>
</tr>
<tr>
<td>$X$</td>
<td>streamwise coordinate</td>
</tr>
<tr>
<td>$Y$</td>
<td>spanwise coordinate</td>
</tr>
<tr>
<td>$Z$</td>
<td>vertical coordinate, positive down</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>wake trailing vortex sheet strength</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>bound circulation</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>average bound circulation</td>
</tr>
<tr>
<td>$\Gamma_0$</td>
<td>bound circulation at outboard edge of wake panel [eq. (4)]</td>
</tr>
<tr>
<td>$\eta$</td>
<td>nondimensional spanwise coordinate</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Lagrange multiplier in equations (10) and (12)</td>
</tr>
<tr>
<td>$\sigma_{ij}$</td>
<td>angle between $y$-axis and orientation of $h_{ij}$ (fig. 1)</td>
</tr>
<tr>
<td>$\phi$</td>
<td>dihedral angle</td>
</tr>
</tbody>
</table>
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

\[
\gamma_j(\delta_j) = \frac{\gamma_{j+1} + \gamma_j}{2} + \frac{\delta_j}{s_j} \cdot \frac{\gamma_{j+1} - \gamma_j}{2} \tag{1}
\]

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

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

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

(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

\[ \Gamma(\delta_i) = \Gamma_0(-s_i) + \frac{s_i}{4} \left( \gamma_{i+1} + \gamma_i \right) + \left( \frac{\gamma_{i+1} + \gamma_i}{2} \right) \delta_i \]

\[ + \left( \frac{\gamma_{i+1} - \gamma_i}{2} \right) \frac{\delta_i^2}{2s_i} \]

(4)

where \( \Gamma_0(-s_i) \) 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,

\[ C_{D,ij} = \frac{1}{s_{ref}} \int_{-s_i}^{+s_i} \Gamma(\delta_i) \left( \frac{w_{n,j}(\delta_i)}{U} \right) d\delta_i \]

(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:

\[ C_{D,ij} = \frac{1}{s_{ref}} \left( G_i \bar{G}_{i,j} I_{1_{ij}} + \bar{G}_i \bar{G}_{i,j} I_{2_{ij}} + \bar{G}_i \bar{G}_j I_{3_{ij}} + \bar{G}_i \bar{G}_j I_{4_{ij}} \right) \]

\[ + \bar{G}_i \bar{G}_j I_{5_{ij}} + \bar{G}_i \bar{G}_j I_{6_{ij}} \]

(6)

where the \( G_i, \bar{G}_i, \bar{G}_j, \) etc., are linear functions of the \( \gamma_j \)'s (ref. 4).

The \( I_{1_{ij}} \) through \( I_{6_{ij}} \) are again determined by wake geometry, being integrals of combinations of the \( A_{ij} \) through \( A_{4_{ij}} \) times \( \delta_i^n \), for \( n = 0, 1, 2 \), as given in reference 4.
The Munk optimization procedure uses (ref. 6)

\[
\frac{W_n}{\cos \phi} = W_0 = \text{constant}
\]  \hspace{1cm} (7)

which for the assumed wake model is written as

\[
\cos \phi_i = \frac{1}{W_0} \sum_{j=1}^{N} \left\{ \left( \frac{\gamma_{i+1} + \gamma_i}{2} \right) \left( A_{1ij} + A_{2ij} \right) \right. \\
+ \left. \left( \frac{\gamma_{i+1} - \gamma_i}{2} \right) \left( A_{3ij} + A_{4ij} \right) \right\}
\]  \hspace{1cm} (8)

This yields \( N \) equations for \( N+1 \) unknowns, the \( N \) \( \gamma_j \) values and \( W_0 \).

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

\[
C_L = \frac{8}{S} \text{ref} \left\{ \frac{1}{3} \sum_{j=1}^{N} \left( \cos \phi_j s_j \left( \frac{\gamma_{j+1}}{W_0} + \frac{2\gamma_j}{W_0} \right) \right) + \sum_{j=1}^{N} \left( \cos \phi_j s_j \frac{\Gamma_o (s_j)}{W_0} \right) \right\} \frac{W_0}{U}
\]  \hspace{1cm} (9)

The direct optimization procedure extremizes (minimizes) the function

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

\[
C_D = \left( \frac{\gamma_j}{U} \right)^T A \left( \frac{\gamma_j}{U} \right)
\]  \hspace{1cm} (11)

The optimal \( \gamma_j/U \) values are then determined by \( N+1 \) equations given by
\[
\sum_{j=1}^{N} \left( A_{1j} + A_{j1} \right) \frac{\gamma_j}{U} + \lambda c_{L,i} = 0, \quad i = 1, \ldots, N
\]  

(12)

\[
\sum_{j=1}^{N} c_{L,j} \frac{\gamma_j}{U} - c_L = 0
\]

(13)

The \( A \) matrix, as given in reference 8, is in terms of the \( I_{1i,j} \) through \( I_{6i,j} \) 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:

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Variable</th>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ICNTRL</td>
<td>(1-5)</td>
<td>A control integer which determines the wake panel spacing on the current lifting surface, as follows:</td>
</tr>
</tbody>
</table>

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
Card Number | Variable | Columns | Description
---|---|---|---
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 NLLINE surfaces must not exceed 50.

For values of ICNTRL = 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(I), 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.

C. 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.

D. 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.
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),</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>
<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:

Card 8 - zero (0)

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:

I; I = 1, ..., (NTOTT + 1)  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.

YHH(I)  Y coordinate of outboard, or tipmost, corner of wake panel I.

ZHH(I)  Z coordinate of outboard corner of wake panel I.

PPP(I)  Dihedral angle, in radians, of wake panel I.

SNN(I)  Semiwidth of wake panel I.

Finally, the following reference quantities are listed:

TOL2  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.

CLDES  The desired lift coefficient value.

SREF  The configuration reference area.

BSAVE  The configuration reference span, taken as twice the maximum absolute value of YHH.

ARAT  The configuration aspect ratio, defined as (BSAVE)^2/SREF.

Solution Data

The output data for the minimum drag solution for the Munk's criterion solution consists, first, of a table of the following:
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Individual wake panel number, numbered tip-to-root, as described above, for each lifting surface of the configuration.</td>
</tr>
<tr>
<td>BGAM(I)</td>
<td>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.</td>
</tr>
<tr>
<td>CDRAG(I)</td>
<td>Nondimensional bound circulation value for minimum induced drag, $\Gamma/\Gamma$, 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 $\Gamma$ value at the root of the planform.</td>
</tr>
<tr>
<td>AOPT(I, NTOTT + 1)</td>
<td>Wake vortex sheet strengths for minimum induced drag, $\gamma/U$, at the outboard corner of wake panel I.</td>
</tr>
<tr>
<td>GAM(I)</td>
<td>Nondimensional wake vortex sheet strength for minimum induced drag, $\gamma/\gamma$, at the outboard corner of wake panel I.</td>
</tr>
<tr>
<td>ETA</td>
<td>Nondimensional spanwise coordinate of outboard corner point of wake panel I, at which the above values are computed.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>Induced drag coefficient computed using optimum value vortex strengths, for Munk's criterion solution.</td>
</tr>
<tr>
<td>DIDEAL</td>
<td>Induced drag coefficient for a planar wing having the same projected span as the current configuration, evaluated at the same lift coefficient value.</td>
</tr>
<tr>
<td>WDBU</td>
<td>The ratio of the constant, $w_0$, appearing in the general statement of Munk's criterion [eq. (7)], divided by $U$.</td>
</tr>
<tr>
<td>DEFF</td>
<td>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.</td>
</tr>
<tr>
<td>I</td>
<td>Wake panel number.</td>
</tr>
<tr>
<td>------</td>
<td>--------------------</td>
</tr>
<tr>
<td>WDOWN</td>
<td>Computed induced normal velocity at the midpoint of wake panel I.</td>
</tr>
<tr>
<td>WOP</td>
<td>Induced normal velocity divided by the cosine of the dihedral angle, evaluated at the midpoint of wake panel I.</td>
</tr>
<tr>
<td>CDAPP</td>
<td>An approximate value of induced drag coefficient, evaluated by assuming $\Gamma$ and $\gamma$ are constant on each wake panel.</td>
</tr>
</tbody>
</table>

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, were \( d/h = 1.0 \), \( h/b = 0.355 \). 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;
\( d/h = 1.0, \ h/b = 0.355 \)

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>8</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-0.1775</td>
<td>0.0</td>
<td>-0.498451</td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.17750</td>
<td>89.5</td>
<td>-0.498451</td>
</tr>
</tbody>
</table>

Diagram:

```
     h
    /|
  d / \\
   /    \
  /      \
 /        \
/          \
/            \
/ d          \
   /         \
   /          \
   /           \
   /             \
   / b
```

\( 0 \) \( \bar{y} \)
\( \bar{z} \)
Output Data for Diamond Wing with End Slates; \(d/h = 1.0, \ h/b = 0.355\)

**GENERAL INPUT GEOMETRY**

<table>
<thead>
<tr>
<th>SEGMENT NO</th>
<th>Y</th>
<th>Z</th>
<th>PHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>1</td>
<td>-0.49845</td>
<td>0.00000</td>
<td>-19.60098</td>
</tr>
<tr>
<td>2</td>
<td>-0.50000</td>
<td>-0.17750</td>
<td>89.50000</td>
</tr>
</tbody>
</table>

**TOTAL PLANFORM PERIPHERAL LENGTH** = \(0.70662\)

**COSINE SEGMENT SPACING**

<table>
<thead>
<tr>
<th>SEGMENT 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>-1.342102</td>
</tr>
<tr>
<td>7</td>
<td>-0.483185</td>
<td>0.005436</td>
<td>-1.342102</td>
</tr>
<tr>
<td>8</td>
<td>-0.459089</td>
<td>0.014017</td>
<td>-1.342102</td>
</tr>
<tr>
<td>9</td>
<td>-0.423689</td>
<td>0.026623</td>
<td>-1.342102</td>
</tr>
<tr>
<td>10</td>
<td>-0.377557</td>
<td>0.042944</td>
<td>-1.342102</td>
</tr>
<tr>
<td>11</td>
<td>-0.322720</td>
<td>0.062578</td>
<td>-1.342102</td>
</tr>
<tr>
<td>12</td>
<td>-0.259637</td>
<td>0.085042</td>
<td>-1.342102</td>
</tr>
<tr>
<td>13</td>
<td>-0.190161</td>
<td>0.109783</td>
<td>-1.342102</td>
</tr>
<tr>
<td>14</td>
<td>-0.116002</td>
<td>0.136191</td>
<td>-1.342102</td>
</tr>
<tr>
<td>15</td>
<td>-0.038987</td>
<td>0.163616</td>
<td>-1.342102</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>-1.342102</td>
</tr>
<tr>
<td>22</td>
<td>-0.483185</td>
<td>0.005436</td>
<td>-1.342102</td>
</tr>
<tr>
<td>23</td>
<td>-0.459089</td>
<td>0.014017</td>
<td>-1.342102</td>
</tr>
<tr>
<td>24</td>
<td>-0.423689</td>
<td>0.026623</td>
<td>-1.342102</td>
</tr>
<tr>
<td>25</td>
<td>-0.377557</td>
<td>0.042944</td>
<td>-1.342102</td>
</tr>
<tr>
<td>26</td>
<td>-0.322720</td>
<td>0.062578</td>
<td>-1.342102</td>
</tr>
<tr>
<td>27</td>
<td>-0.259637</td>
<td>0.085042</td>
<td>-1.342102</td>
</tr>
<tr>
<td>28</td>
<td>-0.190161</td>
<td>0.136191</td>
<td>-1.342102</td>
</tr>
<tr>
<td>29</td>
<td>-0.116002</td>
<td>0.136191</td>
<td>-1.342102</td>
</tr>
</tbody>
</table>
Output Data for Diamond Wing with End Plates;
d/h = 1.0, h/b = 0.355 (continued)

<table>
<thead>
<tr>
<th>I</th>
<th>$\pi$N(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00848</td>
</tr>
<tr>
<td>2</td>
<td>0.02219</td>
</tr>
<tr>
<td>3</td>
<td>0.02743</td>
</tr>
<tr>
<td>4</td>
<td>0.02219</td>
</tr>
<tr>
<td>5</td>
<td>0.00848</td>
</tr>
<tr>
<td>6</td>
<td>0.00326</td>
</tr>
<tr>
<td>7</td>
<td>0.00969</td>
</tr>
<tr>
<td>8</td>
<td>0.01589</td>
</tr>
<tr>
<td>9</td>
<td>0.02169</td>
</tr>
<tr>
<td>10</td>
<td>0.02696</td>
</tr>
<tr>
<td>11</td>
<td>0.03157</td>
</tr>
<tr>
<td>12</td>
<td>0.03540</td>
</tr>
<tr>
<td>13</td>
<td>0.03835</td>
</tr>
<tr>
<td>14</td>
<td>0.04037</td>
</tr>
<tr>
<td>15</td>
<td>0.04139</td>
</tr>
<tr>
<td>16</td>
<td>0.00848</td>
</tr>
<tr>
<td>17</td>
<td>0.02219</td>
</tr>
<tr>
<td>18</td>
<td>0.02743</td>
</tr>
<tr>
<td>19</td>
<td>0.02219</td>
</tr>
<tr>
<td>20</td>
<td>0.00848</td>
</tr>
<tr>
<td>21</td>
<td>0.00326</td>
</tr>
<tr>
<td>22</td>
<td>0.00969</td>
</tr>
<tr>
<td>23</td>
<td>0.01589</td>
</tr>
<tr>
<td>24</td>
<td>0.02169</td>
</tr>
<tr>
<td>25</td>
<td>0.02696</td>
</tr>
<tr>
<td>26</td>
<td>0.03157</td>
</tr>
<tr>
<td>27</td>
<td>0.03540</td>
</tr>
<tr>
<td>28</td>
<td>0.03835</td>
</tr>
<tr>
<td>29</td>
<td>0.04037</td>
</tr>
<tr>
<td>30</td>
<td>0.04139</td>
</tr>
</tbody>
</table>

TOL IN $\pi$NTAN = 2.6057E-05

DESIGN LIFT COEF. = 0.50000

WING REFERENCE AREA = 1.00000

REF WING SPAN = 1.00000

ASPECT RATIO = 1.00000
Output Data for Diamond Wing with End Plates;
d/h = 1.0, h/b = 0.355 (continued)

<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>30907E+01</td>
<td>31321E+02</td>
<td>10000E+01</td>
</tr>
<tr>
<td>2</td>
<td>3309E-01</td>
<td>29612E+00</td>
<td>81359E+00</td>
<td>82448E+01</td>
<td>99970E+00</td>
</tr>
<tr>
<td>3</td>
<td>6248E-01</td>
<td>55920E+00</td>
<td>51136E+00</td>
<td>51820E+01</td>
<td>99803E+00</td>
</tr>
<tr>
<td>4</td>
<td>8756E-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>11089E+00</td>
<td>99232E+00</td>
<td>15637E+01</td>
<td>15847E+00</td>
<td>94811E+00</td>
</tr>
<tr>
<td>9</td>
<td>11156E+00</td>
<td>99831E+00</td>
<td>26488E-01</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>30907E+01</td>
<td>31321E+02</td>
<td>10000E+01</td>
</tr>
<tr>
<td>17</td>
<td>3309E-01</td>
<td>29612E+00</td>
<td>81359E+00</td>
<td>82448E+01</td>
<td>99970E+00</td>
</tr>
<tr>
<td>18</td>
<td>6248E-01</td>
<td>55920E+00</td>
<td>51136E+00</td>
<td>51820E+01</td>
<td>99803E+00</td>
</tr>
<tr>
<td>19</td>
<td>8756E-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>11089E+00</td>
<td>99232E+00</td>
<td>15637E+01</td>
<td>15847E+00</td>
<td>94811E+00</td>
</tr>
<tr>
<td>24</td>
<td>11156E+00</td>
<td>99831E+00</td>
<td>26488E-01</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
Output Data for Diamond Wing with End Plates;
\[ d/h = 1.0, \ h/b = 0.355 \] (continued)

\[ \begin{array}{ccc}
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
200 & \text{ENTERED} \\
\end{array} \]

CD Calculated using sub DRACAL and optim loads using MUNK CRIT = 

CD for flat wing = \[ .79577E-01 \]
RATIO OF ZERO DIHEDRAL DOWNWASH/U = \[ .38317E+00 \]

INDUCED DRAE EFFICIENCY FOR WINGS OF EQUAL SPAN = \[ .166319E+01 \]

<table>
<thead>
<tr>
<th>I</th>
<th>DOWNWASH</th>
<th>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>

26
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 ROUND 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

10 PROGRAM DNWASH (INPUT, OUTPUT, TAPES, TAPEN, OUTPUT)

DIMENSION GAW(51), HGM(51), YHH(51), ZHH(51), PPP(51)
DIMENSION DNWNSH(51)
DIMENSION AOPT(52,52)
DIMENSION AINT(6)
DIMENSION CDRAG(51)
DIMENSION ASIP(51,51), RSP(51,1), IPIVOT(51)
DIMENSION YY(10), PP(10), LSE6(10), DTU(10)
DIMENSION PERIP(10), R(10), GS(10)
DIMENSION CLP(51), SE(10,1), DTA(10)
COMMON /FILL/ TOL, TOL2, TOL3
COMMON /SLG/ SNN(51)
COMMON /DIROPT/ T1(53,53), T2(53,53), T3(53,53), T4(53,53), T5(53,53)
COMMON /FEN/ NSPT(10), NLINE

PI = 4. * ATAN(1.1)
DTP = PI/180.

PROGRAM WRITTEN TO IMPLEMENT TREFFZTZ PLANE INDUCED DRAG

THEORY ASSUMES A TWO DIMENSIONAL ADVANCED PANEL MODEL OF
THE UNDISTURBED, INTERACTING WAVES OF MULTIPLE LIFTING
SURFACES.

WAKE VORTEX SHFT STRENGTHS ARE ASSUMED TO VARY IN A
PIECEWISE LINEAR FASHION. ANALYTICAL EXPRESSIONS FOR
INDUCED NORMAL VELOCITY, ROUND CIRCULATION, INDUCED DRAG,
AND LIFT ARE DEVELOPED IN CR-3154 IN TERMS OF THE ASSUMED
WAKE MODEL.

THESE EXPRESSIONS ARE USED TO OBTAIN MINIMUM DRAG WAKE VORTEX
SHFT STRENGTHS, ROUND CIRCULATION DISTRIBUTIONS, AND CO
VALUES FOR MINIMUM DRAG AT A GIVEN LIFT, FOR NONPLANAR MULTIPLE
INTERACTING LIFTING SURFACES, USING BOTH HUNK'S CRITERION AND
A DIRECT OPTIMIZATION PROCEDURE.

1, T1(53,53), T2(53,53), T3(53,53), T4(53,53), T5(53,53)

1, T1(53,53)

1, T1(53,53)
THEN MUST INPUT TRIPLES OF YMM, ZMM, PPP FOR EACH CORNER POINT
OF EACH SEGMENT, STARTING AT WING TIP.
ICTRL=6 FOR CIRCULAR ARC WING, CONE, TR 120
ICTRL=7 FOR WING OF ARBITRARY PHI AND ETA-EQUAL SPACING
ICTRL=8 FOR WING OF ARBITRARY PHI AND ETA-COSINE SPACING

NTOT=NUMBER OF SEGMENTS ON SEMISPAN, BOTH FOR EQUAL AND COSINE SPA
READ AN INTEGER VARIABLE NLLINE =RENNT LIFTING LINES NEEDED TO MAKE
NTOTT = NUMBER OF SEGMENTS ON NLLINE
NS = START NUMBER ON EACH NLLINE

READ (5,131) NLLINE
IF (NLLINE,F0.0) GO TO 129
NTOTT = 0
LSTART = 1
DTOM = 0.0
WRITE (6,130)
WRITE (6,135)
CONTINUE
READ (5,131) ICNTRL,NTOT
IF (ICNTRL,F0.0) GO TO 1
NS = NTOTT+1
MSPT(LSTART) = NS
NTOT = NTOTT+NTOT
ISPTP = 1
ICRAG = 1
IF (ICNTRL:F0.3) GO TO 12
IF (ICNTRL,F0.6) GO TO 15
IF (ICNTRL,F0.7) GO TO 20
IF (ICNTRL,F0.8) GO TO 20
CALCULATE DOWNWASH AT SEGMENT MIDPOINTS
WRITE (6,133)
DO 11 I=1,NTOTT
WRITE = 0.
DO 10 J=1,NTOTT
S = 5NN(J)
CALL CCAL (1, J, YH1, YH2, PPP, S, AA, UB, DU, FF, GG, EF, AJ, AK, PP, TT, UU, WW)
CALL CONCAL (AA, AN, FF, GG, S, AH, C, D, F, G, CJ, CK, CL, CN, CO, CP, J)
IF (RR, EQ, 0.) GO TO 4
P = .5 *(ATAN(C/ARS(RR)) - ATAN(D/ARS(RR)))/(ARS(RR))
GO TO 5

CONTINUE

P = 2./((FF - 2.*S) - 2./(FF + 2.*S))

CONTINUE

IF (UU, EQ, 0.) GO TO 6
Q = 2.* (ATAN((AJ - 2.*S) / ARS(UU)) - ATAN((AJ - 2.*S) / ARS(UU)) / ARS(UU))
GO TO 7

CONTINUE

Q = 2.*(AJ - 2.*S) - 2./(AJ + 2.*S)

CONTINUE

Z1 = (S*S + FF*S + GG)/(S*S - FF*S + GG)
Z2 = (S*S + AJ*S + AK)/(S*S - AJ*S - AK)
A11J = ((P*A + S*AR*ALOG(Z1))/(2.*PI))
A31J = ((CL*P*A + AR*S*CO*ALOG(Z1))/(2.*5*PI))
A21J = (-Q*A + S*EE*ALOG(Z2))/(2.*PI)
A41J = (-CL*Q*A + 2.*EE*S*CO*ALOG(Z2))/(2.*5*PI)
GO R = K = 1, NLLIKE
KR = K+1
KCHK = KSTP(KK) - 1
IF (J, EQ, KCHK) GO TO 9

CONTINUE

WDOWN = WDOWN + .5*(GAM(J + 1) + GAM(J) - (A11J + A21J) + .5*(GAM(J + 1) - GAM(J))
*(A31J + A41J)
GO TO 10

CONTINUE

WDOWN = WDOWN + .5*GAM(J) + (A11J + A21J) - A31J - A41J

CONTINUE

DOWNWASH AT WING IS .5 WASH AT MINUS INFINITY

C

DOWNWASHAT WING IS .5 WASH AT MINUS INFINITY

C

WDOWN = WDOWN/7.
WOP = WDOWN/COS(PPP(1))
WRITE (*, 134) I, WDOWN, WOP
DNWSHJ = WDOWN

CONTINUE

IF (ISTOP, EQ, 0.) GO TO 124
ISTOP = ISTOP + 1

CONTINUE

CONTINUE
PROGRAM DNWASH

GO TO 14
C
C GENERAL GEOMETRY CALCULATIONS
C

130 CONTINUE
READ (5,136) (YHH(I),ZHH(I),PPP(I),I=NS,NTOT)
HOT = ABS(YHH(NS)+SIN(NS)*COS(PPP(NS))
R(START) = 2.*HOT
DTOT = 0.,
DO 13 I=1,NTOT
DTOT = DTOT+SN(1)
13 CONTINUE
DTOT = 2.*DTOT
WRITE (6,139) DTOT
DTOT(START) = DTOT
DTOT = DTOT+DTOT
14 CONTINUE
GO TO 19

145 C
C GEOMETRY FOR CIRCULAR ARC AIRFOIL
C

150 CONTINUE
NTOT = NTOT+1
READ (5,136) FFT,THET,HOT
THET = THET*DTR
R(START) = 2.*HOT
D = HOT*HOT
R = HOT/SIN(THET)

155 DTHETA = THET/FLOAT(NTOT)
DO 16 I=1,NTOT
YT = -R*SIN(THET-FLOAT(I-1)*DTHETA)
ZT = -R*COS(THET-FLOAT(I-1)*DTHETA)
II = NS-I+1
PPP(II) = ATAN((ZT-ZTL)/(YT-YTL))

165 SNN(II) = .5*SQRT((YT-YTL)**2+(ZT-ZTL)**2)
YHH(II) = .5*(YT+YTL)
ZHH(II) = .5*(ZT+ZTL)

16 CONTINUE
DTOT = 0.,
DO 17 I=NS,NTOT

A 127
A 128
A 129
A 130
A 131
A 132
A 133
A 134
A 135
A 136
A 137
A 138
A 139
A 140
A 141
A 142
A 143
A 144
A 145
A 146
A 147
A 148
A 149
A 150
A 151
A 152
A 153
A 154
A 155
A 156
A 157
A 158
A 159
A 160
A 161
A 162
A 163
A 164
A 165
A 166
A 167
A 168
PROGRAM DNWASH

170 CONTINUE
170 DTOT = DTOT + SHN(I)
170 DTOT = 2. * DTOT
170 WRITE (6, 138) DTOT
170 DTOT(LSTART) = DTOT
170 DTOT = DTOT + DTOT
175 IF (NSN+NF, 1) GO TO 18
175 DTOT = 1. * DTOT
180 CONTINUE
180 CONTINUE
180 GO TO 20

C
20 CONTINUE
C
20 CONTINUE
C
GEOMETRY CALCS FOR WAKE MADE OF STRAIGHT SEGMENTS (CONT'D)
C
185 READ (5, 137) NBRK
185 NBR = NBRK-1
185 READ (5, 137) (LSEG(I), I=1, NBR)
190 NBRK EQUALS NUMBER OF DIHEDRAL CHANGES OR JETS WITH
190 OTHER LIFTING LINES +2
190 READ (5, 136) (YY(I), I=1, NBRK)
190 DO 21 I=1, NBRK
190 21 WRITE (6, 132) YY(I), I=1, NBRK
190 DO 22 I=1, NBR
195 DTOT = 0.
195 DO 23 I=1, NBR
195 23 DTOT = DTOT + PERIF(I)
195 WRITE (6, 138) DTOT
195 DTOT(LSTART) = DTOT
195 DTOT = DTOT + DTOT
200 IF (NSN+NF, 1) GO TO 24
200 DTOT = DTOT
205 PP(I) = DTH * PP(I)
205 K(LSTART) = 2. * K(LSEG(NBRK))
205 IF (ICHTL.EQ.1) GO TO 30
205 DO 26 I=1, NBR
205 26 CONTINUE
210 CONTINUE
PROGRAM DNWASH

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>215</td>
<td>S = SEQ(NBK)</td>
<td>A 211</td>
</tr>
<tr>
<td>216</td>
<td>SNN(NS) = S</td>
<td>A 212</td>
</tr>
<tr>
<td>217</td>
<td>PPP(NS) = PP(NBK)</td>
<td>A 213</td>
</tr>
<tr>
<td>218</td>
<td>YHH(NS) = YY(NBK) + S * COS(PPP(NS))</td>
<td>A 214</td>
</tr>
<tr>
<td>219</td>
<td>ZHH(NS) = Z(NBK) + S * SIN(PPP(NS))</td>
<td>A 215</td>
</tr>
<tr>
<td>220</td>
<td>N1 = NS</td>
<td>A 216</td>
</tr>
<tr>
<td>221</td>
<td>DO 28 J = 1, NBK</td>
<td>A 217</td>
</tr>
<tr>
<td>222</td>
<td>NSEG = N1 * LSEG(NBK - J - 1)</td>
<td>A 218</td>
</tr>
<tr>
<td>223</td>
<td>II = N1 + 1</td>
<td>A 219</td>
</tr>
<tr>
<td>224</td>
<td>LL = NRP - J</td>
<td>A 220</td>
</tr>
<tr>
<td>225</td>
<td>LH = LL + 1</td>
<td>A 221</td>
</tr>
<tr>
<td>226</td>
<td>DO 27 I = II, NSEG</td>
<td>A 222</td>
</tr>
<tr>
<td>227</td>
<td>SNN(I) = SEQ(LF)</td>
<td>A 223</td>
</tr>
<tr>
<td>228</td>
<td>YHH(I) = YHH(I - 1) + 2.0 * SEQ(LH) * COS(PPP(N1))</td>
<td>A 224</td>
</tr>
<tr>
<td>229</td>
<td>ZHH(I) = ZHH(I - 1) + 2.0 * SEQ(LH) * SIN(PPP(N1))</td>
<td>A 225</td>
</tr>
<tr>
<td>230</td>
<td>PPP(I) = PPP(N1)</td>
<td>A 226</td>
</tr>
<tr>
<td>231</td>
<td>N1 = N1 * LSEG(NBK - J)</td>
<td>A 227</td>
</tr>
<tr>
<td>232</td>
<td>IF (LL &lt;= 0.0) GO TO 2A</td>
<td>A 228</td>
</tr>
<tr>
<td>233</td>
<td>SNN(N1) = SEQ(LL)</td>
<td>A 229</td>
</tr>
<tr>
<td>234</td>
<td>YHH(N1) = YHH(NSEG) + COS(PPP(LH)) * SEQ(LL) * COS(PPP(NSEG)) * SEQ(LH)</td>
<td>A 230</td>
</tr>
<tr>
<td>235</td>
<td>ZHH(N1) = ZHH(NSEG) + SIN(PPP(LH)) * SEQ(LL) * SIN(PPP(NSEG)) * SEQ(LH)</td>
<td>A 231</td>
</tr>
<tr>
<td>236</td>
<td>PPP(N1) = PP(LH)</td>
<td>A 232</td>
</tr>
<tr>
<td>237</td>
<td>CONTINUE</td>
<td>A 233</td>
</tr>
<tr>
<td>238</td>
<td>2A CONTINUE</td>
<td>A 234</td>
</tr>
<tr>
<td>239</td>
<td>60 TO 42</td>
<td>A 235</td>
</tr>
</tbody>
</table>

C COSINE SPACING CALCULATIONS

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>240</td>
<td>CONTINUE</td>
<td>A 236</td>
</tr>
<tr>
<td>241</td>
<td>30 CONTINUE</td>
<td>A 237</td>
</tr>
<tr>
<td>242</td>
<td>IF (YY(I), EQ, 0.1) GO TO 32</td>
<td>A 238</td>
</tr>
<tr>
<td>243</td>
<td>NO 31 I = 1, NRP</td>
<td>A 239</td>
</tr>
<tr>
<td>244</td>
<td>31 DTHETF(I) = PI / (FLOAT(LSEG(NBK - I)))</td>
<td>A 240</td>
</tr>
<tr>
<td>245</td>
<td>60 TO 35</td>
<td>A 241</td>
</tr>
<tr>
<td>246</td>
<td>32 DO 34 I = 1, NRP</td>
<td>A 242</td>
</tr>
<tr>
<td>247</td>
<td>IF (I, EQ, NRP) GO TO 33</td>
<td>A 243</td>
</tr>
<tr>
<td>248</td>
<td>THE VARIABLE DTHETA IS NAMED ARRAY DTHETF(I) IN ICNTREAL</td>
<td>A 244</td>
</tr>
<tr>
<td>249</td>
<td>DTHETF(I) = PI / (FLOAT(LSEG(NBK - I)))</td>
<td>A 245</td>
</tr>
<tr>
<td>250</td>
<td>33 CONTINUE</td>
<td>A 246</td>
</tr>
<tr>
<td>251</td>
<td>GO TO 34</td>
<td>A 247</td>
</tr>
<tr>
<td>252</td>
<td>34 CONTINUE</td>
<td>A 248</td>
</tr>
<tr>
<td>253</td>
<td>IF (NBK, EQ, 1) GO TO 35</td>
<td>A 249</td>
</tr>
</tbody>
</table>
PROGRAM DWASH

SN = 0.5*PERFIFA*(1.0-COS(DTHETA))
GO TO 36

35 FH = 0.5*PERFIFA(NNR)
SN = 0.5*RH*(1.0-COS(DTHETA))
36 PPP = PP(NR)
NN = 0
DO 40 I=1,NPP
NN = NN + LSRF(NPP-I)
IF (I,FO,NN,AND,0,YY(I),WU,0) GO TO 37
NN = 0.5*PERFIFA(NPP-I)
GO TO 38

37 CONTINUE

38 CONTINUE

LL = LSRF(NPP-I)
DO 99 J=1,LL
IF (I,FO,NN,J,AND,0) GO TO 39
NN = NN + LL
KM = NN + L
PPP = PP(NPP-I)
SN = 0.5*RH*(COS(0.0) + COS(DTHETA))
GO TO 39

39 CONTINUE

40 CONTINUE

YH(I) = YH(I-1)*SN(I-1)*COS(PPP(I))
ZH(I) = ZH(I-1)*SN(I-1)*SIN(PPP(I))

41 CONTINUE

42 CONTINUE

IF (LSTART.EQ.NLINF) GO TO 43
LSTART = 1
N=1+1
GO TO 43

43 CONTINUE

IF (ICMPL.EQ.N) WRITE (6,139)
IF (ICMPL.EQ.1) WRITE (6,140)
IF (ICMPL.EQ.2) WRITE (6,141)
LN = NLINF+1
NSPT(1:N) = NTOTT + 1
WRITE (6,147)
WRITE (6,143) (I, YH(I), ZH(I), PPI(I), I=1, NTOTT)
WRITE (6,144)
WRITE (6,145) (I, YH(I), I=1, NTOTT)

READ, WRITE CL SREF, ETC
C
READ (5,136) C, DES, SREF
BSAVE = H(I)
IF (NINF.EQ.0,1) GO TO 45
DO 44 I = 2, NLT
RTFM = P(I)
44 BSAVE = AMAX1(BSAVE, RTFM)
CONTINUE
ARAT = BSAVE * SAVF / SRF
SMIN = SMH(I)
DO 46 I = 2, NTOTT
STEM = SMH(I)
46 SMIN = AMIN1(SMIN, STEM)
TOL = 5.E-05 * SMIN * NSPT(2)
TOL2 = TOL
TOL3 = 0.005 * SMIN
WRITE (6,157) TOLZ
C
END GENERAL FROM CALCS
C
47 CONTINUE
IF (IDRAG, EQ, 7) GO TO 52
WRITE (6,146) C, DES, SREF, BSAVE, ARAT
WRITE (6,149)
IDRAG = IDRAG + 1
C
SET UP ALL A FOR MIN CRITERION OPTIMIZATION
DO 49 J = 1, NTOTT
DO 48 J = 1, NTOTT
CALL WCAL (I, J, NTOTT, YH(I), ZH(I), PPI(I), AOPT(I,J))
AOPT(I,J) = AOPT(I,J) + 2. / SRF
48 CONTINUE
49 CONTINUE
TL = NTOTT + 1
DO 35 I = 1, NTOTT
COS(I) = COS(PPI(I))
AOPT(I,J) = COS(PPP(I))

CONTINUE

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

IL1 = NTOTT
C ABOVE II1 IS TEMPORARY SET LT NTOTT
GO TO 77

CONTINUE
C SET UP ALL A THROUGH F FOR ALL I,J TO DO DIRECT OPTIMIZATION
[DPRAG = IDPRAG+1]
IF (NLINE.NF.MF,1) GO TO 7

NTOTT2 = NTOTT + 2
NTOTT3 = NTOTT + 3
DO 53 I = 1, NTOTT3
  DO 53 J = 1, NTOTT3
  TI(I,J) = 0.
  T2(I,J) = 0.
  T3(I,J) = 0.
  TS(I,J) = 0.
  T6(I,J) = 0.
CONTINUE

DO 54 J = 1, NTOTT
  J1 = J + 1
  DO 54 I = 1, NTOTT
    II = I + 1
    CALL DPAACAL (II, J1, YHH, ZHH, PPP, AINT)
    TI(I,J1) = AINT(1)
    T2(I,J1) = AINT(2)
    T3(I,J1) = AINT(3)
    T4(I,J1) = AINT(4)
    TS(I,J1) = AINT(5)
    T6(I,J1) = AINT(6)
    CONTINUE
  CONTINUE
C DIRECT OPTIMIZATION MODS

DO 56 J = 1, NTOTT
  J1 = J + 1
  DO 56 I = 1, NTOTT
J1 = J1 + 1

385
IF (I,F0.1) GO TO 55
A0PT(I,J) = A0PT(I,J) + 5*SNN(I-1) * (T1(T1-1,J1-1) - T2(T1-1,J1-1) + T1(T1-1,J1-1) + T2(T1-1,J1-1)) A386

55 CONTINUE
A0PT(I,J) = A0PT(I,J) * 0.25 A387

390
56 CONTINUE
DO 64 I=1,N
J1 = J1 + 1
IF (I,F0.1) GO TO 60
SCON = 0.5 * (SNN(I-1) + SNN(I-1)) A391
TEMP = 0.
DO 59 J=1,N
J1 = J1 + 1
TEMP = TEMP + SCON * (T1(T1,J1-1) - T2(T1,J1-1) + T1(T1,J1-1) - T2(T1,J1-1)) A392
IF (I,EQ.75) GO TO 58
IS = IS + 1 A393
DO 57 IP=IS,N
IPI = IPI + 1
TEMP = TEMP + (SNN(I-1) * (T1(T1,J1-1) - T2(T1,J1-1) + T1(T1,J1-1) - T2(T1,J1-1))) A394
57 CONTINUE
TEMP = TEMP + SCON * (T1(IPI,J1-1) - T2(IPI,J1-1) + T1(IPI,J1-1) - T2(IPI,J1-1)) A395
55 CONTINUE
A0PT(I,J) = A0PT(I,J) + TEMP A396
A0PT(I,J) = A0PT(I,J) / SREF A397

59 CONTINUE
GO TO 63 A398

60 CONTINUE
SCON = 0.5 * SNN(I-1) A399
DO 62 J=1,N
J1 = J1 + 1
TEMP = 0.
DO 61 IP=J1,N
IPI = IPI + 1
TEMP = TEMP + SCON * (T1(IPI,J1-1) - T2(IPI,J1-1) + T1(IPI,J1-1) - T2(IPI,J1-1)) A400
61 CONTINUE
A0PT(I,J) = A0PT(I,J) + TEMP A401
A0PT(I,J) = A0PT(I,J) / SREF A402

55 CONTINUE
GO TO 63 A403

60 CONTINUE
SCON = 0.5 * SNN(I-1) A404
DO 62 J=1,N
J1 = J1 + 1
TEMP = 0.
DO 61 IP=J1,N
IPI = IPI + 1
TEMP = TEMP + SCON * (T1(IPI,J1-1) - T2(IPI,J1-1) + T1(IPI,J1-1) - T2(IPI,J1-1)) A405
61 CONTINUE
A0PT(I,J) = A0PT(I,J) + TEMP A406
A0PT(I,J) = A0PT(I,J) / SREF A407

59 CONTINUE
GO TO 63 A408

60 CONTINUE
63 CONTINUE
64 CONTINUE
DO 65 J=1,NTOTT
DO 65 J=1,NTOTT
425
T(I,J) = AOPT(I,J)
DO 66 I=1,NTOTT
DO 66 J=1,NTOTT
AP(1,J) = 2.*(AP(1,J)+T(I,J))
66 CONTINUE
430 IL = NTOTT+1
II = IL
II = II + 1
DO 67 II = 1,II
AP(1,IL) = 0.
435 CONTINUE
AP(II,II) = CLDFS
AP(II,II) = P*COS(PP(II))*SNN(II)**2/(3.*SREF)
DO 68 I=2,NTOTT
AP(II,II) = (2*COS(PP(II))*SNN(II)**2+COS(PP(II-1))*SNN(I-1)**2)*
I4./(3.*SREF)
440 69 CONTINUE
SUMX = 0.
DO 69 I=2,NTOTT
SUMX = SUMX*COS(PP(I))*SNN(I)
445 69 CONTINUE
AP(II,II) = AOPT(II,II)*(4./SREF)*SNN(II)*SUMX
DO 72 II = 2,NTOTT
SUMX = COS(PP(I))*SNN(I)*SNN(I-1)
II = II + 1
450 IF (II .EQ. NTOTT) GO TO 71
DO 70 J=1,NTOTT
SUMX = SUMX*COS(PP(J))*SNN(J)*(SNN(I)+SNN(I-1))
70 CONTINUE
71 CONTINUE
AP(II,II) = AOPT(II,II)*(4./SREF)*SUMX
72 CONTINUE
C
DO 73 I=1,NTOTT
AP(II,II) = 2.*AP(II,II)
460 73 CONTINUE
DO 74 I=1,NTOTT
AP(II,II) = AOPT(II,II)
74 CONTINUE
   DO 75 I=1,IL1
   CLP(I) = AOPT(I,IL1+1)
75 CONTINUE
   WRITE(*,147) (CLP(I),I=1,IL1)
   AOPT(IL1+1,IL1) = 0.
   DO 76 I=1,IL1
   DO 76 J=1,IL1
   ASIM(I,J) = AOPT(I,J)
   HSIM(I+1) = AOPT(I+1,IL2)
76 CONTINUE
   WRITE(*,148)
   77 CONTINUE
   CALL SIMEQ (ASIM,IL1,RSIM,DET,PIVOT,IS,SCALE)
   IT = IL1+1
   DO 78 I=1,IL1
78 AOPT(I,IT) = PSIM(I,1)
   C SET IL1 BACK TO NTOTT+1
   IL1 = NTOTT+1
   IF (INPGF,NE.3) GO TO 81
   CLCHK = 0.
   DO 79 I=1,NTOTT
   AOPT(I,IL2) = RSIM(I,1)
   CLCHK = CLCHK+CLP(I)*AOPT(I,IL2)
79 CONTINUE
   WRITE (*,150) CLCHK
   DO 80 I=1,IL1
60 WRITE (*,152) I,AOPT(I,IL2)
   P1 CONTINUE
   WRITE (*,151)
   C
   C CALCULATE RADIUS CIRCULATIONS AND
   C CALCULATE MESH DOWNWASH DIVIDED BY FREESTREAM U
   C
   NTOIT = NTOTT+1
   SUGAM = 0.
   AOPT(NTOIT,IL1) = 0.
   500 IF (INPGF,NE.3) GO TO 83
   DO 82 I=1,IL1
   82 AOPT(I,IL) = AOPT(I,IL2)
   AOPT(1,IL) = 0.
   83 CONTINUE
505  DO 84 I=1,MTOT
      CALL GAMCAL (1,IL,SNN,AOPT,RGAM0)
      RGAM(I) = BGAM0
  84 CONTINUE
      CALL GAMCAL (I,IL,SNN,AOPT,RGAM0)
  510  A505
      IF (NLN线F,FIN1) GO TO 90
      MLI = NLLINE-1
      NI = NLLINE-I+1
      II = NI+1
      JF = NSPT(I)-1
      SUM(I) = RGAM(JF)*AOPT(JF,IL)*SNN(JE)
  515  A515
  85 CONTINUE
      DO 89 I=1,MLI
         I1 = NLLINE-I+2
         JF = NSPT(I)-1
         YT = YHH(JF)*SNN(JF)*COS(PPP(JE))
         IF (ARS(YT) .LT. 0.0001) GO TO 89
         JF = NSPT(2)-1
         DO 86 J=1,JF
            JS = J
            IF (YT.LT.YH(J)) GO TO 87
      86 CONTINUE
      DO 88 J=JS,JF
         RGAM(J) = RGAM(J)+SUM(I)
      88 CONTINUE
      SUMGAM = 0.
      DO 94 I=1,N1011
         NN = NLLINE
         JJ = J+1
         JCHK = NSPT(JJ)-1
         IF (JCHK.GT.0) GO TO 92
      90 CONTINUE
      SUMGAM = SUMGAM+COS(PPP(I))*SNN(I)*2*AOPT(I,IL)+2*AOPT(I,IL)
         I/I/3*SNN(I)*RGAM(I)
      GO TO 93
  540  A540
  91 CONTINUE
      SUMGAM = SUMGAM+COS(PPP(I))*SNN(I)*2*AOPT(I,IL)+3*SNN(I)*RGAM(I)
      GO TO 93
  545  A545
  92 CONTINUE
      SUMGAM = SUMGAM+COS(PPP(I))*SNN(I)*2*AOPT(I,IL)+3*SNN(I)*RGAM(I)
      GO TO 93
  545  A545
  93 CONTINUE
      SUMGAM = SUMGAM+COS(PPP(I))*SNN(I)*2*AOPT(I,IL)+3*SNN(I)*RGAM(I)
      GO TO 93
  545  A545
530  A530
  88 CONTINUE
  531  A531
  90 CONTINUE
  532  A532
  533  A533
  94 CONTINUE
  534  A534
  535  A535
  91 CONTINUE
  536  A536
  537  A537
  92 CONTINUE
  538  A538
  539  A539
  540  A540
  541  A541
  542  A542
  543  A543
  544  A544
  545  A545
  546  A546
PROGRAM DNWASH

94 CONTINUE
IF (NILINE.NE.1) GO TO 95
JE = NSPT(2)-1

550 CONTINUE
RGAM(NTOT) = RGAM(JF)+AOPT(JF,IL)*SNH(JE)
WNU = C1*VBEF(N,SUMGAM)

C
C HESCALE OPT SHEET STRENGTHS
C TO BE DIVIDED BY H INSTEAD OF W
C
IF (IDRAG.0.3) GO TO 97
DO 96 I=1,NTOT
RGAM(I) = RGAM(I)*WNU

96 AOPT(I,IL) = AOPT(I,IL)*WNU
97 CONTINUE
C
C HESCALE ROUND SHEET STRENGTHS TO CALCULATE AVG NON-DIM VALUES
C
560 CONTINUE
C AOPT(I,IL) = OPT SHEET SHEET STRENGTHS
C RGAM(I) = OPT ROUND CIJC VALUES
C IDRAG(I) = OPT NON DIM ROUND CIJCS
C GAM(I) = OPT NON DIM SHEET SHEET VALUES

570 SUBGAM = 0.
SUBGAM = 0.
NS = NSPT(2)-1
DO 100 I=1,NS
TFMP = AOPT(I+1,IL)
DO 99 J=1,NILINE
JJ = J+1
JFHK = NSPT(JJ)-1
IF (1.NE.JFHK) GO TO 98
TEMP = 0.

98 CONTINUE
99 CONTINUE
SUBGAM = SUMGAM*SNN(1)*(TFMP*AOPT(I,IL))
SUBGAM = SUMGAM*SNN(1)*(RGAM(I)+(SNN(1)/12.)*4.*TFMP*AOPT(I,IL))

100 CONTINUE
SUBGAM = SUMGAM/DSAM
SUBGAM = SUMGAM/DSAM
NS = NSPT(2)-2
PROGRAM DNWASH

590
      NS1 = NS+1
      GO TO 101
   101 CONTINUE
      CDAG(I+1) = PGAM(I+1)/SUBGAM
      GO TO 102
   102 CONTINUE
   103 CONTINUE
      JL = NSPT(I)-1
      CDAG(NTOT1) = RGAM(JL)*AOPT(JL+1)*SNN(JL)
   600      CDAG(NTOT1) = CDAG(NTOT1)/SUBGAM
      DO 104 = 1,NTOT1
      GAM(I) = AOPT(I+1)/SUBGAM
      CONTINUE
      CDAG(I) = 0.
      GAM(NTOT1) = 0.
   605      IF (MLLT(I,F+1)) GO TO 109
      JS = NSPT(I)
      II = I+1
   610      JF = NSPT(II)-1
      SUBGAM = 0.
      GO TO 106
      TEMP = AOPT(J+1,F)
      IF (J+1,II) GO TO 105
   615      TEMP = A.
   105 CONTINUE
      SUBGAM = SUBGAM*2.*SNN(J)*(RGAM(J)+(SNN(J)/2.)*(4.*TEMP*B*AOPT(J
      1,II))
   106 CONTINUE
   620      SUBGAM = SUBGAM/DTO(I)
      DO 107 = JS,JF
      CDAG(I) = PGAM(J)/SUBGAM
      CONTINUE
   107 CONTINUE
   108 CONTINUE
   625      DO 109 = 1,NTOT1
      IF (1,FG,NTOT1) GO TO 110
   109      FTA = 2.*[1+1]**(1)*SIN(I)*COS(PPI(J))/SAVE
      GO TO 111
   110 CONTINUE
   630
`FTA = 0.
111 CONTINUE
WRITE (6,152) I,FRAM(I),CDRAG(I),AOPT(I,IL),GAM(I),FTA
112 CONTINUE

C
C INDUCED DRAG CALC USING ANALYTICAL INT OF WN TIMES ROUND CIRC

C
CD = 0.
NLL = NLL(1)*F+1
DO 119 I=1,NIOTT
DO 113 K=1,NLL
ICHK = NSPT(K)-1
IF (ICHK+IQ,Il) GO TO 114
113 CONTINUE
XI = FRAM(I)*.5*SN(N(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 = FRAM(I)*.75*SN(N(I))*AOPT(I,IL)
YI = .5*AOPT(I,IL)
ZI = -YI
115 CONTINUE
DO 119 J=1,NIOTT
DO 116 K=1,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 118
117 CONTINUE
YJ = 0.5*AOPT(J,IL)
ZJ = -YJ
118 CONTINUE
CALL DRFCAL (I,J,YHH,2YY,PP,P,AINT)
CDI = XI*YJ*AINT(1)+XI*YJ*AINT(2)+YJ*YJ*AINT(3)+YJ*YJ*AINT(4)+YJ*YJ*AINT(5)
IJ*AINT(6)
CD = CD*CDI
119 CONTINUE
CI = CD*2./5*REI
IF (IDRAG,EQ,3) GO TO 120
`
PROGRAM QHWSH

WRITE (*,154) CD
GO TO 121

675 120 CONTINUE
WRITE (*,153) CD
121 CONTINUE
DINFI = SF6*CI*DS**2/(P1*(HSAVE)**2)
IF (IMAG.EQ.3) GO TO 122
WRITE (*,155) DINFI, WDFH

122 CONTINUE
DEFF = DINFI/CU
WRITE (*,156) DEFF
ISTOP = ISTOP + 1
DO 123 I=1,MTII
123 GAN(I) = AOPT(I,II)
GO TO 3

124 CONTINUE
CDAPP = 0.
DO 128 I=1,N1
DO 129 J=1,MI,1
JJ = J+1
JCIK = 25*(JJ)-1
IF (I,J,F,JCHK) GO TO 126
125 CONTINUE
CDAPP = (CHAPP + *SNH(I)*QWASH(I)*(HGA(I) + 25 * AOPT(I,II)) + 3 * AOPT(I,II) * QN(I))
GO TO 127
126 CONTINUE
CDAPP = CHAPP + 2 * QN(I) * QWASH(I) * (HGA(I) + 75 * AOPT(I,II) * QN(I))
127 CONTINUE
128 CONTINUE
CDAPP = 4 * CDAPP/SHF
WRITE (*,158) CDAPP
IF (IMAG.EQ.3) GO TO 2
GO TO 41

129 CONTINUE

130 FORMAT(1X11)
131 FORMAT(5I5)
132 FORMAT(25X,3F15.5)
133 FORMAT(//33X,H1,7X,8HDFQNWASH,7X,10HWS/COS(HF)/)  
134 FORMAT(30X,15Z)  
135 FORMAT(/30X,22HGFINFRAL INPUT GEOMETRY/)
PROGRAM DNWASH

715 136 FORMAT(F10.0)
137 FORMAT(10I5)
138 FORMAT(30X,3I10,TOTAL PLANFORM PERIPHERAL LENGTH = ,F15.5/)
139 FORMAT(30X,17HCIRCULAR ARC WING)
140 FORMAT(30X,1HEQUAL SEGMENT SPACING)
720 141 FORMAT(10X,2HCOSINE SEGMENT SPACING)
142 FORMAT(24X,FNSFGMP NO.XX,1MY+15X,1HZ+13X,3HPI1/)
143 FORMAT(25X,15.6F15.5)
144 FORMAT(27X,1HXX,3HSMN(:1)1)
145 FORMAT(25X,15,F12.5)
725 146 FORMAT(/// 25X,2HDESIGN LIFT COLF. = ,F10.5//,25X,2H wing reference
1ICE APEA = ,F10.5//,25X, 1HHEF WING SPAN = ,F10.5//,25X,
215HASPCT RATIO = ,F10.5/)
147 FORMAT(/// 1X,10F10.3)
148 FORMAT(/// 25X,5HDIRECT OPTIMIZATION USING ANALYTICAL EXPRESSION F
1TOR CY)/)
730 149 FORMAT(/// 25X,3HUPTIMUM LOADING USING MUNKS CRITERION/)
150 FORMAT(/// 23X,4HLIFT COEFF CALCULATED FROM CLP AND SOLVED GAM=.
1E13.5/)
735 151 FORMAT(/// 22X,5HSFGMT.3X,10H BOUND CIRC,2X,8HGA/M/AVE,4X,10HSHED STR
1H,5X,7HGA/M/AVE,6X,1H/TA//)
152 FORMAT(/// 20X,15,5E13.5)
153 FORMAT(/// 25X,4THERALCALCULATEO USING SUB DRACAL AND LOADS FROM DIR
1ECT OPTIMIZATION = ,F15.5/)
740 154 FORMAT(/// 25X,6HCDCALCULATEO USING SUB DRACAL AND OPTIM LOADS US
1ING MUNK (KIT = ,F15.5/)
155 FORMAT(/// 25X,19HC FOR FLAT WING = ,E15.5//,25X,35HRATIO OF ZERO DIHED
1PAL DOWN=H/AV = ,E15.5/)
156 FORMAT(/// 25X,49HRINDUCED DRAG EFFICIENCY FOR WINGS OF EQUAL SPAN =
1E15.6/)
745 157 FORMAT(/// 25X,1AH TOL IN SNTAN = ,E15.5/)
158 FORMAT(/// 25X,2HAPPROX CD USING SOLVED ROUND/25X,4HCIRCULATIONS AN
1D WASHES AT SFF MIDPOINTS = ,E13.5/)
1ND
SURROUNTE GAMCAL

SUBROUTINE GAMCAL (I, IIL, SNN, AOPT, HGAMO)
C
CALCULATE BOUND CIRCULATION AT LEFT END PT OF SEGMENT I*HGAM
AOPT LAST COLUMN CONTAINS ARRAY OF OPTIMIZED SHEET SHEET STRENGTHS

COMMON /FEN/ NSPT(I10), NLIN
DIMENSION SNN(I1), AOPT(I1), SNN(I1)
HGAMO = 0,
DO 1 K=1, NLIN
KCHK = NSPT(KK)
1 IF (I, FJ, KCHK) GO TO 4
IF (I, FJ, KCHK) GO TO 2
CONTINUE
2 CONTINUE
KST = KCHK
HGAMO = AOPT(KST, IIL1) SNN(KST) * AOPT(I, IIL1) SNN(I-1)
KCHK = KCHK1
1 IF (I, FJ, KCHK1) GO TO 5
IM = I-1
1P = KCHK+1
DO 3 J=IP, IM
HGAMO = HGAMO * AOPT(J, IIL) * (SNN(J-1) SNN(J))
3 CONTINUE
GO TO 5
4 HGAMO = 0
5 CONTINUE
RETURN
END
SUBROUTINE WCAL

SUBROUTINE WCAL (I, J, NINT, YHH, ZHH, PPP, AAAA)

C
C
CALCULATE MATRIX COEFFICIENT FOR DRAG OPTIMIZATION USING MUNK CRIT
C
FINDS COEF OF ITH DOWNWASH DUE TO JTH SHEET STRENGTH
C
1. FINDS COEF MULTIPLYING STRENGTH J IN EQUATION 1
C
DIMENSION YHH(I), ZHH(I), PPP(I)
COMMON /SEG/ SNN(5)
COMMON /FEN/ NSPT(10), NLINE
INTERF P
PI = 4.*ATAN(1.)
AAAA = 0.
ICTRNL = 0
P = J
K = J
1 CONTINUE

CALL GCAL (PK, YHH, ZHH, PPP, SNN(K), AA, BH, DD, FF, GG, EE, AJ, AK, RR, TT, UU)
1(WK)
CALL GCAL (AA, BH, FF, GG, SNN(K), A, B, C, D, F, G, HJ, BK, RL, RM, BN, BO, BP, 1)
1)

IF (RPP, F0, 0.) GO TO 2
EXPR = 2.*ATAN2(C, AHS(RR)) - ATAN2(A, AHS(RR)) / (AHS(RR))
GO TO 3
2 CONTINUE

EXPR = 2.*U/P, C
3 CONTINUE

RLOG = ALOG(F/G)

A1PK = (A*EXP + 5*HR*LOG)/(2.*PI)
A2PK = (HL*EXP + 2.*HR*LOG)/(2.*PI1*SNK(K))

CALL GCAL (DI, FF, AJ, AK, SNN(K), A, B, C, D, F, G, HJ, BK, RL, RM, BN, BO, BP, 2)
1)

IF (UIII, E0, 0.) GO TO 4
EXPR = 2.*ATAN2(C, AHS(UIII)) - ATAN2(A, AHS(UIII)) / (AHS(UIII))
GO TO 5
35 CONTINUE

EXPR = (U/P, C
4 CONTINUE

RLOG = ALOG(F/G)

A2PK = -(A*EXP + 5*F*LOG)/(2.*PI)
A4PK = -(F + EXP + 5*E*LOG)/(2.*PI1*SNK(K))
IF (ICTRNL, F0, 7) GO TO 7
AAAA = (A1PK + A2PK)*5.0 - (A3PK + A4PK)*5.0
GO TO 1

PAGE 1
ICNTRL = 2
DO 6 L=1,NL LINE
JCHK = NSPT(L)
IF (K,F0,JCHK) GO TO 6
6 CONTINUE
K = J-1
GO TO 1
7 CONTINUE
PPAA = PPAA + 0.5*(A1PK+A2PK+A3PK+A4PK)
8 CONTINUE
RETURN
END
SUBROUTINE CCAL

C
C SUMMIT INF CCAL
C
C CALCULATES GEOMETRICAL CONSTANTS NEEDED IN EVALUATION OF INTEGRALS
C FOR VARYING I AND J VALUES
C
REAL JJ, KK
DIMENSION YHH(I), ZHH(I), PPP(I)

DYIJ = YHH(I) - YHH(J)
DZIJ = ZHH(I) - ZHH(J)
C0I = COS(PPP(I))
SII = SIN(PPP(I))
C0J = COS(PPP(J))
SJJ = SIN(PPP(J))
AA = DYIJ * C0I * DZIJ * SII
WW = -C0S(PPP(J) - PPP(I))
FF = -2 * (DYIJ * C0J * DZIJ * SJJ)

GG = DYIJ * DYIJ + DZIJ * DZIJ

DD = DYIJ * C0I * DZIJ * SII
EE = C0S(PPP(J) * PPP(I))

JJ = 2 * (DYIJ * C0J - DZIJ * SII)
KK = DYIJ * DYIJ + DZIJ * DZIJ
PR = 2 * (DYIJ * SII - DZIJ * C0J)
TT = 2 * SIN(PPP(J) - PPP(I))
IU = 2 * (DYIJ * SII * DZIJ * C0J)

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
E = S*S + S*FF + GG
F = S*S + S*FF + GG
J = 2.0*(AA - S*PP)
K = 2.0*(AA - S*PP)
L = 0.5*(RR*FF - AA*FF - 2.0*Hb*GG)
M = 0.5*(FF - 6.0*AA*HR + 4.0*FF*HR*BB)
N = 2.0*(HR*HR - 1.0)*RR
O = 0.5*(AA - FF*HR)
P = 0.5*(1.0 - 2.0*HR*HR)
RETURN
END
SUBROUTINE SNTAN

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

EVALUATES INTEGRALS OF THE FORM S*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 INTEGRAL APPROXIMATED AS A QUADRATIC IN S WHICH IS
FORCED THROUGH ATAN VALUES AT -S,0, AND S.
A,C ARE CALCULATED IN SUBROUTINE CONCAL
PR,RR,TT ARE CALCULATED IN SUBROUTINE CCAL
RESULTS ARE RTAN,RSTAN,RS2TAN,RS3TAN
APPROXIMATE INTEGRAL EVALUATED USING MACSYMA PROGRAM OF MIT PROJ.
EVALUATION OF INTEGRALS FOR TT=0. BEGIN AT AT LABEL 10
SINGULAR INTEGRALS EVALUATED AT APPROXIMATE ENDPOINTS, +/ - SAWAY
MIDRANGE SINGULARITIES EXCLUDED, ATAN PART OF INT APPROX-
IMATED AS 2 QUADRATICS

DIMENSION AA(3,3), AA(3), IPIVOT(3)
COMMON /TELL/ TOL,TOL2
RR = RP
SSS = S
CCC = C
RTAN = 0.
RSTAN = 0.
RS2TAN = 0.
RS3TAN = 0.
RS4TAN = 0.0
IF (APS(TT),LT,LE-05,A ND,ABS(RR),LT,LE-05) GO TO 7
IF (TT,EQ,0.0,AND,PR,xE,0.0) GO TO 7
IF (TT,EF,0.0) GO TO 4
IF (SZERO,LE,0.) GO TO 3
SZERO = -RR/IT
IF (APS(AHS(SZERO)-S),LT,LE-03,A ND,ABS(SZERO),LE,S) GO TO 3
IF (SZERO,LT,0.0,AND,SZERO,GT,-S) GO TO 16
IF (SZERO,GE,0.0,AND,SZERO,LT,5) GO TO 16
CONTINUE
SUBROUTINE SNTAN

C = ATAN2((CCC+ABS(RR)))
C1 = S*(ATAN2(CC+2,ABS(RR+TT*S))*ABS(RR+TT*S))+ATAN2(CC-2,ABS(RR)*ABS(RR+TT*S))
C2 = (ATAN2(CC+2,ABS(RR+ABS(RR+TT*S)))/S-C*C)
C = INTEGRAND NOW IS C1*S*S+C2*S*C*S*S*S*S/(RR+TT)*S)

CLOGR = ALOG(ABS((RR+TT*S)/(RR-TT*S))
CON0 = (C*TT*C-T*PP+TT*C1*RR)/(TT**3)
CON1 = (C2*TT-C1*RR)/(TT**4)
RTAN = CON0*CLOGR+S*S*CON1
CON2 = (C*TT*C-T*PP+TT*C1*RR+TT*C2*RR)/(TT**3)
CON3 = (C*TT*C-T*PP+TT*C1*RR+TT*C2*RR)/(TT**4)
RTSTAN = 2.0*S*CON2-CON3*CLOGR+2.0*C1*TT*TT*S/(3*TT**3)
CON4 = (C*TT*C-T*PP+TT*C1*RR+TT*C2*RR)/(TT**5)
CON5 = (C2*TT-C1*RR)/(TT**6)
RS3TAN = 2.0*CON5*S*S+S*CON9-S*CON10*CLOGR+2.0*C1*TT**4*S*S*S/(60*TT**5)

CON6 = (-C*PP+TT*C-T*PP+TT*C1*RR-RR**3)/(TT**4)
CON7 = CON4*CLOGR+2.0*S*S*CON6
CON8 = 20.0*(C*TT+C2*RR+TT*C1*RR+TT*C2*RR)/(60*TT**5)
CON9 = (C*TT+C2*RR+TT*C1*RR+TT*C2*RR)/(TT**5)
CON10 = CON10/(TT**6)
RS3TAN = 2.0*CON8*S*S+2.0*S*CON9-S*CON10*CLOGR+2.0*C1*TT**4*S*S*S/(60*TT**5)

CONH = (C-PP+TT*C-T*PP+TT*C1*RR+TT*C2*RR+TT*C3*RR)/(TT**4)
CONI = (C2*TT-C1*RR)/(TT**5)
RS4TAN = CONH*CLOGR+CONI*2.0*S+S*CON2+COND*S*S+S*S*S+S*S*S+S*S/S/(60*TT**5)

IF (RR+TT*S) = 0.0 GO TO 2
RTAN = -RTAN
RTSTAN = -RTSTAN
RS3TAN = -RS3TAN
RS4TAN = -RS4TAN

CONTINUE

GO TO 13

SAWAY = S-TOL
S = SAWAY
SUBROUTINE SNTAN

GO TO 1

C FOR CASE OF RR NOT ZERO, TT=0.0

C CONTINUE

PR = ABS(RR)
ALNUM = (2*PR)**2*8*S**2+4*C*B**2*S+RR**2+C**2
ALNED = (2*PR)**2*8*S**2-4*C*B**2*S+RR**2+C**2
IF (ALNUM.EQ.0.0 OR ALNED.EQ.0.0) GO TO 5
GO TO 6

CONTINUE

S = S-10L
ALNUM = (2*PR)**2*8*S**2+4*C*B**2*S+RR**2+C**2
ALNED = (2*PR)**2*8*S**2-4*C*B**2*S+RR**2+C**2

RATLN = ALOG(ALNUM/ALNED)

TNDIF = ATAN2((C+2.*BB*S)*RR)-ATAN2((C-2.*BB*S)*RR)
TNSIF = ATAN2((C+2.*BB*S)*RR)+ATAN2((C-2.*BB*S)*RR)
RTAN = -(2*PR/BB)*RATLN/(5*S*C*TNDIF/BR)
PTAN = RTAN/SR+5*TNSIF/RR
RSTAN = 0.5*(S+S*(KR*KH-C*C)/(2*BR)**2)*TNDIF-S*SR+5/SBR+(5*C*R)+1)
RSTAN = RSTAN/RR
RS2TAN = (S**2/(S.PR))**TNDIF=(3+C.RR+6-JO+C.PR/PR)/(+PR/PF/RR**3)
RS5TAN = (S**5/(S.PR**5))**TNDIF=(5*C**3+3*CC+3**3)/(32*BB**4)*RATLN-TNDIF
RS**2 = (S**2/(S.PR**2))**TNDIF=(6+3**3)/(64+BB**4)-RR**3/(12+BB)-S*(9+C**2+RR)

GO TO 13

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

C CONTINUE

TOP = C+2.*KK*C
BOT = C-2.*PR*S
IF (C+FQ.0.0 AND BK.EQ.0.0) GO TO 13
SBAN = -C/(2.*MM)
SB4DAR = ABS(4*SPAN)
IF (SB4DAR LT 5) GO TO 8
SUBROUTINE SMTAN

GO TO 9

A CONTINUE
WRITE (*,125)
SUL = SNAI*TOL
SLT = SNAI+TOL
CLOGR = ALUG(TOP/(C+2.*T*#SLL))
CLOGP = ALUG((C+2.*T*#SUL)/BOT)
CLOGT = CLOGR+1.25
RTAN = -(5.*PHR)+CLOGR
RSTAN = -(C**3/(R.*HH**3))*CLOGR+/(C/(4.*BB**2))*(2.*S-SLL+SUL)-(1.*
125/PHR)*(SUL+T-LL+SLL)
R33TAN = -(C**3/(16.*HH**4)) *CLOGR-(2.*S**3-SLL**3+SUL**3)/(6.*BR)-
1 (C**3/(R.*HH**3))*(2.*S-SLL+SUL)*(C/(R.*BB)*BB) + (SUL+T-LL+SLL)
R34TAN = -(C**3/(16.*HH**4)) *CLOGR*(SUL**4-SLL**4)/(8.*BB) + (C/(12.*
1*BR)*HH)**3)*(2.*S**3-SLL**3+SUL**3)*C**2*(SUL**2-SLL**2+SLL)/(16.*BB**3)
2*C**3*(2.*S-SLL+SUL)/(16.*BB**4)
GO TO 13

9 CONTINUE
IF (ARS(TOP),LT,1.0E-5,OR,ARS(BOT),LT,1.0E-5) GO TO 12
IF (TOP,LE,0.0) GO TO 11
IF (BOT,LE,0.0) GO TO 11
10 CLOGR = ALUG(TOP/BOT)

PTAN = -(1/HR)*CLOGR
RTAN = RTAN/2
PSTAN = +(25*C/HAR**2)*CLOGR-S/BH
R33TAN = -(C**3/(R.*HH**3)) *CLOGR-C**3/(2.*BB**2)
R34TAN = -(C**3/(16.*HH**4)) *CLOGR-S**3/(3.*BR)-S**2/(4.*BB**3)
16*B**4)
GO TO 13

11 TPD3BT = TOP/POT
IF (TPD3BT,GT,0.0) GO TO 10
IF (TOP,LE,0.0) GO TO 14
IF (BOT,LE,0.0) GO TO 14
12 CONTINUE
SAWAY = S*TOL
TOP = C+2.*BB*SAWAY
BOT = C-2.*BB*SAWAY
CLOGR = ALUG(TOP/BOT)
RTAN = -(1/HR)*CLOGR
RTAN = RTAN/2
RS1 = \((C/HR*BR**2)\) CLOGR-S/HH
RS2 = \(-((HR*CH**3))\) CLOGR+C/S/2+BR**2
RS3 = \((C**3/(1*HR**2**2))\) CLOGR=5**3/3+**3=-C**2/4+BR**3
RS4 = \((-C**2/(C**2*HR**2))\) CLOGR+2*H**2/17*BR**2+2*SC**3/1
16**4

GO TO 13

RS5 = \(-4\) (\(HR**2\)) CLOGR-S 2/3+\(HR**2\) SC**3/1

CONTINUE

WRITE STATEMENTS GO HERE IF NEEDED

GO TO 15

WRITE \((6,27)\)

FOR CASE OF PP,TT NOT ZERO, BUT WITH MIDRANGE SINGULARITY

WRITE \((6,27)\)

IF \((I.EQ.2)\) GO TO 17

IF \((I.EQ.2)\) GO TO 17

AA1(1,1) = SUL+SIU

AA1(1,2) = SUL

AA1(1,3) = AA1(2,3)=AA1(3,3)=1.

AA1(2,1) = SIMD2*SIMD2

AA1(2,2) = SIMD2

AA1(3,2) = SIMD2

AA1(3,3) = -S

AA1(1) = ANG5

AA(3) = ANG5

AA(3) = ANG3

CLOGR = ALOG(ABS((HR+TT*SIU)/(HR-**S))

PAGE 5
SUAROUTINE SNTAN

215
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 10

17 CONTINUE

220
AA1(1,1) = S*S
AA1(1,2) = S
AA1(1,3) = AA1(2,1)*AA1(3,1)=1*
AA1(2,1) = SMID1*SMID1
AA1(2,2) = SMID1
AA1(3,1) = SLL*SLL
AA1(3,2) = SLL

225
AA(1) = ANG1
AA(2) = ANG2
AA(3) = ANG3

230
CLOGP = ALOG(ABS((P*R*T*S)/(R*TT*SLL)))
SUSE = SMID1
DFLS = S-SLL
DFLS2 = S*S-SLL*SLL
DFLS3 = S**3-SLL**3
DFLS4 = S**4-SLL**4
DFLS5 = S**5-SLL**5
DFLS6 = S**6-SLL**6

10 CONTINUE

240
CALL SIMFO (AA1,3,AA4,DET,IPIVOT,3,ISCALE)
C1 = AA(1)
C2 = AA(2)
C = AA(3)
CON0 = (C**TT+T-C**2*RH*TT+C1*RR*RH)/(TT**3)
CON1 = (C**TT+C1*RR)/(TT**2)
CON2 = 0.5*C1/TT
CON3 = (C**TT-C2*RH*TT+C1*RR*KR)/(TT**3)
CON4 = (C**TT-C2*RH*TT+C1*RR*KR)/(TT**3)
CON5 = CON6 = CON7 = (C2**TT+C2*RH*TT+C1*RR*RH)/(TT**4)
CON31 = (C*TT**3-C2*RR*TT**2+C1*RR**2*TT)/(2*TT**4)  F 253
CON32 = C1/(4*TT)  F 254
CONN = 2*TT**4+C2*RR**2+2*TT**2+C1*RR**4)/(60*TT**5)  F 255
CONV = (C2*TT**2+2*TT**2+C1*RR**4)/(12*TT**5)  F 256
CON10 = C2*TT**3+C1*RR**2)/(TT**6)  F 257
CON10 = CON10/(TT**6)  F 258
CON41 = (C2*TT-C1*RR)/(4*TT**2)  F 259
CON42 = (-C1*RR**2+C2*RR**2*TT-C1*RR**3)/(12*TT**4)  F 260
CONA = (C3*TT**4+C2*RR**2*TT+C1*RR**3)/(TT**7)  F 261
CONA = (C2*TT-C1*RR)/(5*TT**2)  F 262
CONC = (C2*RR+TT+C1*RR**2*TT)/(3*TT**4)  F 263
CONC = (-C1*RR**2+C2*RR**2*TT-C1*RR**3)/(3*TT**4)  F 264
CONF = (C*TT**4-C2*RR*TT**2+C1*RR**2*TT)/(2*TT**4)  F 265
CONF = C1/(4*TT)  F 266
CONG = CONG*AP**7*?/(TT**2)  F 267
IF (I.EQ.1) GO TO 19  F 268
IF (I.EQ.1,AND. (RH*TT*SUSE) .GT.0.1) GO TO 19  F 269
RTAN = RTAN-CON0*CLnRH*CON1*DELS*CON11*DELS2  F 270
RSTAN = RSTAN-1*DELS2*CON2*DELS3*CLnGR+C1*DELS3/(3*TT)  F 271
RS2TAN = RS2TAN-CON4*CLnGR*CON5*DELS5*CON3*DELS3*CON31*DELS2*CON32*DELS4*CON6  F 272
RSTAN = RSTAN*CON4*DELS4*CON8*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 273
RSTAN = RSTAN*CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 274
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 275
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 276
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 277
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 278
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 279
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 280
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 281
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 282
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 283
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 284
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 285
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 286
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 287
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 288
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 289
RSTAN = RSTAN+CON4*CLnGR*CON5*DELS5*CON9*DELS6*CON11*DELS2*CON10*DELS  F 290
RSTAN = -RTAN  F 291
RSTAN = -RSTAN  F 292
RSTAN = -RSTAN  F 293
RSTAN = -RSTAN  F 294
SUBROUTINE SNTAN

295 21 CONTINUE
22 CONTINUE
23 CONTINUE
24 CONTINUE
25 GO TO 13

300 RR = RRR
S = SSS
C = CCC
RETURN

305 25 FORMAT(30X,11H200 ENTERED)
26 FORMAT(30X,3HONE OF THE ENDPOINTS HAS A NEGATIVE LOG ARG)
27 FORMAT(30X,10H00 ENTERED)
END
SUBROUTINE LOGS (S,E,D,RELN,RESLN,RES2LN,RES3LN)

C

SUBROUTINE LOGS

C

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

C

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

C

REAL LATB,LAMB,L1,L2,L3,L4
COMMON /TELL/ TOL,TOL2
RELN = 0.
RESLN = 0.
RES2LN = 0.
RES3LN = 0.
S5 = 5
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,0.0000000001) GO TO 6
IF (BB,LE,0.0000000001) GO TO 6

1 DISC = E**F**A**D
DISQ = SORT(ABS(DISC))
DIS = E**F-2.8D
DIS3 = E**3-3.8D**3
DIS4 = (E**F-4.8D)**(E**F-8D)
DIS44 = E**4-4.8D**4-E**2+8D**2
DIS5 = E**5-6.8D**5-E**3+18D**3-12D**2
LATB = ALOG(A**B)
LADH = ALOG(A/B)

40 IF (AA,LE,0.0000000001) S = SAWAY
IF (BB,LE,0.0000000001) S = SAWAY
RE = S*LATB+0.5*E*LADH

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

45 ESP = F*2*5
ESM = F-25
IF (DISC) A=3*2
2 CONTINUE

L1 = F-DIS*2*5
L2 = E+DIS*2*5
L3 = F-DIS*2*5
L4 = F+DIS*2*5
50
DIFFLN = RCLG(L1*L4/L2/L3)
RLN = 0.5*LADR*S**2+(0.25*E*DISC/DISQ)*DIFFLN
RESLN = RESLN-0.25*LARD*E*S
60
CLOGRT = ALGO(ESP/ESM)
RELN = S*LATR-4*S*F*CLOGRT
RESLN = 0.5*S**2*LARD-0.5*DIS*CLOGRT+E*S
70
RES2LN = (S**3/3)*LATB-DIS4/(6*DISQ)*DIFFLN*LADD*(DIS3/6)-4*S**3/9
1-6*DIS5/9
RES3LN = 0.5*S**4*LADD+DIS5/(8*DISQ)*DIFFLN-LADD*(DIS4/8)+E*S**3
1/6+0.5*S*DIS
75
GO TO 5
6 CONTINUE

SPHAY = S-TOL?
A = SA2W*SAWY+E*SAWY+D
B = SA2W*SAWY-F*SAWY+D
80
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

A5
S = S5
RETURN
END

G 85
G 86
G 87
SUBROUTINE DRACAL

1 SUBROUTINE DRACAL (I,J,YHH,ZHH,PPP,AINT)

2 CALL DRACAL

3 SUBROUTINE DRACAL

4 TREFFT PLAN L DRAG ANALYSIS ASSUMES PIECEWISE LINEARLY VARYING
5 SHEET VORTICITY SHEET STRENGTH

6 CALCULATE INTEGRALS A THROUGH F FOR DRAG COEF. CALCULATION

7 CALLS SUBROUTINES LOGS, SNTAN, CCAL, CONCAL

8 DIMENSION AINT(6)

9 DIMENSION YHH(1), ZHH(1), PPP(1)

10 COMMON /SEG/ SNN(51)

11 PI = 4.*APTNFL.1

12 S = SIN(J)

13 CALL CCAL (I,J,YHH,LHH,PPP,S,AA,HR,DD,FF,GG,EF,AH,AK,RR,TT,WW)

14 CALL CCAL (AA,HH,FF,GG,S,AB,CB,DD,GF,CJ,CK,CL,CM,CN,CO,CP)

15 PI = 4.*ATAN(1.)

16 S = SNN(J)

17 CALL LOGS (S,CJ,F,REL,RES2L,RES3L,RES4L)

18 CALL SNTAN (S,CJ,F,RFF,RS2T,RS3T,RS4T)

19 AAAAA = A*RTAN(0)*RS2TAN*RELN/4

20 BRRRR = 2.*(CL*RTAN+CM*RS2TAN+CN*RS3TAN)*CO*RELN+CP*RESL

21 CCCCC = A*RS2TAN+R*RS2TAN+R*RESL/4

22 DDDDD = 2.*(CL*RTAN+CM*RS2TAN+CM*RS3TAN)*CO*RES2L+CP*RESL

23 EEEFF = 2.*(CL*RS2TAN+CM*RS3TAN)*CO*RES2L+CP*RES3L

24 FFFFF = 2.*(CL*RS2TAN+CM*RS3TAN)*CO*RES2L+CP*RES3L

25 CALL LOGS (S,CJ,F,REL,RES2L,RES3L,RES4L)

26 CALL SNTAN (S,CJ,F,RFF,RS2T,RS3T,RS4T)

27 AAAAA = A*RTAN+R*RTAN+R*RELN/4

28 BRRRR = 2.*(CL*RTAN+CM*RS2TAN+CN*RS3TAN)*CO*RELN+CP*RESL

29 CCCCC = A*RTAN+R*RTAN+R*RELN/4

30 DDDDD = 2.*(CL*RTAN+CM*RS2TAN+CM*RS3TAN)*CO*RES2L+CP*RES3L

31 EEEFF = 2.*(CL*RS2TAN+CM*RS3TAN+CM*RS4TAN)*CO*RES2L+CP*RES3L

32 FFFFF = 2.*(CL*RS2TAN+CM*RS3TAN+CM*RS4TAN)*CO*RES2L+CP*RES3L

33 S = SNN(J)

34 CALL SNTAN (S,CJ,F,RFF,RS2T,RS3T,RS4T)

35 AAAAA = A*RTAN+R*RTAN+R*RELN/4

36 BRRRR = 2.*(CL*RTAN+CM*RS2TAN+CN*RS3TAN)*CO*RELN+CP*RESL

37 CCCCC = A*RS2TAN+R*RS2TAN+R*RESL/4

38 DDDDD = 2.*(CL*RTAN+CM*RS2TAN+CM*RS3TAN)*CO*RES2L+CP*RESL

39 EEEFF = 2.*(CL*RS2TAN+CM*RS3TAN+CM*RS4TAN)*CO*RES2L+CP*RES3L

40 FFFFF = 2.*(CL*RS2TAN+CM*RS3TAN+CM*RS4TAN)*CO*RES2L+CP*RES3L

41 S = SNN(J)

42 CALL SNTAN (S,CJ,F,RFF,RS2T,RS3T,RS4T)
SIJRROUTINE DPACAL PAGE 2

AAANNN = AAAAAA*A*TAN+R*STAN=LE*RELH/4 H 43
BBBDBB = BBBBBB-2.* (CL*RTAN+CM*RS2TAN+CN*RS2TAN)-CO*RELN-CP*RESLN H 44
CCCCCC = CCCCCC*A*STAN+R*STAN-EE*RESLN/4 H 45
DDDDDD = DDDDD-2.* (CL*RTAN+CM*RS2TAN+CN*RS3TAN)-CO*RESLN-CP*RES2 H 46
EEFFFF = EEEEEE-A*RS3TAN+R*RS3TAN-EE*RES2LN/4 H 47
FFFFFF = FFFFFF-2.* (CL*RS2TAN+CM*RS3TAN+CN*RS4TAN)-CO*RES2LN-CP*RE H 48

CALL LOGS (S,CK,G,RHIN,RESLN,RES2LN,RES3LN) H 51
CALL SNTAN (S,EE,HH+MW,RTAN,RS2TAN,RS3TAN,RS4TAN) H 52
AAANNN = AAAAAA*A*TAN+H*RS2TAN+EE*RELN/4 H 53
PPPPPP = PPPPPP+2.* (CL*RTAN+CM*RS2TAN)+CO*RELN+CP*RESLN H 54
CCCCCC = CCCCCC*ARSTAN+H*RS2TAN+EE*RESLN/4 H 55
DDDDDD = DDDDD+2.* (CL*RS2TAN+CM*RS3TAN+CN*RS3TAN)-CO*RES2LN+CP*RES2 H 56

FEFEFE = FEFEFE-A*RS3TAN+R*RS3TAN+EE*RES2LN/4 H 57
FFFFFF = FFFFFF+2.* (CL*RS2TAN+CM*RS3TAN+CN*RS4TAN)-CO*RES2LN+CP*RE H 58

SK = SNN(J) H 61
AAANNN = AAAAAA/P1 H 62
PPPPPP = PPPPPP+2.*P1*SK)+(2.*S/P1)*(BB-EE) H 63
CCCCCC = CCCCCC/P1 H 64
DDDDDD = DDDDD+2.*P1*SK) H 65
FEFEFE = FEFEFE+2.*P1*S) H 66
FFFFFF = FFFFFF+4.*P1*S*SK)+(BB-EE)*S*S/(3.*P1) H 67

AINT(1) = AAAAA H 68
AINT(2) = BBBBBB H 69
AINT(3) = CCCCCC H 70
AINT(4) = DDDDD H 71
AINT(5) = FFFFF H 72
AINT(6) = FFFFF H 74
RETURN H 74

END H 75
**SUBROUTINE SINEQ**

**SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS**

**DIMENSION IPIVOT(N), A(NMAX,N), B(NMAX,M)**

**EQUIVALENCE (IROW,JROW), (ICOLUMN,JCOLUMN), (AMAX,T,SWAP)**

**INITIALIZATION**

1. **ISCALE = 0**
2. **R1 = 10.0**
3. **R2 = 1.0/N**
4. **DETERM = 1.0**
5. **DO J = 1, N**
6. **DO 3 I = 1, N**
7. **IF (DETERM) 3 7, 3**
8. **IF (DETERM) 4, 6, 39**
9. **IF (DETERM) 5, 6, 6**
10. **ICOLU = J**
11. **AMAX = A(J,K)**
12. **CONTINUE**
13. **IF (AMAX) 9, 8, 9**
14. **IF (ISCALE) 10, 14, 10**
15. **DETERM = -DETERM**
16. **GO TO 39**
17. **SWAP = A(IROW,L)**
SURROUNTE SIMEQ

A(IROW,L) = A(ICOLUMN,L)  
11 A(ICOLUMN,L) = SWAP  
45 IF (M) 14,14,12  
12 DO 13 L=1,M  
13 SWAP = $U$(IROW,L)  
14 A(IROW,L) = A(ICOLUMN,L)  
50 A(ICOLUMN,L) = SWAP  
14 PIVOT = A(ICOLUMN,ICOLUMN)  
C IF (PIVOT) 15,15,15  
C SCALE THE DETERMINANT  
C  
15 PIVOT = PIVOT  
16 DETERM = DETERM/R1  
17 DETERM = DETERM/R1  
60 ISCALE = ISCALE+1  
GO TO P1  
18 IF (ARS(DETERM)-R1) 19,19,19  
19 DETERM = DETERM/R1  
20 DETERM = DETERM/R1  
65 ISCALE = ISCALE-1  
21 IF (ARS(DETERM)-R2) 22,22,22  
22 PIVOT = PIVOT/R1  
23 PIVOT = PIVOT/R1  
70 ISCALE = ISCALE+1  
24 IF (ARS(PIVOT)-R1) 25,25,25  
25 ISCALE = ISCALE-1  
26 PIVOT = PIVOT/R1  
27 DETERM = DETERMR1  
C C DIVIDE PIVOT ROW BY PIVOT ELEMENT  
C

Page 2
SUBROUTINE SIHFQ

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

IF (M) 32,32,30
30 DO 31 L=1,N
31 R(ICOLU,L) = R(ICOLU,L)/PIVOT
C
C REDUCE NON-PIVOT ROWS
C
32 DO 38 L=1,N
  IF (L-ICOLU) 33,38,33
33 T = A(L,ICOLU)
34 A(L+L) = A(L,L)-A(ICOLU,L)*T
35 CONTINUE
36 DO 37 L=1,N
  IF (M) 38,38,39
37 B(L+L) = B(L,L)-B(ICOLU,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.
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.