NASA Contractor Report 3458

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

NASA

NASA Contractor Report 3458

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

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_B 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)]
$A_{1_{ij}}$, $A_{2_{ij}}$, $A_{3_{ij}}$, $A_{4_{ij}}$	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
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
$G_i, \bar{G}_i, \hat{G}_i$	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)
$I_{1_{i,j}}$, $I_{2_{i,j}}$, $I_{3_{i,j}}$, $I_{4_{i,j}}$, $I_{5_{ij}}$, $I_{6_{i,j}}$	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
ℓ	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_{ref}	reference wing area
$\mathcal{s}$	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 $\mathcal{s} = \mathcal{s}_i$ on panel i due to panel j and its image [eq. (2)]
w_o	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
γ	wake trailing vortex sheet strength
Γ	bound circulation
$\overline{\Gamma}$	average bound circulation
Γ_o	bound circulation at outboard edge of wake panel [eq. (4)]
η	nondimensional spanwise coordinate
λ	Lagrange multiplier in equations (10) and (12)
σ_{ij}	angle between y-axis and orientation of h_{ij} (fig. 1)
ϕ	dihedral angle

Subscripts:

i	influenced point
j	influencing point
n	normal component
o	value at $s = -s$

Superscripts:

'	image panel quantity
—	average value

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

$$\gamma(s_j) = \frac{\gamma_{j+1} + \gamma_j}{2} + \frac{s_j}{s_j} \cdot \frac{\gamma_{j+1} - \gamma_j}{2} \tag{1}$$

The γ_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 s_i on wake segment i, due to wake segment j and its image, may be written, using the Biot-Savart Law, as

$$w_{n,j} = \frac{\gamma_{j+1} + \gamma_j}{2} \frac{1}{2\pi} \left\{ \int_{-s_j}^{s_j} \left(\frac{R_{ij}}{h_{ij}^2} - \frac{R'_{ij}}{[h'_{ij}]^2} \right) ds_j \right\}$$

$$+ \frac{\gamma_{j+1} - \gamma_j}{2} \frac{1}{2\pi} \left\{ \int_{-s_j}^{s_j} \left(\frac{s_j R_{ij}}{h_{ij}^2} - \frac{s_j R'_{ij}}{[h'_{ij}]^2} \right) ds_j \right\} \tag{2}$$

The R_{ij} and h_{ij} are distances as shown in figure 1. This expression is evaluated (ref. 4) as

$$w_{n,j} = \frac{\gamma_{j+1} + \gamma_j}{2}\left(A_{1_{ij}} + A_{2_{ij}}\right) + \frac{\gamma_{j+1} - \gamma_j}{2}\left(A_{3_{ij}} + A_{4_{ij}}\right) \tag{3}$$

where the $A_{1_{ij}}$, etc., are determined by the wake geometry. Next, the γ distribution is integrated spanwise beginning at the tip of the current planform to obtain the bound circulation as

$$\Gamma(s_i) = \Gamma_o\left(-s_i\right) + \frac{s_i}{4}\left(\gamma_{i+1} + \gamma_i\right) + \left(\frac{\gamma_{i+1} + \gamma_i}{2}\right)s_i$$
$$+ \left(\frac{\gamma_{i+1} - \gamma_i}{2}\right)\frac{s_i^2}{2s_i} \tag{4}$$

where $\Gamma_o\left(-s_i\right)$ is the value of Γ at $s_i = -s_i$, which is a known linear function of the γ values.

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

$$C_{D,ij} = \frac{1}{S_{ref}} \int_{-s_i}^{+s_i} \frac{\Gamma(s_i)}{U}\left(\frac{w_{n,j}(s_i)}{U}\right)ds_i \tag{5}$$

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 γ_j values:

$$C_{D,ij} = \frac{1}{S_{ref}}\left(G_i\bar{G}_j I_{1_{i,j}} + G_i\hat{G}_j I_{2_{i,j}} + \bar{G}_i\bar{G}_j I_{3_{i,j}} + \bar{G}_i\hat{G}_j I_{4_{i,j}}\right.$$
$$\left. + \hat{G}_i\bar{G}_j I_{5_{i,j}} + \hat{G}_i\hat{G}_j I_{6_{i,j}}\right) \tag{6}$$

where the G_i, $\bar{G}_i$, $\hat{G}_i$, etc., are linear functions of the γ_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 $A_{1_{ij}}$ through $A_{4_{ij}}$ times $s_i^{\,n}$, for $n = 0, 1, 2$, as given in reference 4.

The Munk optimization procedure uses (ref. 6)

$$\frac{w_n}{\cos\phi} = w_o = \text{constant} \tag{7}$$

which for the assumed wake model is written as

$$\cos\phi_i = \frac{1}{w_o}\sum_{j=1}^{N}\left\{\left(\frac{\gamma_{j+1}+\gamma_j}{2}\right)\left(A_{1_{ij}} + A_{2_{ij}}\right) + \left(\frac{\gamma_{j+1}-\gamma_j}{2}\right)\left(A_{3_{ij}} + A_{4_{ij}}\right)\right\} \tag{8}$$

This yields N equations for N + 1 unknowns, the N γ_j values and w_o. The system is completed by specifying a C_L value, since (ref. 4)

$$C_L = \frac{8}{S_{ref}}\left\{\frac{1}{3}\sum_{j=1}^{N}\left(\cos\phi_j s_j^2\left(\frac{\gamma_{j+1}}{w_o} + \frac{2\gamma_j}{w_o}\right)\right) + \sum_{j=1}^{N}\left(\cos\phi_j s_j \frac{\Gamma_o(s_j)}{w_o}\right)\right\}\frac{w_o}{U} \tag{9}$$

The direct optimization procedure extremizes (minimizes) the function

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

where λ is a Lagrange multiplier and $C_{L,j}$ is an analytical expression for the derivative of equation (9) above with respect to (γ_j/U). Similarly, expressions for derivatives of C_D with respect to (γ_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

$$C_D = \left(\frac{\gamma}{U}\right)^T A\left(\frac{\gamma}{U}\right) \tag{11}$$

The optimal γ_j/U values are then determined by N+1 equations given by

$$\sum_{j=1}^{N} \left(A_{ij} + A_{ji}\right)\frac{\gamma_j}{U} + \lambda\, C_{L,i} = 0, \; i = 1, \ldots, N \qquad (12)$$

and

$$\sum_{j=1}^{N} C_{L,j}\frac{\gamma_j}{U} - C_L = 0 \qquad (13)$$

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 γ 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 ℓ 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 ϕ = 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 ϕ = 30°. For minimum drag, both wings of the diamond shaped wake carry the same lift and have the same Γ 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, DSIN, 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 ϕ may be approached ($\phi \cong 89.5°$) 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

$$\int_{-s}^{s} \frac{s^n}{|R+Ts|} \tan^{-1}\left\{\frac{c+2Bs}{|R+Ts|}\right\} ds$$

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

$$\int_{-s}^{s} \frac{s^n}{|R+Ts|} \left\{ \tan^{-1}\left[\frac{c+2Bs}{|R+Ts|}\right] - \tan^{-1}\left[\frac{d+2Bs}{|R+Ts|}\right]\right\} ds$$

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

$$-\!\!\!\!\!\!\int_{-s}^{s} \frac{s^n}{c+2Bs} ds$$

However, there is currently no logic in the code to automatically replace the original integrand with the simpler, finite part in the viscinity 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 γ and Γ 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:

Card Number	Variable	Columns	Description
1	NLLINE	(1-5)	Total number of lifting surfaces for current configuration; NLLINE $\leq$ 10.

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

Card Number	Variable	Columns	Description
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

Card Number	Variable	Columns	Description
			wake semispan. Cards 3 to 5 described below are not needed when ICNTRL = 6.
			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.
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 ICNTL = 7 or 8, cards 3 to 5 must be specified as follows:

Card Number	Variable	Columns	Description
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, ..., (NBRK-1)	(1-25)	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:

Card Number	Variable	Columns	Description
5	YY(I), Z(I), PP(I), I = 1, ..., NBRK	(1-60)	Values of the Y, Z (in appropriate units), and ϕ (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.

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

6	CLDES	(1-10)	Desired lift coefficient, in F10.0 format.
6	SREF	(11-20)	Total configuration reference area (in appropriate units), in F10.0 format.
7	-	(5)	zero (0)
7	-	(10)	zero (0)

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:

8	-	(5)	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, ϕ, 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 $\leq$ 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 $(\text{BSAVE})^2/\text{SREF}$.

Solution Data

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

I	Individual wake panel number, numbered tip-to-root, as described above, for each lifting surface of the configuration.
BGAM(I)	Bound circulation value, Γ/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, $\Gamma/\bar{\Gamma}$, 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 Γ value at the root of the planform.
AOPT(I, NTOTT + 1)	Wake vortex sheet strengths for minimum induced drag, γ/U, at the outboard corner of wake panel I.
GAM(I)	Nondimensional wake vortex sheet strength for minimum induced drag, $\gamma/\bar{\gamma}$, 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_o, 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 Γ and γ 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

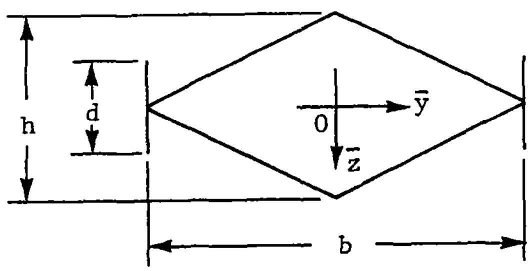
Sample input data and output data are presented for one of the configurations of figure 6 of this report, were $d/h = 1.0$, $h/b = 0.355$. Input data and a sketch of the input wake shape appear on page 22, while the output data begins on page 23.

Input Data and Sketch of Wake for

Diamond Wing with End Plates;

d/h = 1.0,   h/b = 0.355

```
      2
      8    15
      3
     10     5
0.            -0.1775    0.0        -0.498451 0.0          -19.600981
-.5           -.17750   89.5
      8    15
      3
     10     5
0.             0.1775    0.0        -0.498451 0.0           19.600981
-.5            .17750  -89.5
0.5           1.0
      0     0
      0
      0
```



Output Data for Diamond Wing with
End Plates; d/h = 1.0, h/b = 0.355

GENERAL INPUT GEOMETRY

0.00000	-.17750	0.00000	
-.49845	0.00000	-19.60098	
-.50000	-.17750	89.50000	
TOTAL PLANFORM PERIPHERAL LENGTH=			.70662
0.00000	.17750	0.00000	
-.49845	0.00000	19.60098	
-.50000	.17750	-89.50000	
TOTAL PLANFORM PERIPHERAL LENGTH=			.70662

COSINE SEGMENT SPACING

SEGMT NO	Y	Z	PHI
1	-.499926	-.169025	1.562070
2	-.499658	-.138363	1.562070
3	-.499225	-.088750	1.562070
4	-.498793	-.039137	1.562070
5	-.498525	-.008475	1.562070
6	-.495383	-.001093	-.342102
7	-.483185	-.005436	-.342102
8	-.459089	-.014017	-.342102
9	-.423689	-.026623	-.342102
10	-.377857	-.042944	-.342102
11	-.322720	-.062578	-.342102
12	-.259637	-.085042	-.342102
13	-.190161	-.109783	-.342102
14	-.116002	-.136191	-.342102
15	-.038987	-.163616	-.342102
16	-.499926	.169025	-1.562070
17	-.499658	.138363	-1.562070
18	-.499225	.088750	-1.562070
19	-.498793	.039137	-1.562070
20	-.498525	.008475	-1.562070
21	-.495383	.001093	.342102
22	-.483185	.005436	.342102
23	-.459089	.014017	.342102
24	-.423689	.026623	.342102
25	-.377857	.042944	.342102
26	-.322720	.062578	.342102
27	-.259637	.085042	.342102
28	-.190161	.109783	.342102
29	-.116002	.136191	.342102

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

30	-.038987	.163616	.342102

I	SNN(I)
1	.00848
2	.02219
3	.02743
4	.02219
5	.00848
6	.00326
7	.00969
8	.01589
9	.02169
10	.02696
11	.03157
12	.03540
13	.03835
14	.04037
15	.04139
16	.00848
17	.02219
18	.02743
19	.02219
20	.00848
21	.00326
22	.00969
23	.01589
24	.02169
25	.02696
26	.03157
27	.03540
28	.03835
29	.04037
30	.04139

TOL IN SNTAN = .26057E-05

DESIGN LIFT COEF. = .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)

OPTIMUM LOADING USING MUNKS CRITERION

SEGMT	BOUND CIRC	BGAM/AVE	SHED STRH.	GAM/AVE	ETA
1	0.	0.	.30907E+01	.31321E+02	.10000E+01
2	.33090E-01	.29612E+00	.81359E+00	.82448E+01	.99970E+00
3	.62488E-01	.55920E+00	.51136E+00	.51820E+01	.99893E+00
4	.87560E-01	.78357E+00	.40281E+00	.40820E+01	.99797E+00
5	.10456E+00	.93567E+00	.36320E+00	.36806E+01	.99720E+00
6	.10955E+00	.98035E+00	.22603E+00	.22905E+01	.99690E+00
7	.11040E+00	.98795E+00	.34701E-01	.35165E+00	.98463E+00
8	.11089E+00	.99232E+00	.15637E-01	.15847E+00	.94811E+00
9	.11156E+00	.99831E+00	.26488E-01	.26842E+00	.88825E+00
10	.11322E+00	.10132E+01	.50322E-01	.50996E+00	.80651E+00
11	.11658E+00	.10432E+01	.74120E-01	.75113E+00	.70492E+00
12	.12160E+00	.10882E+01	.84958E-01	.86096E+00	.58596E+00
13	.12748E+00	.11408E+01	.81092E-01	.82178E+00	.45258E+00
14	.13314E+00	.11915E+01	.66558E-01	.67449E+00	.30806E+00
15	.13762E+00	.12315E+01	.44410E-01	.45004E+00	.15595E+00
16	0.	0.	.30907E+01	.31321E+02	.10000E+01
17	.33090E-01	.29612E+00	.81359E+00	.82448E+01	.99970E+00
18	.62488E-01	.55920E+00	.51136E+00	.51820E+01	.99893E+00
19	.87560E-01	.78357E+00	.40281E+00	.40820E+01	.99797E+00
20	.10456E+00	.93567E+00	.36320E+00	.36806E+01	.99720E+00
21	.10955E+00	.98035E+00	.22603E+00	.22905E+01	.99690E+00
22	.11040E+00	.98795E+00	.34701E-01	.35165E+00	.98463E+00
23	.11089E+00	.99232E+00	.15637E-01	.15847E+00	.94811E+00
24	.11156E+00	.99831E+00	.26488E-01	.26842E+00	.88825E+00
25	.11322E+00	.10132E+01	.50322E-01	.50996E+00	.80651E+00
26	.11658E+00	.10432E+01	.74120E-01	.75113E+00	.70492E+00
27	.12160E+00	.10882E+01	.84958E-01	.86096E+00	.58596E+00
28	.12748E+00	.11408E+01	.81092E-01	.82178E+00	.45258E+00
29	.13314E+00	.11915E+01	.66558E-01	.67449E+00	.30806E+00
30	.13762E+00	.12315E+01	.44410E-01	.45004E+00	.15595E+00
31	.13946E+00	.12480E+01	0.	0.	0.

200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERFD
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)

```
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
200 ENTERED
```

```
CD CALCULATED USING SUB DRACAL AND OPTIM LOADS USING MUNK CRIT =
                                                                  .47846E-01
CD FOR FLAT WING =       .79577E-01
RATIO OF ZERO DIHEDRAL DOWNWSH/U =       .38317E+00


INDUCED DRAG EFFICIENCY FOR WINGS OF EQUAL SPAN =     .166319E+01
```

I	DOWNWASH	W/COS(PHI)
1	.83594E-03	.95793E-01
2	.83594E-03	.95793E-01
3	.83594E-03	.95793E-01
4	.83594E-03	.95793E-01
5	.83594E-03	.95793E-01
6	.90242E-01	.95793E-01
7	.90242E-01	.95793E-01
8	.90242E-01	.95793E-01
9	.90242E-01	.95793E-01
10	.90242E-01	.95793E-01
11	.90242E-01	.95793E-01
12	.90242E-01	.95793E-01
13	.90242E-01	.95793E-01
14	.90242E-01	.95793E-01

Output Data for Diamond Wing with End Plates;
d/h = 1.0, h/b = 0.355 (concluded)

15	.90242E-01	.95793E-01
16	.83594E-03	.95793E-01
17	.83594E-03	.95793E-01
18	.83594E-03	.95793E-01
19	.83594E-03	.95793E-01
20	.83594E-03	.95793E-01
21	.90242E-01	.95793E-01
22	.90242E-01	.95793E-01
23	.90242E-01	.95793E-01
24	.90242E-01	.95793E-01
25	.90242E-01	.95793E-01
26	.90242E-01	.95793E-01
27	.90242E-01	.95793E-01
28	.90242E-01	.95793E-01
29	.90242E-01	.95793E-01
30	.90242E-01	.95793E-01

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:

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

```
      PROGRAM  DNWASH(INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT)
      DIMENSION  GAM(51), HGAM(51), YHH(51), ZHH(51), PPP(51)
      DIMENSION  DNWSH(51)
      DIMENSION  AOPT(52,52)
      DIMENSION  AINT(6)
      DIMENSION  CDRAG(51)
      DIMENSION  ASIM(51,51), BSIM(51,1), IPIVOT(51)
      DIMENSION  YY(10), Z(10), PP(10), LSEG(10), DTHETE(10)
      DIMENSION  PERIF(10), B(10), GSUM(10)
      DIMENSION  CLP(53), SEQ(10), DTO(10)
      COMMON  /TELL/ TOL,TOL2,TOL3
      COMMON  /SEG/ SNN(51)
      COMMON  /DIROPT/ T1(53,53),T2(53,53),T3(53,53),T4(53,53),T5(53,53)
     1,T6(53,53)
      COMMON  /FEN/ NSPT(10),NLLINE
      PI = 4.*ATAN(1.)
      DTR = PI/180.
C     PROGRAM WRITTEN TO IMPLEMENT TREFFTZ PLANE INDUCED DRAG
C     OPTIMIZATION THEORY DESCRIBED IN NASA CR-3154, JUNE 1979.
C
C     PROGRAM WRITTEN BY DR. JOHN M. KUHLMAN, DEPT MECHANICAL
C     ENGINEERING AND MECHANICS, OLD DOMINION UNIVERSITY, NOR-
C     FOLK, VA 23508, UNDER NASA LANGLEY GRANT NSG-1357.
C
C     THEORY ASSUMES A TWO DIMENSIONAL ADVANCED PANEL MODEL OF
C     THE UNDISTORTED, INTERACTING WAKES OF MULTIPLE LIFTING
C     SURFACES.
C
C     WAKE VORTEX SHEET STRENGTHS ARE ASSUMED TO VARY IN A
C     PIECEWISE LINEAR FASHION.  ANALYTICAL EXPRESSIONS FOR
C     INDUCED NORMAL VELOCITY, BOUND CIRCULATION, INDUCED DRAG,
C     AND LIFT ARE DEVELOPED IN CR-3154 IN TERMS OF THE ASSUMED
C     WAKE MODEL.
C
C     THESE EXPRESSIONS ARE USED TO OBTAIN MINIMUM DRAG WAKE VORTEX
C     SHEET STRENGTHS, BOUND CIRCULATION DISTRIBUTIONS, AND CD
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
C
C     ICTRL=3 FOR GENERAL INPUT OF NON-PLANAR WING CASES
```

```
      C     THEN MUST INPUT TRIPLES OF YHH,ZHH,PPP FOR EACH CORNER POINT
      C     OF EACH SEGMENT, STARTING AT WING TIP
      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     NTOT=NUMBER OF SEGMENTS ON SEMISPAN, BOTH FOR EQUAL AND COSINE SPA
      C
      C     READ AN INTEGER VARIABLE NLLINE =BENT LIFTING LINES NEEDED TO MAKE
      C     NTOTT = NUMBER OF SEGMENTS ON NLLINE
      C     NS = START NUMBER ON EACH NLLINE
      C
          1 READ (5,131) NLLINE
            IF (NLLINE.EQ.0)  GO TO 129
            NTOTT = 0
            LSTART = 1
            DTOB = 0.0
            WRITE (6,130)
            WRITE (6,135)
          2 CONTINUE
            READ (5,131) ICNTRL,NTOT
            IF (ICNTRL.EQ.0)  GO TO 1
            NS = NTOTT+1
            NSPT(LSTART) = NS
            NTOTT = NTOTT+NTOT
            ISTOP = 1
            IDRAG = 1
      C
            IF (ICNTRL.EQ.3)  GO TO 12
            IF (ICNTRL.EQ.6)  GO TO 15
            IF (ICNTRL.EQ.7)  GO TO 20
            IF (ICNTRL.EQ.8)  GO TO 20
      C
      C
      C     CALCULATE DOWNWASH AT SEGMENT MIDPOINTS
      C
          3 CONTINUE
            WRITE (6,133)
            DO 11 I=1,NTOTT
            WDOWN = 0.
            DO 10 J=1,NTOTT
            S = SNN(J)
```

```
      CALL CCAL (I,J,YHH,ZHH,PPP,S,AA,BB,DD,FF,GG,EE,AJ,AK,RR,TT,UU,WW)
      CALL CONCAL (AA,BB,FF,GG,S,A,B,C,D,F,G,CJ,CK,CL,CM,CN,CO,CP,1)
      IF (RR.EQ.0.)  GO TO 4
      P = 2.*(ATAN(C/ABS(RR))-ATAN(D/ABS(RR)))/(ABS(RR))
      GO TO 5
    4 CONTINUE
      P = 2./(FF-2.*S)-2./(FF+2.*S)
    5 CONTINUE
      IF (UU.EQ.0.)  GO TO 6
      Q = 2.*(ATAN((AJ+2.*S)/ABS(UU))-ATAN((AJ-2.*S)/ABS(UU)))/ABS(UU)
      GO TO 7
    6 CONTINUE
      Q = 2./(AJ-2.*S)-2./(AJ+2.*S)
    7 CONTINUE
      Z1 = (S*S+FF*S+GG)/(S*S-FF*S+GG)
      Z2 = (S*S+AJ*S+AK)/(S*S-AJ*S+AK)
      A1IJ = (P*A+.5*BB*ALOG(Z1))/(2.*PI)
      A3IJ = (CL*P+2.*BB*S+CO*ALOG(Z1))/(2.*S*PI)
      CALL CONCAL (DD,EE,AJ,AK,S,A,B,C,D,F,G,CJ,CK,CL,CM,CN,CO,CP,2)
      A2IJ = -(Q*A+.5*EE*ALOG(Z2))/(2.*PI)
      A4IJ = -(CL*Q+2.*EE*S+CO*ALOG(Z2))/(2.*S*PI)
      DO 8 K=1,NLLINE
      KK = K+1
      KCHK = NSPT(KK)-1
      IF (J.EQ.KCHK)  GO TO 9
    8 CONTINUE
      WDOWN = WDOWN+.5*(GAM(J+1)+GAM(J))*(A1IJ+A2IJ)+.5*(GAM(J+1)-GAM(J)
     1)*(A3IJ+A4IJ)
      GO TO 10
    9 CONTINUE
      WDOWN = WDOWN+.5*GAM(J)*(A1IJ+A2IJ-A3IJ-A4IJ)
   10 CONTINUE
C
C     DOWNWASH AT WING IS .5 WASH AT MINUS INFINITY
C
      WDOWN = WDOWN/2.
      WOP = WDOWN/COS(PPP(I))
      WRITE (6,134) I,WDOWN,WOP
      DNWSH(I) = WDOWN
   11 CONTINUE
      IF (ISTOP.GE.2)  GO TO 124
      ISTOP = ISTOP+1
```

```
      GO TO 14
C
C     GENERAL GEOMETRY CALCULATIONS
C
   12 CONTINUE
      READ (5,136) (YHH(I),ZHH(I),PPP(I),I=NS,NTOTT)
      HOT = ABS(YHH(NS))+SNN(NS)*COS(PPP(NS))
      H(LSTART) = 2.*HOT
      DTOT = 0.
      DO 13 I=1,NTOT
      DTOT = DTOT+SNN(I)
   13 CONTINUE
      DTOT = 2.*DTOT
      WRITE (6,138) DTOT
      DTO(LSTART) = DTOT
      DTOB = DTOB+DTOT
   14 CONTINUE
      GO TO 19
C
C     GEOMETRY FOR CIRCULAR ARC AIRFOIL
C
   15 CONTINUE
      NTOT1 = NTOT+1
      READ (5,136) BET,THET,HOT
      THET = THET*DTR
      H(LSTART) = 2.*HOT
      D = BET*HOT
      R = HOT/SIN(THET)
      DTHETA = THET/(FLOAT(NTOT))
      DO 16 I=1,NTOT
      YT = -R*SIN(THET-FLOAT(I-1)*DTHETA)
      ZT = -R+R*COS(THET-FLOAT(I-1)*DTHETA)
      YT1 = -R*SIN(THET-FLOAT(I)*DTHETA)
      ZT1 = -R+R*COS(THET-FLOAT(I)*DTHETA)
      II = NS-1+I
      PPP(II) = ATAN((ZT-ZT1)/(YT-YT1))
      SNN(II) = .5*SQRT((YT-YT1)**2+(ZT-ZT1)**2)
      YHH(II) = .5*(YT+YT1)
      ZHH(II) = .5*(ZT+ZT1)
   16 CONTINUE
      DTOT = 0.
      DO 17 I=NS,NTOTT
```

```
      DTOT = DTOT+SNN(I)
   17 CONTINUE
      DTOT = 2.*DTOT
      WRITE (6,138) DTOT
      DTO(LSTART) = DTOT
      DTOB = DTOB+DTOT
      IF (NS.NE.1)  GO TO 18
      DTO1 = DTOT
   18 CONTINUE
   19 CONTINUE
      GO TO 29
C
   20 CONTINUE
C
C     GEOMETRY CALCS FOR WAKE MADE OF STRAIGHT SEGMTS(CONCTD)
C
      READ (5,137) NBRK
      NBR = NBRK-1
      READ (5,137) (LSEG(I),I=1,NBR)
C     NBRK EQUALS NUMBER OF DIHEDRAL CHANGES OR JCTS WITH
C     OTHER LIFTING LINES   +2
      READ (5,136) (YY(I),Z(I),PP(I),I=1,NBRK)
      DO 21 I=1,NBRK
   21 WRITE (6,132) YY(I),Z(I),PP(I)
      DO 22 I=1,NBR
   22 PERIF(I) = SQRT((Z(I+1)-Z(I))**2+(YY(I+1)-YY(I))**2)
      DTOT = 0.
      DO 23 I=1,NBR
   23 DTOT = DTOT+PERIF(I)
      WRITE (6,138) DTOT
      DTO(LSTART) = DTOT
      DTOB = DTOB+DTOT
      IF (NS.NE.1)  GO TO 24
      DTO1 = DTOT
   24 CONTINUE
      DO 25 I=1,NBRK
   25 PP(I) = DTR*PP(I)
      B(LSTART) = 2.*ABS(YY(NBRK))
      IF (ICNTRL.EQ.8)  GO TO 30
      DO 26 I=1,NBR
      SEG(I) = 0.5*PERIF(I)/LSEG(I)
   26 CONTINUE
```

```
      S = SEQ(NBR)
      SNN(NS) = S
      PPP(NS) = PP(NBRK)
      YHH(NS) = YY(NBRK)+S*COS(PPP(NS))
      ZHH(NS) = Z(NBRK)+S*SIN(PPP(NS))
      N1 = NS
      DO 28 J=1,NBR
      NSEG = N1+LSEG(NBRK-J)-1
      II = N1+1
      LL = NBR-J
      LH = LL+1
      DO 27 I=II,NSEG
      SNN(I) = SEQ(LH)
      YHH(I) = YHH(I-1)+2.*SEQ(LH)*COS(PPP(N1))
      ZHH(I) = ZHH(I-1)+2.*SEQ(LH)*SIN(PPP(N1))
   27 PPP(I) = PPP(N1)
      N1 = N1+LSEG(NBRK-J)
      IF (LL.EQ.0)  GO TO 28
      SNN(N1) = SEQ(LL)
      YHH(N1) = YHH(NSEG)+COS(PP(LH))*SEQ(LL)+COS(PPP(NSEG))*SEQ(LH)
      ZHH(N1) = ZHH(NSEG)+SIN(PP(LH))*SEQ(LL)+SIN(PPP(NSEG))*SEQ(LH)
      PPP(N1) = PP(LH)
   28 CONTINUE
   29 CONTINUE
      GO TO 42
C
C     COSINE SPACING CALCULATIONS
C
   30 CONTINUE
      IF (YY(1).EQ.0.)  GO TO 32
      DO 31 I=1,NBR
   31 DTHETE(I) = PI/(FLOAT(LSEG(NBRK-I)))
      GO TO 35
   32 DO 34 I=1,NBR
      IF (I.EQ.NBR)  GO TO 33
C     THE VARIABLE DTHETA IS NAMED ARRAY DTHETE(I) IN ICNTRL=8
      DTHETE(I) = PI/(FLOAT(LSEG(NBRK-I)))
      GO TO 34
   33 CONTINUE
      DTHETE(I) = PI/(2.*FLOAT(LSEG(NBRK-I)))
   34 CONTINUE
      IF (NBR.NE.1)  GO TO 35
```

```
      SNN(NS) = 0.5*PERIF(1)*(1.-COS(DTHETE(1)))                        A 253
      GO TO 36                                                          A 254
   35 BH = 0.5*PERIF(NBR)                                               A 255
      SNN(NS) = 0.5*BH*(1.-COS(DTHETE(1)))                              A 256
   36 PPP(NS) = PP(NBRK)                                                A 257
      NN = 0                                                            A 258
      DO 40 I=1,NBR                                                     A 259
      NN = NN+LSEG(NBRK-I)                                              A 260
      IF (I.EQ.NBR.AND.YY(1).EQ.0.)  GO TO 37                           A 261
      BH = 0.5*PERIF(NBRK-I)                                            A 262
      GO TO 38                                                          A 263
   37 CONTINUE                                                          A 264
      BH = PERIF(NBRK-I)                                                A 265
   38 CONTINUE                                                          A 266
      LL = LSEG(NBRK-I)                                                 A 267
      DO 39 J=1,LL                                                      A 268
      IF (I.EQ.1.AND.J.EQ.1)  GO TO 39                                  A 269
      NM = NN-LL+J                                                      A 270
      KM = NS+NM-1                                                      A 271
      PPP(KM) = PP(NBRK+1-I)                                            A 272
      SNN(KM) = 0.50*BH*(COS(FLOAT(J)*DTHETE(I))-COS(FLOAT(J-1)*DTHETE(I A 273
     1)))                                                               A 274
      SNN(KM) = ABS(SNN(KM))                                            A 275
   39 CONTINUE                                                          A 276
   40 CONTINUE                                                          A 277
      YHH(NS) = YY(NBRK)+SNN(NS)*COS(PPP(NS))                           A 278
      ZHH(NS) = Z(NBRK)+SNN(NS)*SIN(PPP(NS))                            A 279
      NSEG = NTOT                                                       A 280
      DO 41 II=2,NSEG                                                   A 281
      I = NS-1+II                                                       A 282
      YHH(I) = YHH(I-1)+SNN(I-1)*COS(PPP(I-1))+SNN(I)*COS(PPP(I))       A 283
      ZHH(I) = ZHH(I-1)+SNN(I-1)*SIN(PPP(I-1))+SNN(I)*SIN(PPP(I))       A 284
   41 CONTINUE                                                          A 285
   42 CONTINUE                                                          A 286
      IF (LSTART.EQ.NLLINE)  GO TO 43                                   A 287
      LSTART = LSTART+1                                                 A 288
      GO TO 2                                                           A 289
   43 CONTINUE                                                          A 290
      IF (ICNTRL.EQ.6)  WRITE (6,139)                                   A 291
      IF (ICNTRL.EQ.7)  WRITE (6,140)                                   A 292
      IF (ICNTRL.EQ.8)  WRITE (6,141)                                   A 293
      LN = NLLINE+1                                                     A 294
```

```
      NSPT(LN) = NTOTT+1
      WRITE (6,142)
      WRITE (6,143) (I,YHH(I),ZHH(I),PPP(I),I=1,NTOTT)
      WRITE (6,144)
      WRITE (6,145) (I,SNN(I),I=1,NTOTT)
C
C     READ,WRITE CL SREF , ETC
C
      READ (5,136) CLDES,SREF
      BSAVE = B(1)
      IF (NLLINE.EQ.1)  GO TO 45
      DO 44 I=2,NLLINE
      BTEM = B(I)
   44 BSAVE = AMAX1(BSAVE,BTEM)
   45 CONTINUE
      ARAT = BSAVE*BSAVE/SREF
      SMIN = SNN(1)
      DO 46 I=2,NTOTT
      STEM = SNN(I)
   46 SMIN = AMIN1(SMIN,STEM)
      TOL = 5.E-05*SMIN*NSPT(2)
      TOL2 = TOL
      TOL3 = 0.005*SMIN
      WRITE (6,157) TOL2
C
C     END GENERAL GEOM CALCS
C
   47 CONTINUE
      IF (IDRAG.EQ.2)  GO TO 52
      WRITE (6,146) CLDES,SREF,BSAVE,ARAT
      WRITE (6,149)
      IDRAG = IDRAG+1
C     SET UP ALL A FOR MUNK CRITERION OPTIMIZATION
      DO 49 I=1,NTOTT
      DO 48 J=1,NTOTT
      CALL WCAL (I,J,NTOTT,YHH,ZHH,PPP,AOPT(I,J))
      AOPT(I,J) = AOPT(I,J)*2./SREF
   48 CONTINUE
   49 CONTINUE
      IL = NTOTT+1
      DO 50 I=1,NTOTT
      BSIM(I,1) = COS(PPP(I))
```

```
      AOPT(I,IL) = COS(PPP(I))
   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 EQ NTOTT
      GO TO 77
   52 CONTINUE
C     SET UP ALL A THROUGH F FOR ALL I,J TO DO DIRECT OPTIMIZATION
      IDRAG = IDRAG+1
      IF (NLLINE.NE.1)  GO TO 2
      NTOT2 = NTOTT+2
      NTOT3 = NTOTT+3
      DO 53 I=1,NTOT3
      DO 53 J=1,NTOT3
      T1(I,J) = 0.
      T2(I,J) = 0.
      T3(I,J) = 0.
      T4(I,J) = 0.
      T5(I,J) = 0.
      T6(I,J) = 0.
   53 CONTINUE
      DO 54 I=1,NTOTT
      I1 = I+1
      DO 54 J=1,NTOTT
      J1 = J+1
      CALL DRACAL (I,J,YHH,ZHH,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
C     DIRECT OPTIMIZATION MODS
C
      DO 56 I=1,NTOTT
      I1 = I+1
      DO 56 J=1,NTOTT
```

```
      J1 = J+1
      AOPT(I,J) = T3(I1,J1)-T4(I1,J1)-T5(I1,J1)+T6(I1,J1)+T3(I1-1,J1)-T4
     1(I1-1,J1)+T5(I1-1,J1)-T6(I1-1,J1)+T3(I1-1,J1-1)+T4(I1-1,J1-1)+T5(I
     21-1,J1-1)+T6(I1-1,J1-1)+T3(I1,J1-1)+T4(I1,J1-1)-T5(I1,J1-1)-T6(I1,
     3J1-1)+1.5*SNN(I)*(T1(I1,J1)-T2(I1,J1))+1.5*SNN(I)*(T1(I1,J1-1)+T2(
     4I1,J1-1))
      IF (I.EQ.1)  GO TO 55
      AOPT(I,J) = AOPT(I,J)+.5*SNN(I-1)*(T1(I1-1,J1)-T2(I1-1,J1)+T1(I1-1
     1,J1-1)+T2(I1-1,J1-1))
   55 CONTINUE
      AOPT(I,J) = AOPT(I,J)*0.25
   56 CONTINUE
      DO 64 I=1,NTOTT
      I1 = I+1
      IF (I.EQ.1)  GO TO 60
      SCON = 0.5*(SNN(I)+SNN(I-1))
      TEMP = 0.
      DO 59 J=1,NTOTT
      J1 = J+1
      TEMP = .5*SNN(I-1)*(T1(I1,J1)-T2(I1,J1)+T1(I1,J1-1)+T2(I1,J1-1))
      IF (I.EQ.NTOTT)  GO TO 58
      IS = I+1
      DO 57 IP=IS,NTOTT
      IP1 = IP+1
   57 TEMP = TEMP+SCON*(T1(IP1,J1)-T2(IP1,J1)+T1(IP1,J1-1)+T2(IP1,J1-1))
   58 CONTINUE
      AOPT(I,J) = AOPT(I,J)+TEMP
      AOPT(I,J) = AOPT(I,J)/SREF
   59 CONTINUE
      GO TO 63
   60 CONTINUE
      SCON = .5*SNN(1)
      DO 62 J=1,NTOTT
      J1 = J+1
      TEMP = 0.
      DO 61 IP=2,NTOTT
      IP1 = IP+1
      TEMP = TEMP+SCON*(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 I=1,NTOTT
      DO 65 J=1,NTOTT
   65 T1(I,J) = AOPT(J,I)
      DO 66 I=1,NTOTT
      DO 66 J=1,NTOTT
      AOPT(I,J) = 2.*(AOPT(I,J)+T1(I,J))
   66 CONTINUE
      IL = NTOTT+1
      IL1 = IL
      IL2 = IL+1
      DO 67 I=1,IL
      AOPT(I,IL2) = 0.
   67 CONTINUE
      AOPT(IL1,IL2) = CLDES
      AOPT(IL1,1) = 8*COS(PPP(1))*SNN(1)**2/(3.*SREF)
      DO 68 I=2,NTOTT
      AOPT(IL1,I) = (2*COS(PPP(I))*SNN(I)**2+COS(PPP(I-1))*SNN(I-1)**2)*
     14./(3.*SREF)
   68 CONTINUE
      SUMX = 0.
      DO 69 I=2,NTOTT
      SUMX = SUMX+COS(PPP(I))*SNN(I)
   69 CONTINUE
      AOPT(IL1,1) = AOPT(IL1,1)+(4./SREF)*SNN(1)*SUMX
      DO 72 I=2,NTOTT
      SUMX = COS(PPP(I))*SNN(I)*SNN(I-1)
      I1 = I+1
      IF (I.EQ.NTOTT)  GO TO 71
      DO 70 J=I1,NTOTT
      SUMX = SUMX+COS(PPP(J))*SNN(J)*(SNN(I)+SNN(I-1))
   70 CONTINUE
   71 CONTINUE
      AOPT(IL1,I) = AOPT(IL1,I)+(4./SREF)*SUMX
   72 CONTINUE
C
      DO 73 I=1,NTOTT
      AOPT(IL1,I) = 2.*AOPT(IL1,I)
   73 CONTINUE
      DO 74 I=1,NTOTT
      AOPT(I,IL1) = AOPT(IL1,I)
```

```
   74 CONTINUE
      DO 75 I=1,IL1
      CLP(I) = AOPT(IL1,I)
   75 CONTINUE
      WRITE (6,147) (CLP(I),I=1,IL1)
      AOPT(IL1,IL1) = 0.
      DO 76 I=1,IL1
      DO 76 J=1,IL1
      ASIM(I,J) = AOPT(I,J)
      BSIM(I,1) = AOPT(I,IL2)
   76 CONTINUE
      WRITE (6,148)
   77 CONTINUE
      CALL SIMEQ (ASIM,IL1,BSIM,1,DET,IPIVOT,51,ISCALE)
      IT = IL1+1
      DO 78 I=1,IL1
   78 AOPT(I,IT) = BSIM(I,1)
C     SET IL1 BACK TO NTOTT+1
      IL1 = NTOTT+1
      IF (IDRAG.NE.3)  GO TO 81
      CLCHK = 0.
      DO 79 I=1,NTOTT
      AOPT(I,IL2) = BSIM(I,1)
      CLCHK = CLCHK+CLP(I)*AOPT(I,IL2)
   79 CONTINUE
      WRITE (6,150) CLCHK
      DO 80 I=1,IL
   80 WRITE (6,152) I,AOPT(I,IL2)
   81 CONTINUE
      WRITE (6,151)
C
C     CALCULATE BOUND CIRCULATIONS AND
C     CALCULATE CNTRLN DOWNWASH DIVIDED BY FREESTREAM U
C
      NTOT1 = NTOTT+1
      SUMGAM = 0.
      AOPT(NTOT1,IL) = 0.
      IF (IDRAG.NE.3)  GO TO 83
      DO 82 I=1,IL
   82 AOPT(I,IL) = AOPT(I,IL2)
      AOPT(IL,IL) = 0.
   83 CONTINUE
```

```
      DO 84 I=1,NTOTT
      CALL GAMCAL (I,IL,SNN,AOPT,BGAM0)
      BGAM(I) = BGAM0
   84 CONTINUE
      IF (NLLINE.EQ.1)  GO TO 90
      NL1 = NLLINE-1
      DO 85 I=1,NL1
      II = NLLINE-I+1
      I1 = II+1
      JE = NSPT(I1)-1
      GSUM(I) = BGAM(JE)+AOPT(JE,IL)*SNN(JE)
   85 CONTINUE
      DO 89 I=1,NL1
      I1 = NLLINE-I+2
      JE = NSPT(I1)-1
      YT = YHH(JE)+SNN(JE)*COS(PPP(JE))
      IF (ABS(YT).LT.0.0001)  GO TO 89
      JE = NSPT(2)-1
      DO 86 J=1,JE
      JS = J
      IF (YT.LT.YHH(J))  GO TO 87
   86 CONTINUE
   87 CONTINUE
      DO 88 J=JS,JE
      BGAM(J) = BGAM(J)+GSUM(I)
   88 CONTINUE
   89 CONTINUE
   90 CONTINUE
      SUMGAM = 0.
      DO 94 I=1,NTOTT
      DO 91 J=1,NLLINE
      JJ = J+1
      JCHK = NSPT(JJ)-1
      IF (I.EQ.JCHK)  GO TO 92
   91 CONTINUE
      SUMGAM = SUMGAM+COS(PPP(I))*(SNN(I)**2*(AOPT(I+1,IL)+2.*AOPT(I,IL)
     1)/3.+SNN(I)*BGAM(I))
      GO TO 93
   92 CONTINUE
      SUMGAM = SUMGAM+COS(PPP(I))*(SNN(I)**2*2.*AOPT(I,IL)/3.+SNN(I)*BGA
     1M(I))
   93 CONTINUE
```

```
   94 CONTINUE
      IF (NLLINE.NE.1)  GO TO 95
      JE = NSPT(2)-1
   95 CONTINUE
      BGAM(NTOT1) = BGAM(JE)+AOPT(JE,IL)*SNN(JE)
      WDBU = CLDES*SREF/(8.*SUMGAM)
C
C     RESCALE OPT SHED SHEET STRENGTHS
C     TO BE DIVIDED BY U INSTEAD OF W
C
      IF (IDRAG.EQ.3)  GO TO 97
      DO 96 I=1,NTOT1
      BGAM(I) = BGAM(I)*WDBU
   96 AOPT(I,IL) = AOPT(I,IL)*WDBU
   97 CONTINUE
C
C     RESCALE BOUND + SHED STRENGTHS TO CALCULATE AVG + NON-DIM VALUES
C
C     AOPT(I,IL)=OPT SHED SHEET STRENGTHS
C     BGAM(I)=OPT BOUND CIRC VALUES
C     CDRAG(I)=OPT NON DIM BOUND CIRCS
C     GAM(I)= OPT NON DIM SHED SHEET VALUES
C
      SUBGAM = 0.
      SUMGAM = 0.
      NS = NSPT(2)-1
      DO 100 I=1,NS
      TEMP = AOPT(I+1,IL)
      DO 99 J=1,NLLINE
      JJ = J+1
      JCHK = NSPT(JJ)-1
      IF (I.NE.JCHK)  GO TO 98
      TEMP = 0.
   98 CONTINUE
   99 CONTINUE
      SUMGAM = SUMGAM+SNN(I)*(TEMP+AOPT(I,IL))
      SUBGAM = SUBGAM+2.*SNN(I)*(BGAM(I)+(SNN(I)/12.)*(4.*TEMP+8.*AOPT(I
     1,IL)))
  100 CONTINUE
      SUMGAM = SUMGAM/DTOB
      SUBGAM = SUBGAM/DTO(1)
      NS = NSPT(2)-2
```

```
      NS1 = NS+1
      DO 103 I=1,NS
      IF (ICNTRL.EQ.6)  GO TO 101
      CDRAG(I+1) = BGAM(I+1)/SUBGAM
      GO TO 102
  101 CONTINUE
      CDRAG(I+1) = BGAM(I+1)/BGAM(NS1)
  102 CONTINUE
  103 CONTINUE
      JL = NSPT(2)-1
      CDRAG(NTOT1) = BGAM(JL)+AOPT(JL,IL)*SNN(JL)
      CDRAG(NTOT1) = CDRAG(NTOT1)/SUBGAM
      DO 104 I=1,NTOTT
      GAM(I) = AOPT(I,IL)/SUMGAM
  104 CONTINUE
      CDRAG(1) = 0.
      GAM(NTOT1) = 0.
      IF (NLLINE.EQ.1)  GO TO 109
      DO 108 I=2,NLLINE
      JS = NSPT(I)
      II = I+1
      JF = NSPT(II)-1
      SUBGAM = 0.
      DO 106 J=JS,JF
      TEMP = AOPT(J+1,IL)
      IF (J.NE.JF)  GO TO 105
      TEMP = 0.
  105 CONTINUE
      SUBGAM = SUBGAM+2.*SNN(J)*(BGAM(J)+(SNN(J)/12.)*(4.*TEMP+8.*AOPT(J
     1,IL)))
  106 CONTINUE
      SUBGAM = SUBGAM/DTO(I)
      DO 107 J=JS,JF
      CDRAG(J) = BGAM(J)/SUBGAM
  107 CONTINUE
  108 CONTINUE
  109 CONTINUE
      DO 112 I=1,NTOT1
      IF (I.EQ.NTOT1)  GO TO 110
      ETA = 2.*(-YHH(I)+SNN(I)*COS(PPP(I)))/BSAVE
      GO TO 111
  110 CONTINUE
```

```
      ETA = 0.
  111 CONTINUE
      WRITE (6,152) I,RGAM(I),CDRAG(I),AOPT(I,IL),GAM(I),ETA
  112 CONTINUE
C
C     INDUCED DRAG CALC USING ANALYTICAL INT OF WN TIMES ROUND CIRC
C
      CD = 0.
      NLL = NLLINF+1
      DO 119 I=1,NTOTT
      DO 113 K=2,NLL
      ICHK = NSPT(K)-1
      IF (ICHK.EQ.I)  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,NTOTT
      DO 116 K=2,NLL
      JCHK = NSPT(K)-1
      IF (J.EQ.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 DRACAL (I,J,YHH,ZHH,PPP,AINT)
      CDI = XI*YJ*AINT(1)+XI*ZJ*AINT(2)+YI*YJ*AINT(3)+YI*ZJ*AINT(4)+ZI*Y
     1J*AINT(5)+ZI*ZJ*AINT(6)
      CD = CD+CDI
  119 CONTINUE
      CD = CD*2./SREF
      IF (IDRAG.EQ.3)  GO TO 120
```

```
      WRITE (6,154) CD
      GO TO 121
  120 CONTINUE
      WRITE (6,153) CD
  121 CONTINUE
      DIDEAL = SREF*CLDES**2/(PI*(HSAVE)**2)
      IF (IDRAG.EQ.3)  GO TO 122
      WRITE (6,155) DIDEAL,WDBU
  122 CONTINUE
      DEFF = DIDEAL/CD
      WRITE (6,156) DEFF
      ISTOP = ISTOP+1
      DO 123 I=1,NTOTI
  123 GAM(I) = AOPT(I,IL)
      GO TO 3
  124 CONTINUE
      CDAPP = 0.
      DO 128 I=1,NTOTT
      DO 125 J=1,MLLINF
      JJ = J+1
      JCHK = NSPT(JJ)-1
      IF (I.EQ.JCHK)  GO TO 126
  125 CONTINUE
      CDAPP = CDAPP+2.*SNN(I)*DNWSH(I)*(BGAM(I)+.25*(AOPT(I+1,IL)+3.*AOP
     1T(I,IL))*SNN(I))
      GO TO 127
  126 CONTINUE
      CDAPP = CDAPP+2.*SNN(I)*DNWSH(I)*(BGAM(I)+.75*AOPT(I,IL)*SNN(I))
  127 CONTINUE
  128 CONTINUE
      CDAPP = 4.*CDAPP/SREF
      WRITE (6,158) CDAPP
      IF (IDRAG.EQ.3)  GO TO 2
      GO TO 47
  129 CONTINUE

  130 FORMAT(1H1)
  131 FORMAT(5I5)
  132 FORMAT(25X,3F15.5)
  133 FORMAT(//33X,1HI,7X,8HDOWNWASH,7X,10HW/COS(PHI)/)
  134 FORMAT(30X,I5,2F15.5)
  135 FORMAT(/30X,22HGENERAL INPUT GEOMETRY//)
```

```
      136 FORMAT(6F10.0)
      137 FORMAT(10I5)
      138 FORMAT(30X,33HTOTAL PLANFORM PERIPHERAL LENGTH=,F15.5/)
      139 FORMAT(/30X,17HCIRCULAR ARC WING)
      140 FORMAT(/30X,21HEQUAL SEGMENT SPACING)
      141 FORMAT(/30X,22HCOSINE SEGMENT SPACING)
      142 FORMAT(/24X,8HSEGMT NO.8X,1HY,15X,1HZ,13X,3HPHI/)
      143 FORMAT(25X,I5,3F15.6)
      144 FORMAT(/27X,1HI,8X,6HSNN(I)/)
      145 FORMAT(25X,I5,F12.5)
      146 FORMAT(/// 25X,20HDESIGN LIFT COEF. = ,F10.5//,25X,22HWING REFEREN
         1CE AREA = ,F10.5//,25X, 16HREF WING SPAN = , F10.5//,25X,
         215HASPECT RATIO = ,F10.5/)
      147 FORMAT(/1X,10F10.3)
      148 FORMAT(///25X,54HDIRECT OPTIMIZATION USING ANALYTICAL EXPRESSION F
         1OR CD//)
      149 FORMAT(///25X,37HOPTIMUM LOADING USING MUNKS CRITERION//)
      150 FORMAT(//23X,46HLIFT COEFF CALCULATED FROM CLP AND SOLVED GAM=,
         1E13.5/)
      151 FORMAT(//22X,5HSEGMT,3X,10HBOUND CIRC,2X,8HBGAM/AVE,4X,10HSHED STR
         1H,5X,7HGAM/AVE,8X,3HETA//)
      152 FORMAT(20X,I5,5E13.5)
      153 FORMAT(///25X,67HCD CALCULATED USING SUB DRACAL AND LOADS FROM DIR
         1ECT OPTIMIZATION = ,F15.5//)
      154 FORMAT(///25X,64HCD CALCULATED USING SUB DRACAL AND OPTIM LOADS US
         1ING MUNK CRIT =,F15.5//)
      155 FORMAT(25X,19HCD FOR FLAT WING = ,E15.5/25X,35HRATIO OF ZERO DIHED
         1RAL DOWNWSH/U = ,E15.5/)
      156 FORMAT(//25X,49HINDUCED DRAG EFFICIENCY FOR WINGS OF EQUAL SPAN =
         1,F15.6/)
      157 FORMAT(/25X,14HTOL IN SNTAN =,E15.5/)
      158 FORMAT(/25X,28HAPPROX CD USING SOLVED BOUND/25X,42HCIRCULATIONS AN
         1D WASHES AT SEG MIDPOINTS =,E13.5/)
          END
```

```
      SUBROUTINE GAMCAL (I,IL,SNN,AOPT,BGAM0)
C
C     CALCULATE BOUND CIRCULATION AT LEFT END PT OF SEGMENT I,BGAM
C     AOPT LAST COLUMN CONTAINS ARRAY OF OPTIMIZED SHED SHEET STRENGTHS
C
      COMMON  /FEN/ NSPT(10),NLLINE
      DIMENSION  SNN(1), AOPT(52,1)
      BGAM0 = 0.
      DO 1 K=1,NLLINE
      KK = NLLINE-K+1
      KCHK = NSPT(KK)
      IF (I.EQ.KCHK)  GO TO 4
      IF (I.GT.KCHK)  GO TO 2
    1 CONTINUE
    2 CONTINUE
      KST = KCHK
      BGAM0 = AOPT(KST,IL)*SNN(KST)+AOPT(I,IL)*SNN(I-1)
      KCHK1 = KCHK+1
      IF (I.EQ.KCHK1)  GO TO 5
      IM = I-1
      IP = KCHK+1
      DO 3 J=IP,IM
      BGAM0 = BGAM0+AOPT(J,IL)*(SNN(J-1)+SNN(J))
    3 CONTINUE
      GO TO 5
    4 BGAM0 = 0.
    5 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WCAL (I,J,NTOT,YHH,ZHH,PPP,AAAA)
C
C     CALCULATE MATRIX COEFFICIENT FOR DRAG OPTIMIZATION USING MUNK CRIT
C     FINDS COEF OF ITH DOWNWASH DUE TO JTH SHED SHEET STRENGTH
C     IE, FINDS COEF MULTIPLYING STRENGTH J IN EQUATION I
C
      DIMENSION  YHH(1), ZHH(1), PPP(1)
      COMMON  /SEG/ SNN(5))
      COMMON  /FEN/ NSPT(10),NLLINE
      INTEGER  P
      PI = 4.*ATAN(1.)
      AAAA = 0.
      ICNTRL = 0
      P = I
      K = J
    1 CONTINUE
      CALL CCAL  (P,K,YHH,ZHH,PPP,SNN(K),AA,BB,DD,FF,GG,EE,AJ,AK,RR,TT,UU
     1,WW)
      CALL CONCAL  (AA,BB,FF,GG,SNN(K),A,B,C,D,F,G,BJ,BK,BL,BM,BN,BO,BP,1
     1)
      IF  (RR.EQ.0.)   GO TO 2
      EXPR = 2.*(ATAN2(C,ABS(RR))-ATAN2(D,ABS(RR)))/(ABS(RR))
      GO TO 3
    2 CONTINUE
      EXPR = 2./D-2./C
    3 CONTINUE
      RLOG = ALOG(F/G)
      A1PK = (A*EXPR+.5*BB*RLOG)/(2.*PI)
      A3PK = (BL*EXPR+2.*BB*SNN(K)+BO*RLOG)/(2.*PI*SNN(K))
      CALL CONCAL  (DD,FF,AJ,AK,SNN(K),A,B,C,D,F,G,BJ,BK,BL,BM,BN,BO,BP,2
     1)
      IF  (UU.EQ.0.)   GO TO 4
      EXPR = 2.*(ATAN2(C,ABS(UU))-ATAN2(D,ABS(UU)))/(ABS(UU))
      GO TO 5
    4 CONTINUE
      EXPR = 2./D-2./C
    5 CONTINUE
      RLOG = ALOG(F/G)
      A2PK = -(A*EXPR+.5*EF*RLOG)/(2.*PI)
      A4PK = -(BL*EXPR+2.*EE*SNN(K)+BO*RLOG)/(2.*PI*SNN(K))
      IF  (ICNTRL.EQ.2)   GO TO 7
      AAAA = (A1PK+A2PK)*.50-(A3PK+A4PK)*.50
```

```
      ICNTRL = 2
      DO 6 L=1,NLLINE
      JCHK = NSPT(L)
      IF (K.EQ.JCHK)  GO TO 8
    6 CONTINUE
      K = J-1
      GO TO 1
    7 CONTINUE
      AAAA = AAAA+0.5*(A1PK+A2PK+A3PK+A4PK)
    8 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CCAL (I,J,YHH,ZHH,PPP,S,AA,BB,DD,FF,GG,EE,JJ,KK,RR,TT,U
     1U,WW)
C
C     SUBROUTINE CCAL
C
C     CALCULATES GEOMETRICAL CONSTANTS NEEDED IN EVALUATION OF INTEGRALS
C     FOR VARYING I AND J VALUES
C
      REAL  JJ,KK
      DIMENSION  YHH(1), ZHH(1), PPP(1)
      DYIJ = YHH(I)-YHH(J)
      DZIJ = ZHH(I)-ZHH(J)
      COI = COS(PPP(I))
      SII = SIN(PPP(I))
      COJ = COS(PPP(J))
      SIJ = SIN(PPP(J))
      AA = DYIJ*COI+DZIJ*SII
      BB = -COS(PPP(J)-PPP(I))
      FF = -2*(DYIJ*COJ+DZIJ*SIJ)
      GG = DYIJ*DYIJ+DZIJ*DZIJ
      DYIJP = YHH(I)+YHH(J)
      DZIJP = DZIJ
      DD = DYIJP*COI+DZIJP*SII
      EE = COS(PPP(J)+PPP(I))
      JJ = 2.*(DYIJP*COJ-DZIJP*SIJ)
      KK = DYIJP*DYIJP+DZIJP*DZIJP
      RR = 2*(DYIJ*SIJ-DZIJ*COJ)
      TT = 2*SIN(PPP(J)-PPP(I))
      UU = 2.*(DYIJP*SIJ+DZIJP*COJ)
      WW = 2.*SIN(PPP(J)+PPP(I))
      RETURN
      END
```

```
      SUBROUTINE CONCAL (AA,BB,FF,GG,S,A,B,C,D,F,G,J,K,L,M,N,O,P,ICNTRL)
C
C     SUBROUTINE CONCAL
C
C     CALCULATES GEOMETRICAL CONSTANTS NEEDED IN EVALUATION OF INTEGRALS
C
      REAL  J,K,L,M,N
      A = AA-0.5*BB*FF
      B = 1.-BB*BB
      C = FF+2.*S
      D = FF-2.*S
      F = S*S+S*FF+GG
      G = S*S-S*FF+GG
      J = 2.*(AA+S*BB)
      K = 2.*(AA-S*BB)
      L = 0.5*(BB*FF*FF-AA*FF-2.*BB*GG)
      M = 0.5*(-FF-6.*AA*BB+4.*FF*BB*BB)
      N = 2.*(BB*BB-1.)*BB
      O = 0.5*(AA-FF*BB)
      P = 0.5*(1.-2.*BB*BB)
      RETURN
      END
```

```
      SUBROUTINE SNTAN (S,C,BB,RR,TT,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)
C
C     SUBROUTINE SNTAN
C
C     EVALUATES INTEGRALS OF THE FORM S**N*ATAN((C+2*BB*S)/(RR+S*TT))
C     ALL DIVIDED BY (RR+S*TT)
C     WITH RESPECT TO S BETWEEN LIMITS OF -S AND S FOR N=0,1,2,3.
C
C     ATAN PART OF INTEGRAND APPROXIMATED AS A QUADRATIC IN S WHICH IS
C     FORCED THROUGH ATAN VALUES AT -S,0,AND S.
C
C     A,C ARE CALCULATED IN SUBROUTINE CONCAL
C     BB,RR,TT ARE CALCULATED IN SUBROUTINE CCAL
C
C     RESULTS ARE RTAN,RSTAN,RS2TAN,RS3TAN
C     APPROXIMATE INTEGRAL EVALUATED USING MACSYMA PROGRAM OF MIT PROJ.
C     EVALUATION OF INTEGRALS FOR TT=0. BEGIND AT AT LABEL 10
C     SINGULAR INTEGRALS EVALUATED AT APPROXIMATE ENDPOINTS,+-SAWAY
C     MIDRANGE SINGULARITIES EXCLUDED,ATAN PART OF INT APPROX-
C     IMATED AS 2 QUADRATICS
C
      DIMENSION  AA1(3,3), AA(3), IPIVOT(3)
      COMMON  /TELL/ TOL,TOL2
      RRR = RR
      SSS = S
      CCC = C
      RTAN = 0.
      RSTAN = 0.
      RS2TAN = 0.
      RS3TAN = 0.
      RS4TAN = 0.0
      IF (ABS(TT).LT.1E-05.AND.ABS(RR).LT.1E-05)  GO TO 7
      IF (TT.EQ.0.0.AND.RR.EQ.0.0)  GO TO 7
      IF (TT.EQ.0.0)  GO TO 4
C
C     FIRST, CHECK FOR MIDRANGE SINGULARITIES,EXCLUDING ANY FOUND
C
      SZERO = -RR/TT
      IF (ABS(ABS(SZERO)-S).LT.1E-03.AND.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.S)  GO TO 16
    1 CONTINUE
```

```
      C = ATAN2(CCC,ABS(RR))
      C1 = .5*(ATAN2((CCC+2.*BB*S),ABS(RR+TT*S))+ATAN2((CCC-2.*BB*S),ABS
     1(RR-TT*S)))-C
      C1 = C1/(S*S)
      C2 = (ATAN2((CCC+2.*BB*S),ABS(RR+TT*S)))/S-C/S-C1*S
C
C     INTEGRAND NOW IS (C1*S*S+C2*S+C)*S**N/(RR+TT*S))
C
      CLOGR = ALOG(ABS((RR+TT*S)/(RR-TT*S)))
      CON0 = (C*TT*TT-C2*RR*TT+C1*RR*RR)/(TT**3)
      CON1 = (C2*TT-C1*RR)/(TT*TT)
      RTAN = CON0*CLOGR+2.*S*CON1
      CON2 = (C*TT*TT-C2*RR*TT+C1*RR*RR)/(TT**3)
      CON3 = (C*RR*TT*TT-C2*RR*RR*TT+C1*RR**3)/(TT**4)
      RSTAN = 2.*S*CON2-CON3*CLOGR+2.*C1*TT*TT*S**3/(3*TT**3)
      CON4 = (C*RR*RR*TT*TT-C2*RR**3*TT+C1*RR**4)/(TT**5)
      CON5 = 4.*C2*TT**3-4.*C1*RR*TT**2
      CON5 = CON5/(12*TT**4)
      CON6 = (-C*RR*TT*TT+C2*RR*RR*TT-C1*RR**3)/(TT**4)
      RS2TAN = CON4*CLOGR+2.*CON5*S**3+2.*S*CON6
      CON8 = 20.*(C*TT**4-C2*RR*TT**3+C1*RR**2*TT**2)/(60*TT**5)
      CON9 = (C*RR**2*TT**2-C2*RR**3*TT+C1*RR**4)/(TT**5)
      CON10 = C*RR**3*TT**2-C2*RR**4*TT+C1*RR**5
      CON10 = CON10/(TT**6)
      RS3TAN = 2.*CON8*S**3+2.*CON9*S-CON10*CLOGR+24.*C1*TT**4*S**5/(60*
     1TT**5)
      CONA = (C*RR**4*TT**2-C2*RR**5*TT+C1*RR**6)/(TT**7)
      CONB = (C2*TT-C1*RR)/(5*TT**2)
      CONC = (-C*RR*TT**2+C2*RR**2*TT-C1*RR**3)/(3*TT**4)
      COND = CONC*RR**2/(TT**2/3)
      RS4TAN = CONA*CLOGR+CONB*2*S**5+CONC*2*S**3+COND*2*S
      IF (RR.GT.0.)  GO TO 2
      RTAN = -RTAN
      RSTAN = -RSTAN
      RS2TAN = -RS2TAN
      RS3TAN = -RS3TAN
      RS4TAN = -RS4TAN
    2 CONTINUE
      GO TO 13
    3 CONTINUE
      SAWAY = S-TOL
      S = SAWAY
```

```
          GO TO 1
C
C         FOR CASE OF RR NOT ZERO,TT=0.0
C
        4 CONTINUE
          RR = ABS(RR)
          ALNNUM = (2*BB)**2*S**2+4*C*BB*S+RR**2+C**2
          ALNDEN = (2*BB)**2*S**2-4*C*BB*S+RR**2+C**2
          IF (ALNNUM.EQ.0.0.OR.ALNDEN.EQ.0.0)  GO TO 5
          GO TO 6
        5 CONTINUE
          S = S-TOL
          ALNNUM = (2*BB)**2*S**2+4*C*BB*S+RR**2+C**2
          ALNDEN = (2*BB)**2*S**2-4*C*BB*S+RR**2+C**2
        6 RATLN = ALOG(ALNNUM/ALNDEN)
          TNDIF = ATAN2((C+2.*BB*S),RR)-ATAN2((C-2.*BB*S),RR)
          TNSUM = ATAN2((C+2.*BB*S),RR)+ATAN2((C-2.*BB*S),RR)
          RTAN = -(.25*RR/BB)*RATLN+0.5*C*TNDIF/BB
          RTAN = RTAN/RR+S*TNSUM/RR
          RSTAN = 0.5*(S*S+(RR*RR-C*C)/(2*BB)**2)*TNDIF-.5*RR*S/BB+((.5*C*RR
         1)/(2*BB)**2)*RATLN
          RSTAN = RSTAN/RR
          RS2TAN = (S**3/3)*TNSUM+((RR**3-3*C**2*RR)/(48*BB**3))*RATLN+C*RR*
         1S/(3*BB**2)-((6*C*RR**2-2*C**3)/(6*(2*BB)**3))*TNDIF
          RS2TAN = RS2TAN/RR
          RS3TAN = (S**4/4)*TNDIF-((C*RR**3-C**3*RR)/(32*BB**4))*RATLN-TNDIF
         1*(RR**4-6*C**2*RR**2+C**4)/(64*BB**4)-RR*S**3/(12*BB)-S*(9*C**2*RR
         2**2-3*RR**4)/(48*RR*BB**3)
          RS3TAN = RS3TAN/RR
          RS4TAN = (S**5/(5*RR))*TNSUM+TNDIF*(5*C*RR**6-10*C**3*RR**4+C**5*R
         1R**2)/(160*BB**5*RR**3)-RATLN*(RR**6-10*C**2*RR**4+5*C**4*RR**2)/(
         2320*RR**2*BB**5)+C*S**3/(15*BB**2)+S*(C**3-C*RR**2)/(10*BB**4)
          GO TO 13
C
C         FOR CASE OF RR=TT=0.0, IF I=J
C
        7 TOP = C+2.*BB*S
          BOT = C-2.*BB*S
          IF (C.EQ.0.0.AND.BB.EQ.0.0)  GO TO 13
          SBAD = -C/(2.*BB)
          SBADAB = ABS(SBAD)
          IF (SBADAB.LT.S)  GO TO 8
```

```
      GO TO 9
    8 CONTINUE
      WRITE (6,25)
      SUL = SBAD-TOL
      SLL = SBAD+TOL
      CLOGR1 = ALOG(TOP/(C+2.*BB*SLL))
      CLOGR2 = ALOG((C+2.*BB*SUL)/BOT)
      CLOGR = CLOGR1+CLOGR2
      RTAN = -(.5/BB)*CLOGR
      RSTAN = (.25*C/BB**2)*CLOGR-(.5/BB)*(2.*S-SLL+SUL)
      RS2TAN = -(C**2/(8.*BB**3))*CLOGR+(C/(4.*BB**2))*(2.*S-SLL+SUL)-(.
     125/BB)*(SUL*SUL-SLL*SLL)
      RS3TAN = (C**3/(16.*BB**4))*CLOGR-(2.*S**3-SLL**3+SUL**3)/(6.*BB)-
     1(C*C/(8.*BB**3))*(2.*S-SLL+SUL)+(C/(8.*BB*BB))*(SUL*SUL-SLL*SLL)
      RS4TAN = -(C**4/(32.*BB**5))*CLOGR-(SUL**4-SLL**4)/(8.*BB)+(C/(12.
     1*BB*BB))*(2.*S**3-SLL**3+SUL**3)-C**2*(SUL**2-SLL*SLL)/(16.*BB**3)
     2+C**3*(2.*S-SLL+SUL)/(16.*BB**4)
      GO TO 13
    9 CONTINUE
      IF (ABS(TOP).LT.1E-9.OR.ABS(BOT).LT.1E-9)  GO TO 12
      IF (TOP.LE.0.0)  GO TO 11
      IF (BOT.LE.0.0)  GO TO 11
   10 CLOGR = ALOG(TOP/BOT)
      RTAN = -(1/BB)*CLOGR
      RTAN = RTAN/2
      RSTAN = (.25*C/BB**2)*CLOGR-S/BB
      RS2TAN = -(C**2/(8*BB**3))*CLOGR+C*S/(2.*BB**2)
      RS3TAN = (C**3/(16.*BB**4))*CLOGR-S**3/(3.*BB)-S*C**2/(4.*BB**3)
      RS4TAN = -(C**4/(32*BB**5))*CLOGR+2.*C*S**3/(12*BB**2)+2.*S*C**3/(
     116*BB**4)
      GO TO 13
   11 TPDBBT = TOP/BOT
      IF (TPDBBT.GT.0.0)  GO TO 10
      IF (TOP.LT.0.0)  GO TO 14
      IF (BOT.LT.0.0)  GO TO 14
   12 CONTINUE
      SAWAY = S-TOL
      TOP = C+2.*BB*SAWAY
      BOT = C-2.*BB*SAWAY
      CLOGR = ALOG(TOP/BOT)
      RTAN = -(1/BB)*CLOGR
      RTAN = RTAN/2
```

```
      RSTAN = (.25*C/BB**2)*CLOGR-S/BB
      RS2TAN = -(C**2/(8*BB**3))*CLOGR+C*S/(2.*BB**2)
      RS3TAN = (C**3/(16.*BB**4))*CLOGR-S**3/(3.*BB)-S*C**2/(4.*BB**3)
      RS4TAN = -(C**4/(32*BB**5))*CLOGR+2.*C*S**3/(12*BB**2)+2.*S*C**3/(
     116*BB**4)
      GO TO 13
   13 CONTINUE
C
C     WRITE STATEMENTS GO HERE IF NEEDED
C
      GO TO 15
   14 WRITE (6,26)
   15 CONTINUE
      GO TO 24
C
C     FOR CASE OF RR,TT NOT ZERO, BUT WITH MIDRANGE SINGULARITY
C
   16 CONTINUE
      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.*BB*S),ABS(RR+TT*S))
      ANG2 = ATAN2((C+2.*BB*SMID1),ABS(RR+TT*SMID1))
      ANG3P = ATAN2((C+2.*BB*SLL),ABS(RR+TT*SLL))
      ANG3 = ATAN2((C-2.*BB*S),ABS(RR-TT*S))
      ANG4 = ATAN2((C+2.*BB*SMID2),ABS(RR+TT*SMID2))
      ANG5 = ATAN2((C+2.*BB*SUL),ABS(RR+TT*SUL))
      DO 23 I=1,2
      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(3,1) = S*S
      AA1(3,2) = -S
      AA(1) = ANG5
      AA(2) = ANG4
      AA(3) = ANG3
      CLOGR = ALOG(ABS((RR+TT*SUL)/(RR-TT*S)))
```

```
      SUSE = SMID2
      DELS = 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 18
   17 CONTINUE
      AA1(1,1) = S*S
      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
      AA(1) = ANG1
      AA(2) = ANG2
      AA(3) = ANG3P
      CLOGR = ALOG(ABS((RR+TT*S)/(RR+TT*SLL)))
      SUSE = SMID1
      DELS = S-SLL
      DELS2 = S*S-SLL*SLL
      DELS3 = S**3-SLL**3
      DELS4 = S**4-SLL**4
      DELS5 = S**5-SLL**5
      DELS6 = S**6-SLL**6
   18 CONTINUE
      CALL SIMEQ (AA1,3,AA,1,DET,IPIVOT,3,ISCALE)
      C1 = AA(1)
      C2 = AA(2)
      C = AA(3)
      CON0 = (C*TT*TT-C2*RR*TT+C1*RR*RR)/(TT**3)
      CON1 = (C2*TT-C1*RR)/(TT*TT)
      CON11 = 0.5*C1/TT
      CON2 = (C*TT*TT-C2*RR*TT+C1*RR*RR)/(TT**3)
      CON3 = (C*RR*TT*TT-C2*RR*RR*TT+C1*RR**3)/(TT**4)
      CON21 = C2/(2*TT)-C1*RR/(2*TT*TT)
      CON4 = (C*RR*RR*TT*TT-C2*RR**3*TT+C1*RR**4)/(TT**5)
      CON5 = 4.*C2*TT**3-4.*C1*RR*TT**2
      CON5 = CON5/(12*TT**4)
      CON6 = (-C*RR*TT*TT+C2*RR*RR*TT-C1*RR**3)/(TT**4)
```

```
      CON31 = (C*TT**3-C2*RR*TT**2+C1*RR**2*TT)/(2*TT**4)
      CON32 = C1/(4*TT)
      CON8 = 20.*(C*TT**4-C2*RR*TT**3+C1*RR**2*TT**2)/(60*TT**5)
      CON9 = (C*RR**2*TT**2-C2*RR**3*TT+C1*RR**4)/(TT**5)
      CON10 = C*RR**3*TT**2-C2*RR**4*TT+C1*RR**5
      CON10 = CON10/(TT**6)
      CON41 = (C2*TT-C1*RR)/(4*TT**2)
      CON42 = (-C*RR*TT**2+C2*RR**2*TT-C1*RR**3)/(2*TT**4)
      CONA = (C*RR**4*TT**2-C2*RR**5*TT+C1*RR**6)/(TT**7)
      CONB = (C2*TT-C1*RR)/(5*TT**2)
      CONC = (-C*RR*TT**2+C2*RR**2*TT-C1*RR**3)/(3*TT**4)
      COND = CONC*RR**2/(TT**2/3)
      CONE = (C*TT**2-C2*RR*TT+C1*RR**2)/(4*TT**3)
      CONF = C1/(6*TT)
      CONG = CONE*RR**2*2/(TT**2)
      IF (I.EQ.1)  GO TO 19
      IF (I.EQ.2.AND.(RR+TT*SUSE).GT.0.)  GO TO 19
      RTAN = RTAN-CON0*CLOGR+CON1*DELS+CON11*DELS2
      RSTAN = RSTAN-CON21*DELS2+CON2*DELS-CON3*CLOGR+C1*DELS3/(3*TT)
      RS2TAN = RS2TAN-CON4*CLOGR+CON5*DELS3+CON31*DELS2+CON32*DELS4+CON6
     1*DELS
      RS3TAN = RS3TAN-CON41*DELS4+CON8*DELS3+CON42*DELS2+CON9*DELS-CON10
     1*CLOGR+C1*DELS5/(5*TT)
      RS4TAN = RS4TAN-CONA*CLOGR+CONB*DELS5+CONE*DELS4+CONF*DELS6+CONC*D
     1ELS3+CONG*DELS2+COND*DELS
      GO TO 22
   19 CONTINUE
      RTAN = RTAN+CON0*CLOGR+CON1*DELS+CON11*DELS2
      RSTAN = RSTAN+CON21*DELS2+CON2*DELS-CON3*CLOGR+C1*DELS3/(3*TT)
      RS2TAN = RS2TAN+CON4*CLOGR+CON5*DELS3+CON31*DELS2+CON32*DELS4+CON6
     1*DELS
      RS3TAN = RS3TAN+CON41*DELS4+CON8*DELS3+CON42*DELS2+CON9*DELS-CON10
     1*CLOGR+C1*DELS5/(5*TT)
      RS4TAN = RS4TAN+CONA*CLOGR+CONB*DELS5+CONE*DELS4+CONF*DELS6+CONC*D
     1ELS3+CONG*DELS2+COND*DELS
      IF (I.EQ.1.AND.(RR+TT*SUSE).LT.0.)  GO TO 20
      GO TO 21
   20 RTAN = -RTAN
      RSTAN = -RSTAN
      RS2TAN = -RS2TAN
      RS3TAN = -RS3TAN
      RS4TAN = -RS4TAN
```

```
   21 CONTINUE
   22 CONTINUE
   23 CONTINUE
      GO TO 13
   24 CONTINUE
      RR = RRR
      S = SSS
      C = CCC
      RETURN

   25 FORMAT(30X,11H200 ENTERED)
   26 FORMAT(30X,43HONE OF THE ENDPOINTS HAS A NEGATIVE LOG ARG)
   27 FORMAT(30X,10H80 ENTERED)
      END
```

```
      SUBROUTINE LOGS (S,E,D,RELN,RESLN,RES2LN,RES3LN)
C
C     SUBROUTINE LOGS
C
C     CALCULATES INTEGRALS OF FORM S**N*ALOG(S*S+E*S+D) WITH
C     RESPECT TO S OVER LIMITS OF -S TO S FOR N=0,1,2,3.
C
C     S= PANEL SEGMENT HALFWIDTH
C     E,D ARE CALCULATED IN SUBROUTINE CONCAL
C     INTEGRAL RESULTS ARE RELN,RESLN,RES2LN,RES3LN
C
C     EVALUATION OF INTEGRALS PERFORMED USING MACSYMA ALGEBRAIC
C     MANIPULATION PROGRAM OF MIT PROJECT MAC
C     IF I=J INTEGRAL EVALUALTED AT APPROXIMATE ENDPOINTS,+-SAWAY
C
      REAL  LATB,LADB,L1,L2,L3,L4
      COMMON  /TELL/ TOL,TOL2
      RELN = 0.
      RESLN = 0.
      RES2LN = 0.
      RES3LN = 0.
      SS = S
      A = S*S+E*S+D
      B = S*S-E*S+D
      A = ABS(A)
      B = ABS(B)
      AA = ABS(A)
      BB = ABS(B)
      IF (AA.LE.0.000000001)  GO TO 6
      IF (BB.LE.0.000000001)  GO TO 6
    1 DISC = E*E-4*D
      DISQ = SQRT(ABS(DISC))
      DIS = E*E-2.*D
      DIS3 = E**3-3.*D*E
      DIS4 = (E*E-4.*D)*(E*E-D)
      DIS44 = E**4-4.*D*E*E+2.*D*D
      DIS5 = E**5-6.*D*E**3+8.*E*D**2
      LATB = ALOG(A*B)
      LADB = ALOG(A/B)
      IF (AA.LE.0.000000001)  S = SAWAY
      IF (BB.LE.0.000000001)  S = SAWAY
      RE = S*LATB+0.5*E*LADB
```

```
      ESP = E+2*S
      ESM = E-2*S
      IF (DISC)  4,3,2
    2 CONTINUE
      L1 = E-DISQ+2*S
      L2 = E+DISQ+2*S
      L3 = E-DISQ-2*S
      L4 = E+DISQ-2*S
      DIFFLN = ALOG(L1*L4/L2/L3)
      RELN = PE-4*S-0.5*DISQ*DIFFLN
      RESLN = 0.5*LADB*S**2+(0.25*E*DISC/DISQ)*DIFFLN
      RESLN = RESLN-0.25*DIS*LADB+E*S
      RES2LN = (S**3/3)*LATB-DIS4/(6*DISQ)*DIFFLN+LADB*(DIS3/6)-4*S**3/9
     1-6*DIS*S/9
      RES3LN = 0.25*S**4*LADB+DIS5/(8*DISQ)*DIFFLN-LADB*(DIS44/8)+E*S**3
     1/6+0.5*S*DIS3
      GO TO 5
    3 CLOGRT = ALOG(ESP/ESM)
      RELN = S*LATB-4*S+E*CLOGRT
      RESLN = 0.5*S**2*LADB-0.5*DIS*CLOGRT+E*S
      RES2LN = (S**3*LATB+DIS3*CLOGRT+(DIS4)*((1/ESP)-(1/ESM))-2*S*DIS)/
     13-4*(S**3)/9
      RES3LN = 0.25*S**4*LADB-DIS44*0.25*CLOGRT-.25*DIS5*(1/ESP-1/ESM)+E
     1*S**3/6+0.5*S*DIS3
      GO TO 5
    4 TNRAT = ATAN2(ESP,DISQ)-ATAN2(ESM,DISQ)
      RELN = PE-4*S-(DISC/DISQ)*TNRAT
      RESLN = 0.5*(S**2-0.5*DIS)*LADB+0.5*E*DISC/DISQ*TNRAT+E*S
      RES2LN = S**3/3*LATB+(DIS3/6)*LADB-(DIS4/(3*DISQ))*TNRAT-4*S**3/9-
     12*S*DIS/3
      RES3LN = (0.25*S**4-DIS44/8)*LADB+0.25*DIS5/DISQ*TNRAT+E*S**3/6+S*
     1DIS3/2
    5 CONTINUE
      GO TO 7
    6 CONTINUE
      SAWAY = S-TOL2
      A = SAWAY*SAWAY+E*SAWAY+D
      B = SAWAY*SAWAY-E*SAWAY+D
      A = ABS(A)
      B = ABS(B)
      GO TO 1
    7 CONTINUE
```

```
85                S = SS
                  RETURN
                  END
```

```
      SUBROUTINE DRACAL (I,J,YHH,ZHH,PPP,AINT)
C
C     SUBROUTINE DRACAL
C
C     TREFFTZ PLANE DRAG ANALYSIS ASSUMES PIECEWISE LINEARLY VARYING
C     SHED VORTICITY SHEET STRENGTH
C
C     CALCULATE INTEGRALS A THROUGH F FOR DRAG COEF CALCULATION
C
C     CALLS SUBROUTINES LOGS,SNTAN,CCAL,CONCAL
C
      DIMENSION  AINT(6)
      DIMENSION  YHH(1), ZHH(1), PPP(1)
      COMMON  /SEG/ SNN(51)
      PI = 4.*ATAN(1.)
      S = SNN(J)
      CALL CCAL (I,J,YHH,ZHH,PPP,S,AA,BB,DD,FF,GG,EE,AJ,AK,RR,TT,UU,WW)
      CALL CONCAL (AA,BB,FF,GG,S,A,B,C,D,F,G,CJ,CK,CL,CM,CN,CO,CP,1)
      S = SNN(I)
      CALL LOGS (S,CJ,F,RELN,RESLN,RES2LN,RES3LN)
      CALL SNTAN (S,C,BB,RR,TT,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)
      AAAAAA = A*RTAN+B*RSTAN+BB*RELN/4
      BBBBBB = 2.*(CL*RTAN+CM*RSTAN+CN*RS2TAN)+CO*RELN+CP*RESLN
      CCCCCC = A*RSTAN+B*RS2TAN+BB*RESLN/4
      DDDDDD = 2.*(CL*RSTAN+CM*RS2TAN+CN*RS3TAN)+CO*RESLN+CP*RES2LN
      EEEEEE = A*RS2TAN+B*RS3TAN+BB*RES2LN/4
      FFFFFF = 2.*(CL*RS2TAN+CM*RS3TAN+CN*RS4TAN)+CO*RES2LN+CP*RES3LN
      CALL LOGS (S,CK,G,RELN,RESLN,RES2LN,RES3LN)
      CALL SNTAN (S,D,BB,RR,TT,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)
      AAAAAA = AAAAAA-A*RTAN-B*RSTAN-BB*RELN/4
      BBBBBB = BBBBBB-2.*(CL*RTAN+CM*RSTAN+CN*RS2TAN)-CO*RELN-CP*RESLN
      CCCCCC = CCCCCC-A*RSTAN-B*RS2TAN-BB*RESLN/4
      DDDDDD = DDDDDD-2.*(CL*RSTAN+CM*RS2TAN+CN*RS3TAN)-CO*RESLN-CP*RES2
     1LN
      EEEEEE = EEEEEE-A*RS2TAN-B*RS3TAN-BB*RES2LN/4
      FFFFFF = FFFFFF-2.*(CL*RS2TAN+CM*RS3TAN+CN*RS4TAN)-CO*RES2LN-CP*RE
     1S3LN
      S = SNN(J)
      CALL CONCAL (DD,EE,AJ,AK,S,A,B,C,D,F,G,CJ,CK,CL,CM,CN,CO,CP,2)
      S = SNN(I)
      CALL LOGS (S,CJ,F,RELN,RESLN,RES2LN,RES3LN)
      CALL SNTAN (S,C,EE,UU,WW,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)
```

```
      AAAAAA = AAAAAA-A*RTAN-B*RSTAN-EE*RELN/4
      BBBBBB = BBBBBB-2.*(CL*RTAN+CM*RSTAN+CN*RS2TAN)-CO*RELN-CP*RESLN
      CCCCCC = CCCCCC-A*RSTAN-B*RS2TAN-EE*RESLN/4
      DDDDDD = DDDDDD-2.*(CL*RSTAN+CM*RS2TAN+CN*RS3TAN)-CO*RESLN-CP*RES2
     1LN
      EEEEEE = EEEEEE-A*RS2TAN-B*RS3TAN-EE*RES2LN/4
      FFFFFF = FFFFFF-2.*(CL*RS2TAN+CM*RS3TAN+CN*RS4TAN)-CO*RES2LN-CP*RE
     1S3LN
      CALL LOGS (S,CK,G,RELN,RESLN,RES2LN,RES3LN)
      CALL SNTAN (S,D,EE,UU,WW,RTAN,RSTAN,RS2TAN,RS3TAN,RS4TAN)
      AAAAAA = AAAAAA+A*RTAN+B*RSTAN+EE*RELN/4
      BBBBBB = BBBBBB+2.*(CL*RTAN+CM*RSTAN+CN*RS2TAN)+CO*RELN+CP*RESLN
      CCCCCC = CCCCCC+A*RSTAN+B*RS2TAN+EE*RESLN/4
      DDDDDD = DDDDDD+2.*(CL*RSTAN+CM*RS2TAN+CN*RS3TAN)+CO*RESLN+CP*RES2
     1LN
      EEEEEE = EEEEEE+A*RS2TAN+B*RS3TAN+EE*RES2LN/4
      FFFFFF = FFFFFF+2.*(CL*RS2TAN+CM*RS3TAN+CN*RS4TAN)+CO*RES2LN+CP*RE
     1S3LN
      SK = SMN(J)
      AAAAAA = AAAAAA/PI
      BBBBBB = BBBBBB/(2.*PI*SK)+(2.*S/PI)*(BB-EE)
      CCCCCC = CCCCCC/PI
      DDDDDD = DDDDDD/(2.*PI*SK)
      EEEEEE = EEEEEE/(2*PI*S)
      FFFFFF = 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) = EEEEEE
      AINT(6) = FFFFFF
      RETURN
      END
```

```
      SUBROUTINE SIMEQ (A,N,B,M,DETERM,IPIVOT,NMAX,ISCALE)
C     SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
C     *** DOCUMENT DATE 08-01-68   SUBROUTINE REVISED 08-01-68 *********
C
      DIMENSION  IPIVOT(N), A(NMAX,N), B(NMAX,M)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX,T,SWAP)
C
C     INITIALIZATION
C
    1 ISCALE = 0
      R1 = 10.0**100
      R2 = 1.0/R1
      DETERM = 1.0
      DO 2 J=1,N
    2 IPIVOT(J) = 0
      DO 38 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
      AMAX = 0.0
      DO 7 J=1,N
      IF (IPIVOT(J)-1)  3,7,3
    3 DO 6 K=1,N
      IF (IPIVOT(K)-1)  4,6,39
    4 IF (ABS(AMAX)-ABS(A(J,K)))  5,6,6
    5 IROW = J
      ICOLUM = K
      AMAX = A(J,K)
    6 CONTINUE
    7 CONTINUE
      IF (AMAX)  9,8,9
    8 DETERM = 0.0
      ISCALE = 0
      GO TO 39
    9 IPIVOT(ICOLUM) = IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW-ICOLUM)  10,14,10
   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 = B(IROW,L)
          B(IROW,L) = B(ICOLUM,L)
       13 B(ICOLUM,L) = SWAP
       14 PIVOT = A(ICOLUM,ICOLUM)
          IF (PIVOT)  15,8,15
C
C         SCALE THE DETERMINANT
C
       15 PIVOTI = PIVOT
          IF (ABS(DETERM)-R1)  18,16,16
       16 DETERM = DETERM/R1
          ISCALE = ISCALE+1
          IF (ABS(DETERM)-R1)  21,17,17
       17 DETERM = DETERM/R1
          ISCALE = ISCALE+1
          GO TO 21
       18 IF (ABS(DETERM)-R2)  19,19,21
       19 DETERM = DETERM*R1
          ISCALE = ISCALE-1
          IF (ABS(DETERM)-R2)  20,20,21
       20 DETERM = DETERM*R1
          ISCALE = ISCALE-1
       21 IF (ABS(PIVOTI)-R1)  24,22,22
       22 PIVOTI = PIVOTI/R1
          ISCALE = ISCALE+1
          IF (ABS(PIVOTI)-R1)  27,23,23
       23 PIVOTI = PIVOTI/R1
          ISCALE = ISCALE+1
          GO TO 27
       24 IF (ABS(PIVOTI)-R2)  25,25,27
       25 PIVOTI = PIVOTI*R1
          ISCALE = ISCALE-1
          IF (ABS(PIVOTI)-R2)  26,26,27
       26 PIVOTI = PIVOTI*R1
          ISCALE = ISCALE-1
       27 DETERM = DETERM*PIVOTI
C
C         DIVIDE PIVOT ROW BY PIVOT ELEMENT
```

```
      C
            DO 29 L=1,N
            IF (IPIVOT(L)-1)  28,29,39
         28 A(ICOLUM,L) = A(ICOLUM,L)/PIVOT
         29 CONTINUE
            IF (M)  32,32,30
         30 DO 31 L=1,M
         31 B(ICOLUM,L) = B(ICOLUM,L)/PIVOT
      C
      C     REDUCE NON-PIVOT ROWS
      C
         32 DO 38 L1=1,N
            IF (L1-ICOLUM)  33,38,33
         33 T = A(L1,ICOLUM)
            DO 35 L=1,N
            IF (IPIVOT(L)-1)  34,35,39
         34 A(L1,L) = A(L1,L)-A(ICOLUM,L)*T
         35 CONTINUE
            IF (M)  38,38,36
         36 DO 37 L=1,M
         37 B(L1,L) = B(L1,L)-B(ICOLUM,L)*T
         38 CONTINUE
         39 RETURN
            END
```

REFERENCES

1. Whitcomb, Richard T.: A Design Approach and Selected Wind-Tunnel Results at High Subsonic Speeds for Wing-Tip Mounted Winglets. NASA TN D-8260, 1976.

2. Lange, R. H.; Cahill, J. F.; Bradley, E. S.; Eudaily, R. R.; Jenness, C. M.; and MacWilkinson, D. G.: Feasibility Study of the Transonic Biplane Concept for Transonic Aircraft Application. NASA CR-132462, June 1974.

3. Wolkovitch, Julian: Joined Wing Aircraft. U.S. Patent No. 3,942,747, issue date March 9, 1976. Other patents pending.

4. Kuhlman, John M.: Numerical Optimization Techniques for Bound Circulation Distribution for Minimum Induced Drag of Nonplanar Wings: Basic Formulations. NASA CR-3154, June 1979.

5. Karamcheti, Krishnamurty: Principles of Ideal-Fluid Aerodynamics. Wiley, 1966.

6. Munk, Max M.: The Minimum Induced Drag of Airfoils. NACA TM No. 856, 1938.

7. Loth, John L.; and Boyle, Robert E.: Optimum Loading on Nonplanar Wings at Minimum Induced Drag. Aerospace Engineering TR-19 (Contract N0014-68-A-1512), West Virginia Univ., Aug. 1969. (Available from DTIC as AD 704 502.)

8. Kuhlman, John M.: Integration of Advanced Panel Trefftz Plane Drag Minimization Technique in Vortex Lattice Wing Design Code. Published in ASME Book No. G00147, pp. 201-210, 1979; Presented at ASME/AIAA/SAE Joint Symposium on Aerodynamics of Transportation (Niagara Falls, N.Y.), June 18-20, 1979.

9. Lamar, John E.: A Vortex-Lattice Method for the Mean Camber Shapes of Trimmed Noncoplanar Planforms with Minimum Vortex Drag. NASA TN D-8090, June 1976.

10. Kuhlman, John M.: Higher Order Farfield Drag Minimization for a Subcritical Wing Design Code. J. Aircraft, Vol. 17, No. 9, Sept. 1980, pp. 648-655.

11. Bogen, Richard; Golden, Jeffrey; Genesereth, Michael; and Doohovskoy, Alexander: MACSYMA Reference Manual, Version 8, The Math Lab Group, Project MAC, Massachusetts Institute of Technology, Nov. 1975.

12. Lundry, John L.; and Lissaman, P. B. S.: A Numerical Solution for the Minimum Induced Drag of Nonplanar Wings. J. Aircraft, Vol. 5, No. 1, 1968.

13. Cone, Clarence D., Jr.: The Theory of Induced Lift and Minimum Induced Drag for Nonplanar Lifting Systems. NASA TR-139, 1962.

14. Lundry, John L.: A Numerical Solution for the Minimum Induced Drag, and the Corresponding Loading, of Nonplanar Wings. NASA CR-1218, 1968.

15. Lundry, John L.: A Numerical Solution for the Minimum Induced Drag, and the Corresponding Loading of Nonplanar Wings. Part I - User's Manual for Computer Program 55VD; Part II - Descriptive Manual for Computer Program 55VD. Supplement to NASA CR-1218, Nov. 1968.

16. Letcher, John S., Jr.: V-Wings and Diamond Ring-Wings of Minimum Induced Drag. J. Aircraft, Vol. 9, No. 8, Aug. 1972, pp. 605-607.

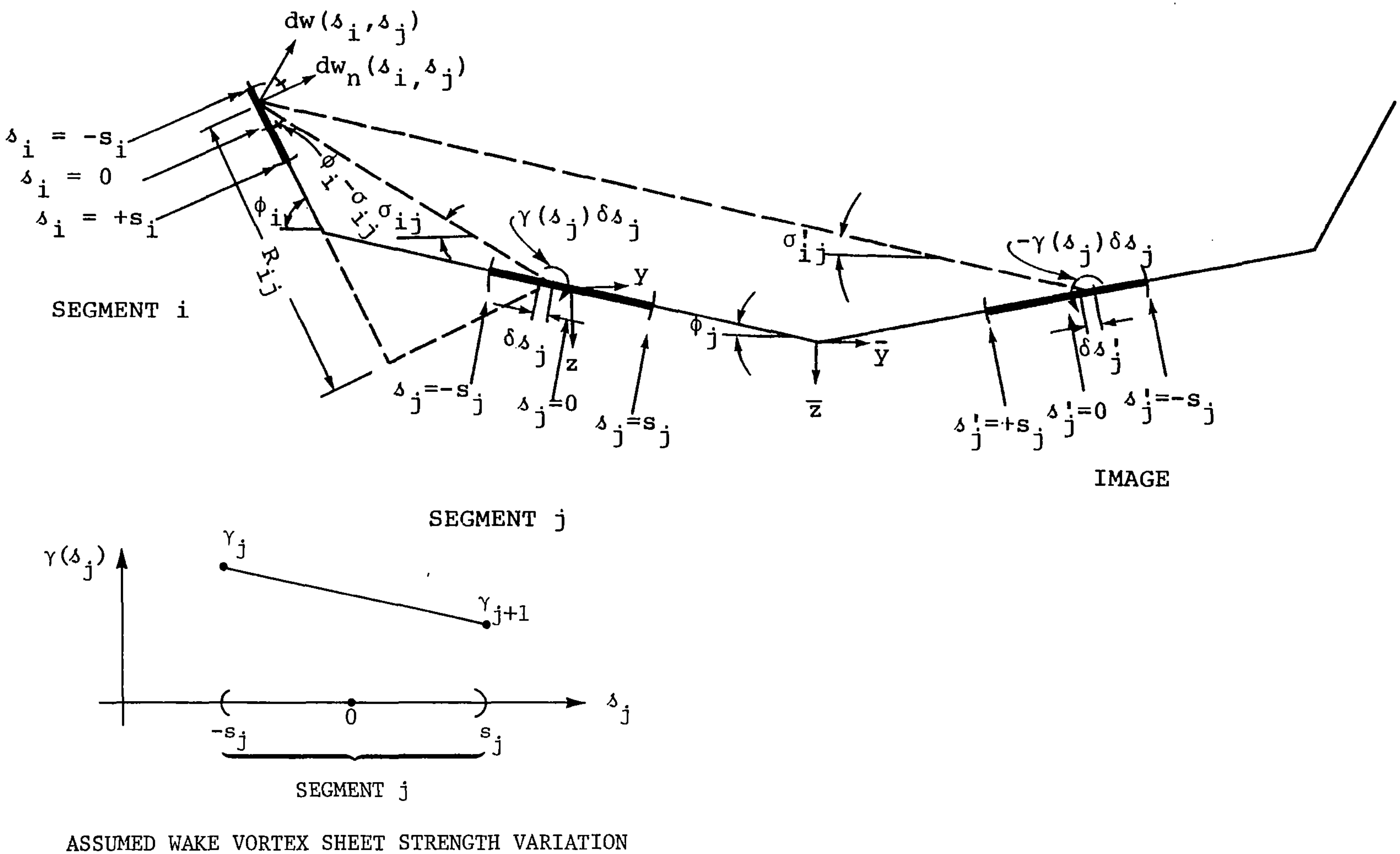


Figure 1. Trefftz plane geometry used in the present method.

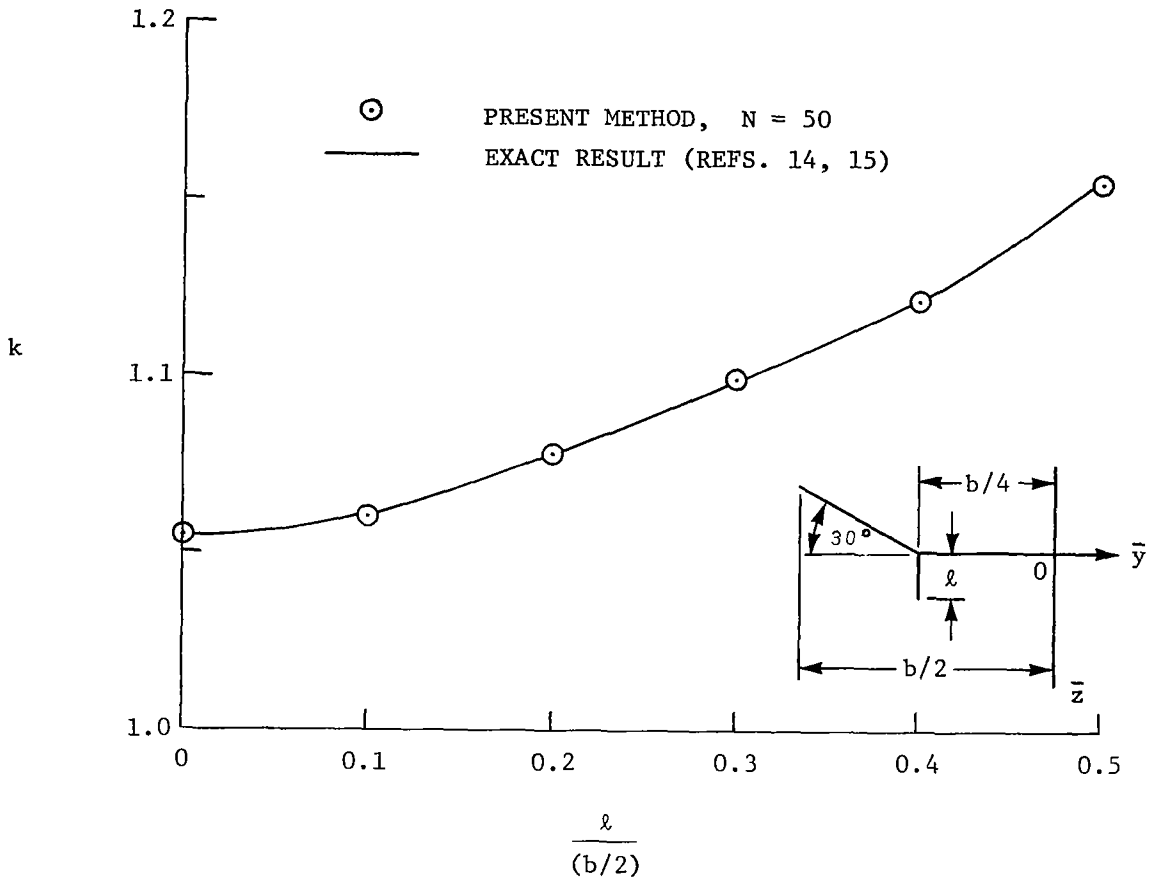


Figure 2. Induced drag efficiency for nonplanar wing with vertical fences of variable size.

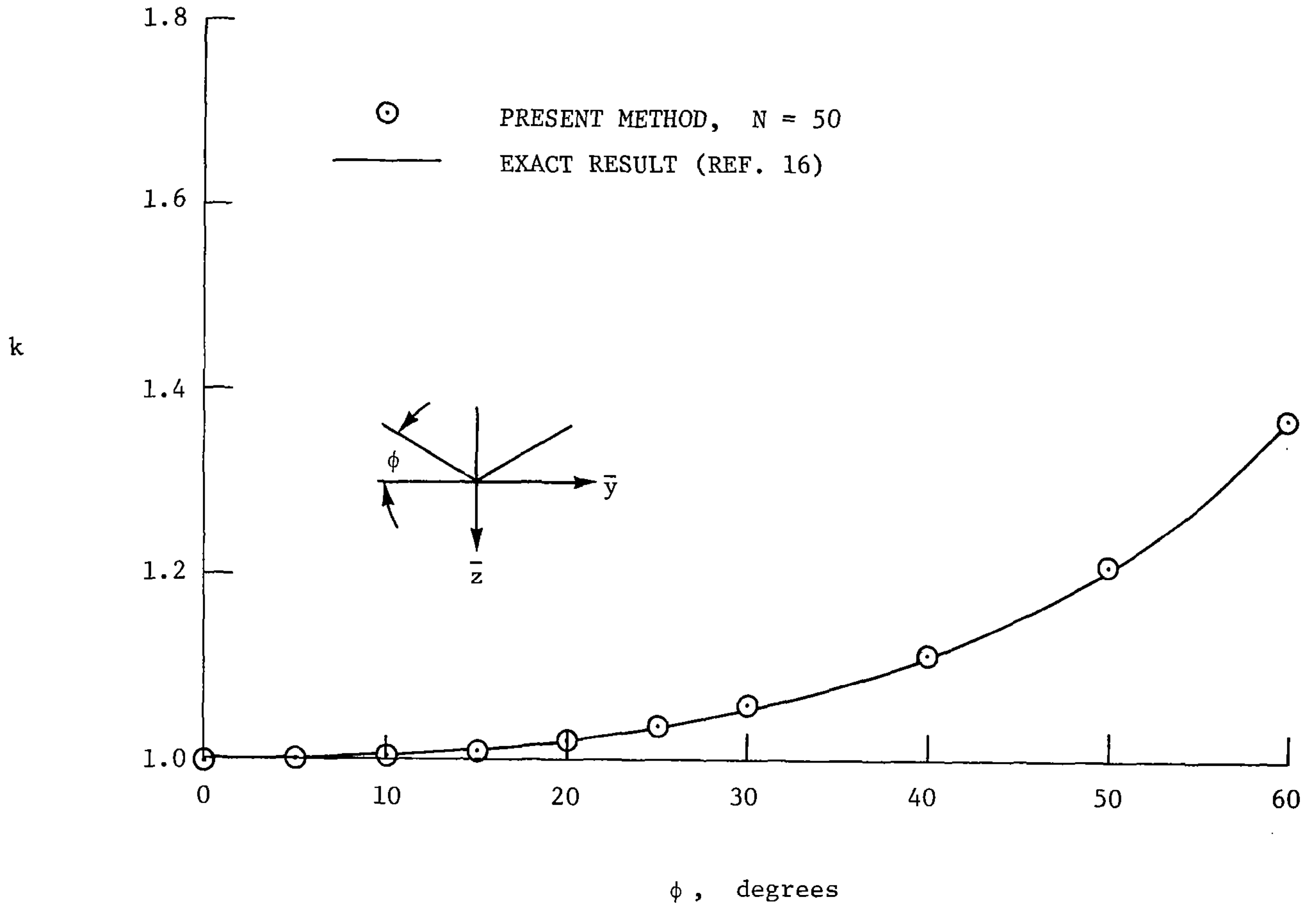


Figure 3. Induced drag efficiency for a series of vee wings.

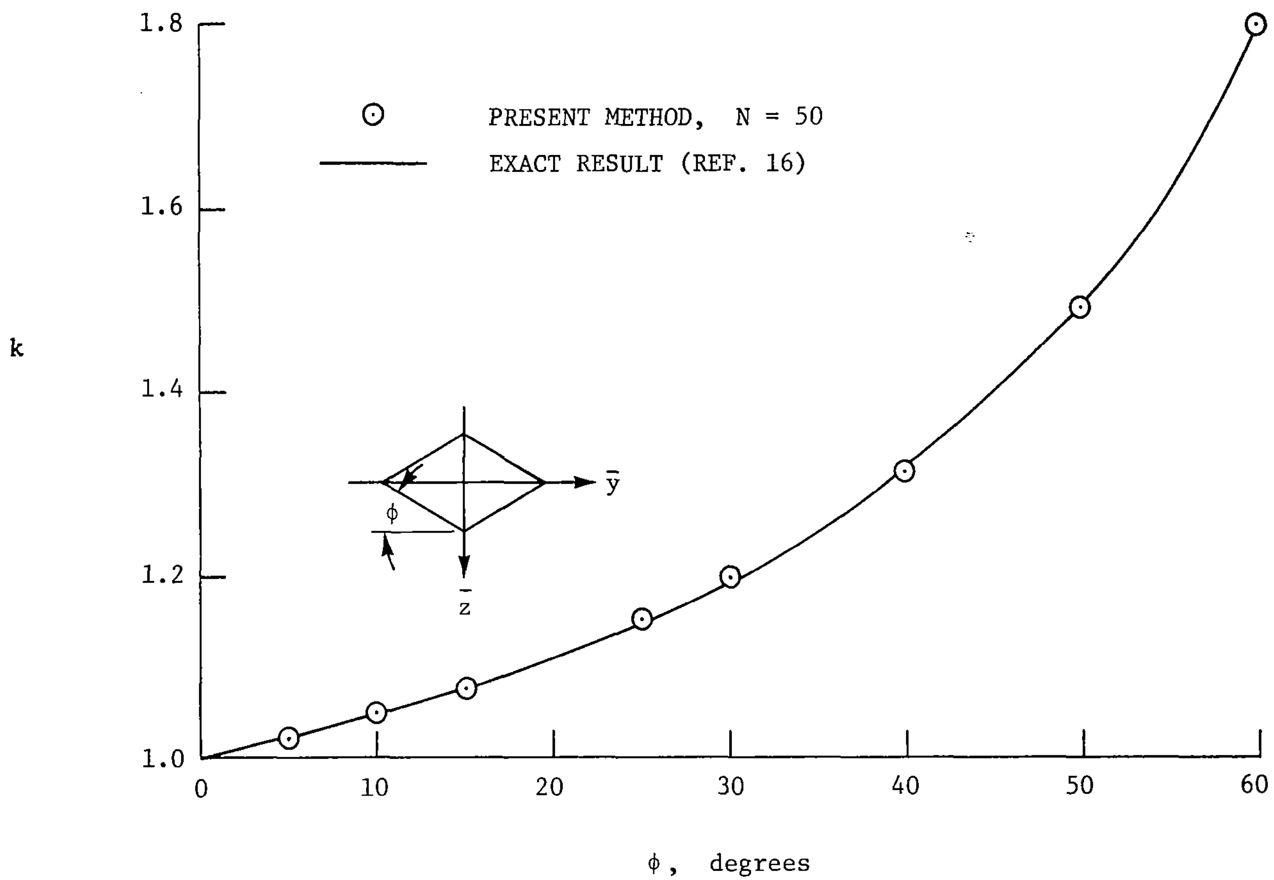


Figure 4: Induced drag efficiency for a series of diamond wings.

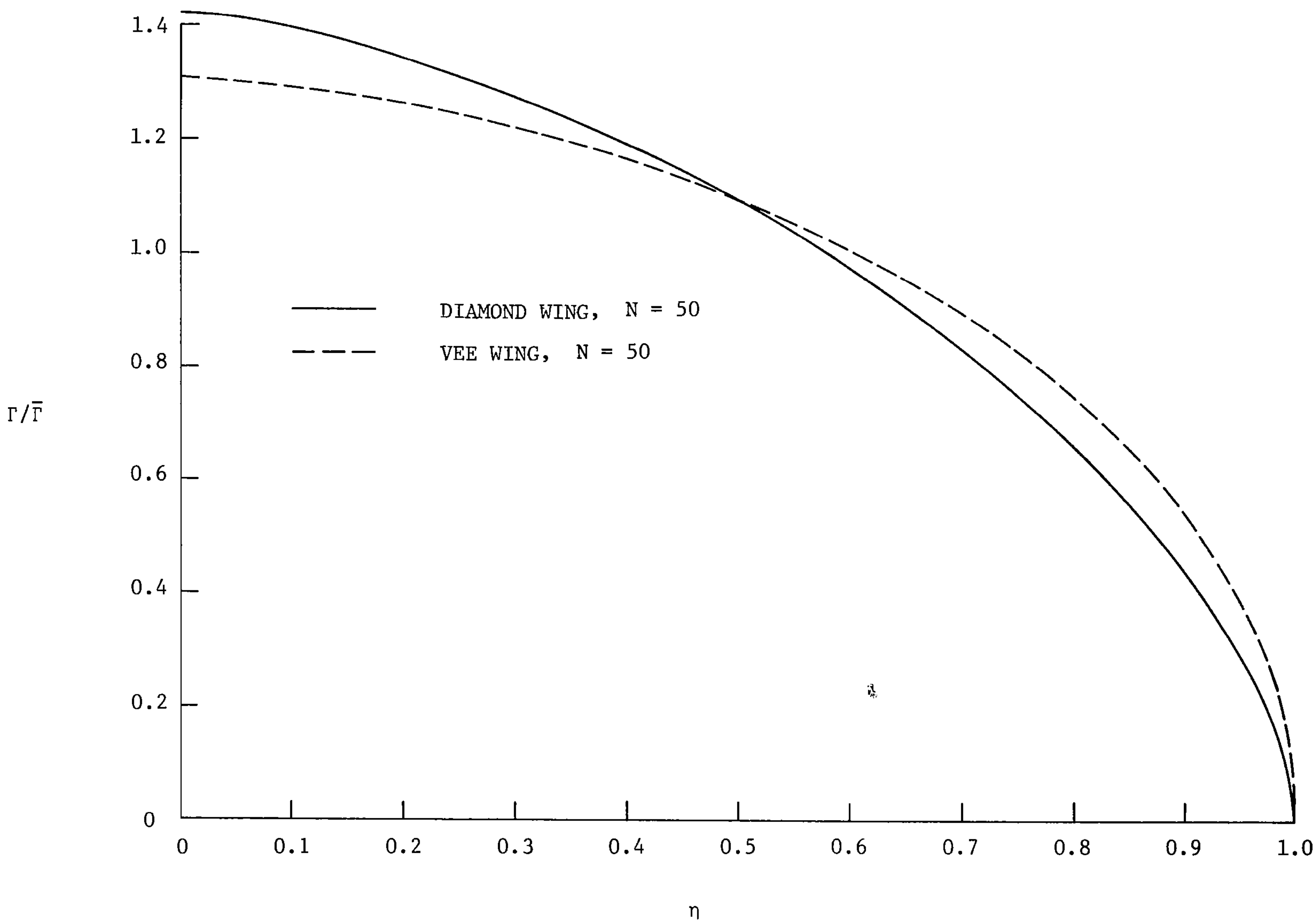


Figure 5. Comparison of bound circulation distributions for vee wing and diamond wing, ϕ = 30°, using present theory.

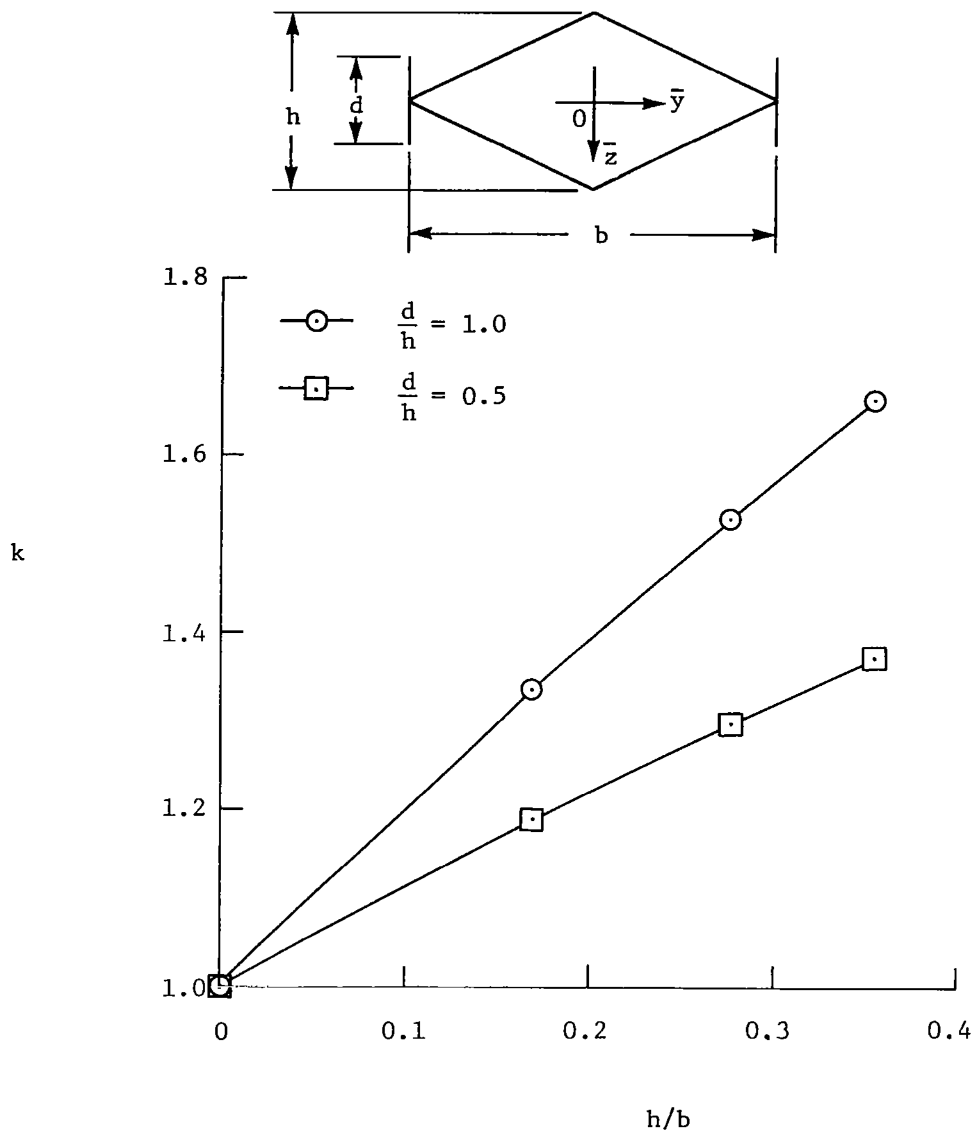


Figure 6. Induced drag efficiency for diamond wing fitted with end plates; N = 30 cosine segment spacing, and $\phi = 89.5°$ on end plates for all results.

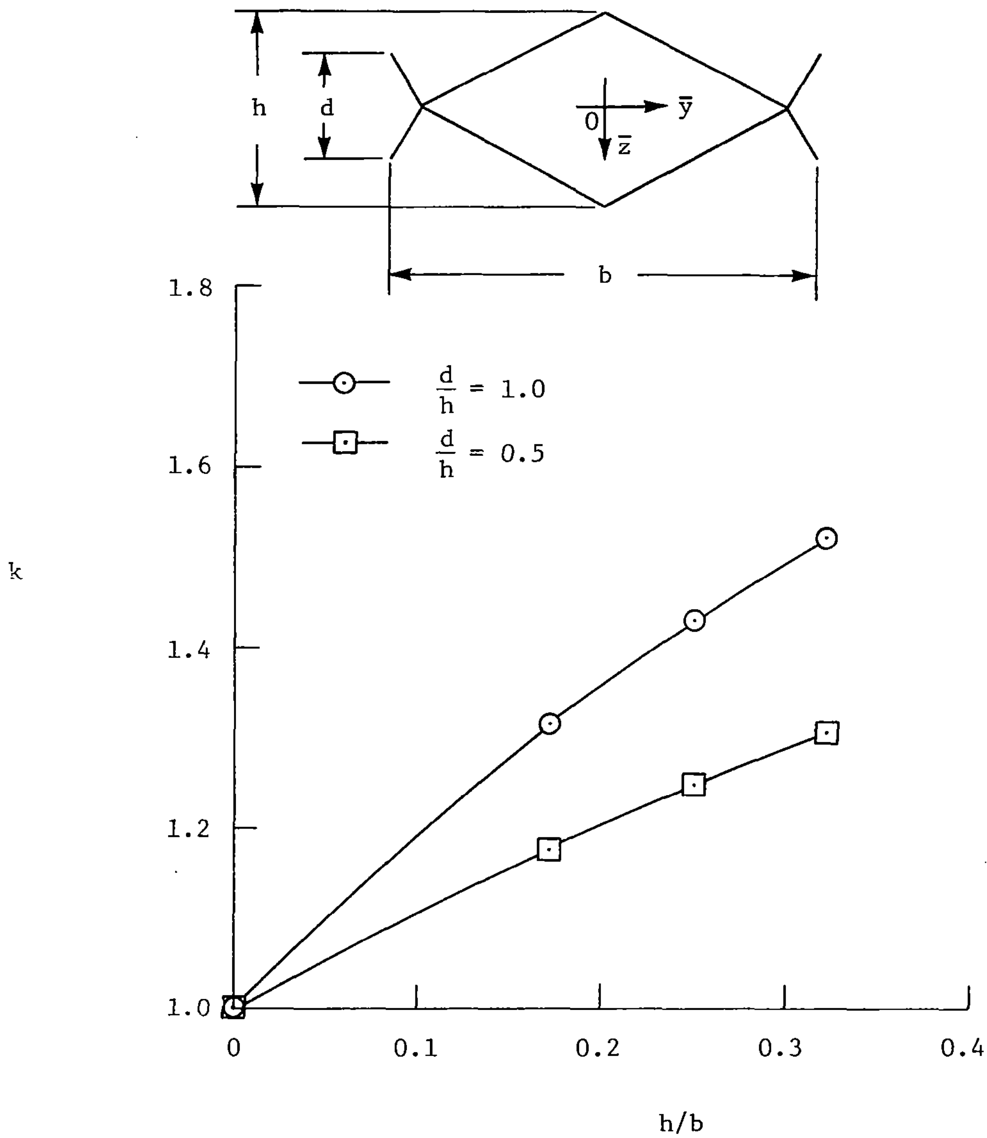


Figure 7. Induced drag efficiency for diamond wing fitted with winglets. Winglets perpendicular to wings, N = 30, and cosine segment spacing for all results.

1. Report No. NASA CR-3458	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Numerical Optimization Techniques for Bound Circulation Distribution for Minimum Induced Drag of Nonplanar Wings: Computer Program Documentation		5. Report Date August 1981 6. Performing Organization Code
7. Author(s) John M. Kuhlman, Principal Investigator, and Tzuchun Jeffrey Ku		8. Performing Organization Report No.
9. Performing Organization Name and Address Old Dominion University Research Foundation P.O. Box 6369 Norfolk, Virginia 23508		10. Work Unit No. 11. Contract or Grant No. NSG-1357
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Contractor Report March—September 1979 14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitors: John B. Peterson, Jr., and John E. Lamar.
Topical Report

16. 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.

17. Key Words (Suggested by Author(s))		18. Distribution Statement	
Subsonic Aerodynamics Induced Drag Minimization Nonplanar Configurations FORTRAN Computer Program Users Manual		FEDD Distribution Subject Category 02	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 77	22. Price

Available: NASA's Industrial Applications Centers

NASA-Langley, 1981