N73-11999

NASA CONTRACTOR REPORT

NASA CR-2157

NASA CR-2157

CALCULATIVE TECHNIQUES FOR TRANSONIC FLOWS ABOUT CERTAIN CLASSES OF WING-BODY COMBINATIONS - PHASE II

by Stephen S. Stahara and John R. Spreiter

Prepared by
NIELSEN ENGINEERING & RESEARCH, INC.
Mountain View, Calif. 94040
for Ames Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • DECEMBER 1972

1. Report No. NASA CR-2157	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Calculative Techniques for Transonic Flows About Certain Classes of Wing-Body Combinations - Phase II		5. Report Date December 1972
		6. Performing Organization Code
7. Author(s) Stephen S. Stahara and John R. Spreiter		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Nielsen Engineering and Research, Inc. 850 Maude Avenue Mountain View, CA 94040		11. Contract or Grant No. NAS 2-6259
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Theoretical analysis and associated computer programs were developed for predicting properties of transonic flows about certain classes of wing-body combinations. The procedures used are based on the transonic equivalence rule and employ either an arbitrarily-specified solution or the local linerization method for determining the nonlifting transonic flow about the equivalent body. The class of wind planform shapes include wings having sweptback trailing edges and finite tip chord. Theoretical results are presented for surface and flow-field pressure distributions for both nonlifting and lifting situations at Mach number one.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Transonic Flow Aerodynamics Wing-Body Combinations Transonic Equivalence Rule Local Linearization	UNCLASSIFIED-UNLIMITED

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	114	3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

CALCULATIVE TECHNIQUES FOR TRANSONIC FLOWS ABOUT CERTAIN CLASSES OF WING-BODY COMBINATIONS - PHASE II

By Stephen S. Stahara and John R. Spreiter*
Nielsen Engineering & Research, Inc.

SUMMARY

Theoretical analysis and the development of associated computer programs were carried out for the purpose of developing computational techniques for predicting properties of transonic flows about certain classes of wing-body combinations. The procedures used are based on the transonic equivalence rule and employ either an arbitrarily-specified solution or the local linearization method for determining the nonlifting transonic flow about the equivalent body. Theoretical results obtained by using the local linearization method are presented for surface and flow-field pressure distributions for certain members of the general classes of configurations studied, for both nonlifting and lifting situations, at $M_\infty = 1$.

The computational programs developed under this report are documented and presented in a general user's manual included as part of the report.

INTRODUCTION

Stimulated by the need for accurate prediction of transonic flows about realistic aircraft configurations, recent research is producing significant advances in the ability to predict theoretically both two and three-dimensional transonic flows about a wide variety of aerodynamic shapes. While current emphasis seems to be placed on the development of numerical techniques (refs. 1, 2, 3, 4, 5), it has become clear that, although significant accuracy limitations need not exist for advanced computer programs, these techniques do have cost limitations with regard to both accuracy and the use of alternate methods. Consequently, in

*Professor, Departments of Applied Mechanics and Aeronautics and Astronautics, Stanford University, Stanford, California. (Consultant at *Nielsen Engineering & Research, Inc.*)

order to enhance these computational efforts, the parallel development of proven analytical and analytic/numeric methods to provide accurate first approximations, for example, in the systematic study of a large number of configurations, is clearly warranted.

Previous investigations by Nielsen Engineering & Research, Inc. (NEAR) in reference 6, where the local linearization method and the transonic equivalence rule were applied to predict surface and flow-field properties of several general classes of axisymmetric and nonaxisymmetric bodies for both lifting and nonlifting situations, and in reference 7, where those results were extended to include several classes of wing-body combinations, have demonstrated the effectiveness of such a combined approach.

While the ultimate goal of the present investigation is to develop computational techniques for the prediction of the flow field, pressure distribution, and aerodynamic characteristics of three-dimensional, lifting, wing-body combinations, the purposes of this study are to extend the results of reference 7 to include a more general class of wing planform shapes, specifically, wings having (1) sweptback trailing edges, and (2) finite tip chords. In addition, the computer programs developed in reference 7 were to be further enhanced with regard to minimization of computational time and applicability to wider classes of equivalent body shapes and equivalent body transonic solutions.

LIST OF SYMBOLS

a	major axis of elliptic cross section of indented body
a_{sb}	major axis of elliptic cross section of smooth (non-indented) body
AR	aspect ratio
b	minor axis of elliptic cross section of indented body
c	equal to $\sqrt{a^2 - b^2}$
c_w	wing chord
C	Euler's constant

$C_{D_{eb}}$	drag coefficient of equivalent body of revolution, $D_{eb}/q_\infty S_m$ eq. (22)
C_{D_t}	total drag coefficient, $D_t/q_\infty S_m$
$C_{D_{\alpha=o}}$	drag coefficient at zero lift, $D_{\alpha=o}/q_\infty S_m$
C_L	lift coefficient, $L/q_\infty S_m$
C_p	pressure coefficient, $(p - p_\infty)/\frac{1}{2}\rho_\infty U_\infty^2$
$C_{p_{eb}}$	pressure coefficient due to equivalent body of revolution
C_m	pitching moment coefficient about nose, $M_y/q_\infty S_m \ell$ (positive nose-up)
c_{tip}	wing tip chord
c_{R_t}	wing root chord
D	maximum diameter of equivalent body of revolution
D_{eb}	drag of equivalent body of revolution
D_i	drag due to lift
D_t	total drag
$D_{\alpha=o}$	total drag at zero lift
k	equal to $M_\infty^2(\gamma + 1)/U_\infty$
K	complete elliptic integral of first kind
ℓ	complete body length
L	lift
m	exponent describing wing ordinates, eqs. (8), (9)
$M_{cr,\ell}$	lower critical Mach number on equivalent body of revolution
$M_{cr,u}$	upper critical Mach number on equivalent body of revolution
M_y	pitching moment about nose, positive nose-up

M_∞	free-stream Mach number
n	exponent describing equivalent body ordinates and related to location of point of maximum thickness, eqs. (11), (12), (13), (14)
p_∞	free-stream pressure
q_1, q_2, q_3 q_4, q_5, q_6	quantities defined by eqs. (55), (56), (57)
q_∞	free-stream dynamic pressure
r	radial distance in crossflow plane, $\sqrt{y^2 + z^2}$
R_b	radius of indented body of revolution
R_{eb}	radius of equivalent body of revolution
R_1	radius of circular body in transformed (σ_1) plane, eq. (33)
s	semispan of wing which, depending on axial location, represents either leading ($s = s_\ell$) or trailing ($s = s_t$) edge
s_ℓ	semispan of wing leading edge
s_t	semispan of wing trailing edge
s_1	semispan of wing in transformed (σ_1) plane, eq. (34)
s_{1_ℓ}	semispan of wing leading edge in transformed (σ_1) plane
s_{1_t}	semispan of wing trailing edge in transformed (σ_1) plane
S_b	area distribution of indented body of revolution
S_{eb}	area distribution of equivalent body of revolution
S_m	maximum area distribution of equivalent body of revolution, $\pi D^2/4$
TR	wing planform taper ratio, C_{tip}/C_{R_t}
u, v, w	perturbation velocity components parallel to the x, y, z axes, respectively

u_B, v_B, w_B	perturbation velocity components associated with solution for transonic flow about equivalent body of revolution
$u_{2,B}, v_{2,B}, w_{2,B}$	perturbation velocity components associated with two-dimensional incompressible solution of expansion or contraction of equivalent cross section in crossflow plane
$u_{2,t}, v_{2,t}, w_{2,t}$	perturbation velocity components associated with two-dimensional incompressible solution of expanding or contracting cross section in crossflow plane
$u_{2,\alpha}, v_{2,\alpha}, w_{2,\alpha}$	perturbation velocity components associated with two-dimensional incompressible solution of translating cross section in crossflow plane
U_∞	free-stream velocity
$W_{2,t}$	complex potential describing two-dimensional incompressible flow about expanding or contracting cross section in crossflow plane
$W_{2,\alpha}$	complex potential describing two-dimensional incompressible flow about translating cross section in crossflow plane
x,y,z	body-fixed Cartesian coordinate system with x axis direction rearward and aligned with longitudinal axis of body, y axis directed to the right facing forward, and z axis directed vertically upward
x_s	location of point closest to origin where $S''_{eb}(x) = 0$
X_b	axial location of body base
$X_{r\ell e}$	axial location of wing leading edge root chord
$X_{r\ell e_1}$	axial location of point where wing leading edge pierces body surface
X_{rte}	axial location of wing trailing-edge root chord
X_{rte_1}	axial location of point where wing trailing edge pierces body surface
X_{sm_1}	axial location of wing tip chord leading edge
X_{sm_2}	axial location of wing tip chord trailing edge
$\bar{x}$	axial distance from wing leading edge

$\bar{y}$	lateral distance from wing leading edge
z_w	wing ordinates, eqs. (8), (9)
α	angle of attack
$\beta_{\ell e}$	wing leading-edge sweep angle
β_{te}	wing trailing-edge sweep angle
γ	ratio of specific heats
θ	polar angle in crossflow plane
λ	ratio of major to minor axes of elliptic cross section, a/b
ξ,ξ_1	dummy variables
ρ_∞	free-stream density
σ	complex variable in crossflow plane, $y + iz$
σ_1	complex variable in transformed crossflow plane, $y_1 + iz_1$
τ_{eb}	thickness ratio of equivalent body of revolution, D/ℓ
τ_w	thickness-to-chord ratio of wing profile, eq. (10)
ϕ	perturbation velocity potential
ϕ_B	perturbation velocity potential associated with transonic flow about equivalent body of revolution
ϕ_2	perturbation velocity potential associated with two-dimensional incompressible solutions to translation and growth of cross section in crossflow plane
$\phi_{2,B}$	perturbation velocity potential associated with two-dimensional incompressible solution for expansion or contraction of equivalent cross section in crossflow plane
$\phi_{2,t}$	perturbation velocity potential associated with two-dimensional incompressible solution for expansion or contraction of cross section in crossflow plane
$\phi_{2,\alpha}$	perturbation velocity potential associated with two-dimensional incompressible solution for translation of cross section in crossflow plane

ANALYSIS

General Considerations

Because the current work is an extension of that of reference 7, the basic theory and equations used are discussed in depth in that reference and their derivation will not be repeated here. For convenience, however, those points relevant to the present work will be outlined.

The coordinate system used for all of the three-dimensional flows considered herein is a body-fixed Cartesian system centered at the body nose with the x axis directed rearward and aligned with the longitudinal axis of the body, the y axis directed to the right, facing forward, and the z axis directed vertically upward, as shown in figure 1. For lifting situations, the free-stream direction is taken to be inclined at any arbitrary small angle α to the x axis and confined to the x-z plane so that there is no sideslip. The governing partial differential equation for the perturbation potential ϕ is given by

$$(1 - M_\infty^2)\phi_{xx} + \phi_{yy} + \phi_{zz} = \frac{M_\infty^2(\gamma + 1)}{U_\infty} \phi_x \phi_{xx} \tag{1}$$

where M_∞ is the free-stream Mach number, γ the ratio of specific heats, and U_∞ the free-stream velocity. The pressure coefficient C_p in the above reference frame is given by

$$C_p = -\frac{2}{U_\infty}(\phi_x + \alpha\phi_z) - \frac{1}{U_\infty^2}(\phi_y^2 + \phi_z^2) \tag{2}$$

The transonic equivalence rule enables the perturbation potential ϕ to be expressed in the form

$$\phi = \phi_{2,\alpha} + \phi_{2,t} - \phi_{2,B} + \phi_B \tag{3}$$

where each of the individual components has the meaning indicated in figure 1. Since $\phi_{2,\alpha}$, $\phi_{2,t}$, and $\phi_{2,B}$ satisfy the two dimensional incompressible Laplace equation

$$(\phi_{2,i})_{yy} + (\phi_{2,i})_{zz} = 0 \tag{4}$$

(where the subscript i depends upon the particular potential in question), they are independent of Mach number. Hence, the only portion of the solution dependent upon M_∞ is ϕ_B and this term represents the solution to the full transonic equation (1) for the nonlifting flow about the equivalent body of revolution. Because the equivalence rule places no essential restrictions on the methods of calculating ϕ_B, its solution may be determined in a variety of ways. For example, it can be given by an exact numerical solution, by experimental data, by an approximate analytic solution, or by a combined analytic/numeric solution such as the local linearization method. One of the tasks of the present work is to extend the applicability of the computer programs developed in reference 7 to include general, arbitrarily-specified solutions for ϕ_B, and the method of doing this is detailed in the user's manual. If the local linearization method is used to determine ϕ_B, or more conveniently, $u_B = (\phi_B)_x$, then one of the following set of three first-order nonlinear differential equations must be integrated according to whether $M_\infty \approx 1$, $M_\infty < M_{cr,\ell}$, or $M_{cr,u} < M_\infty$, where $M_{cr,\ell}$, $M_{cr,u}$ are the lower and upper critical Mach numbers, respectively.

Thus, for accelerating flows with $M_\infty \approx 1$

$$\frac{d}{dx}\left(\frac{u_B}{U_\infty}\right) = \frac{S'_{eb}(x)S''_{eb}(x)}{4\pi S_{eb}(x)} + \exp\left\{\frac{4\pi}{S''_{eb}(x)}\left[\frac{u_B}{U_\infty} + \frac{M_\infty^2 - 1}{M_\infty^2(\gamma + 1)}\right.\right.$$

$$\left.\left. - \frac{S''_{eb}(x)}{4\pi}\ln\frac{M_\infty^2(\gamma + 1)S_{eb}(x)e^C}{4\pi x} - \frac{1}{4\pi}\int_0^x \frac{S''_{eb}(x) - S''_{eb}(\xi)}{x - \xi}\,d\xi\right]\right\} \tag{5}$$

for purely subsonic flows ($M_\infty < M_{cr,\ell}$)

$$\frac{d}{dx}\left(\frac{u_B}{U_\infty}\right) = \frac{S'''_{eb}(x)}{4\pi} \ln (1 - M_\infty{}^2 - ku_B)$$

$$+ \frac{d}{dx}\left[\frac{S''_{eb}(x)}{4\pi} \ln \frac{S_{eb}(x)}{4\pi x(\ell - x)} + \frac{1}{4\pi} \int_0^\ell \frac{S''_{eb}(x) - S''_{eb}(\xi)}{|x - \xi|} d\xi\right] \qquad (6)$$

and for purely supersonic flows ($M_{cr,u} < M_\infty$)

$$\frac{d}{dx}\left(\frac{u_B}{U_\infty}\right) = \frac{S'''_{eb}(x)}{4\pi} \ln (M_\infty{}^2 - 1 + ku_B)$$

$$+ \frac{d}{dx}\left[\frac{S''_{eb}(x)}{4\pi} \ln \frac{S_{eb}(x)}{4\pi x^2} + \frac{1}{2\pi} \int_0^x \frac{S''_{eb}(x) - S''_{eb}(\xi)}{x - \xi} d\xi\right] \qquad (7)$$

where C in equation (5) is Euler's constant ≈0.577, k in equations (6) and (7) is equal to $M_\infty{}^2(\gamma + 1)/U_\infty$, $S_{eb}(x)$ represents the area distribution of the equivalent body, and primes indicate differentiation with respect to the appropriate variable. These differential equations have been programmed for computation in reference 6 where details regarding starting conditions, numerical techniques, accuracy, limitations, etc. are provided.

Wing and Body Geometry

The classes of wing-body configurations examined in reference 7 and in this study are composed of finite thickness wing and either circular or elliptic cross-sectional bodies in which the bodies are area-rule indented along the wing-body junction in such a manner that the total cross-sectional area distribution (body plus wing) remains identical to that of a smooth body having a certain specified profile. The general class of wings considered have symmetric planforms consisting of straight leading and trailing edges, swept at arbitrary angles $\beta_{\ell e}$ and β_{te} respectively, to the y axis. In reference 7, the planform shapes were restricted to wings having either straight or sweptforward trailing edges and zero taper ratio. This work extends that class to wings with sweptback trailing edges and taper ratio between zero and one, as shown in figures 2 and 3. The wing profiles are represented by expressions of the form

$$\frac{Z_w}{c_w} = \frac{\tau_w m^{(m/m-1)}}{2(m-1)} \left(\frac{\bar{x}}{c_w} - \left(\frac{\bar{x}}{c_w}\right)^m\right) \tag{8}$$

or

$$\frac{Z_w}{c_w} = \frac{\tau_w m^{(m/m-1)}}{2(m-1)} \left(1 - \frac{\bar{x}}{c_w} - \left(1 - \frac{\bar{x}}{c_w}\right)^m\right) \tag{9}$$

where c_w is the local chord, $\bar{x}$ the distance from the leading edge, m is a constant ≥ 2, and τ_w is the wing thickness-to-chord ratio. In addition, the wings are assumed to maintain a constant thickness-to-chord ratio across the span, with the consequence that the wing profiles at all spanwise locations are geometrically similar. Thus,

$$\frac{\tau_w}{2} = \frac{(Z_w(x,y))_{max}}{c_w(y)} = \frac{(Z_w(x,o))_{max}}{C_{R_t}} \tag{10}$$

where C_{R_t} is the wing root chord.

Two categories of body shape are considered. Figure 2(a) and (b), illustrates two members of the first category which have indented bodies that are circular in cross section, while figure 3(a) and (b) illustrates two members of the second category which have indented bodies that are elliptic in cross section and that maintain a constant ratio $\lambda(=a/b)$ of semimajor to semiminor axis along the entire body length. In reference 7, the profiles of the equivalent bodies of revolution of the wing-circular body combinations are described by the expressions

$$\frac{R_{eb}}{\ell} = \frac{\tau_{eb} n^{n/n(n-1)}}{2(n - 1)} \left[\frac{x}{\ell} - \left(\frac{x}{\ell}\right)^n\right] \tag{11}$$

where the exponent n is given in terms of the location of maximum radius by

$$\left(\frac{x}{\ell}\right)_{R_{max}} = \left(\frac{1}{n}\right)^{1/(n-1)} \tag{12}$$

or

$$\frac{R_{eb}}{\ell} = \frac{\tau_{eb} n^{n/(n-1)}}{2(n - 1)} \left[1 - \frac{x}{\ell} - \left(1 - \frac{x}{\ell}\right)^n\right] \tag{13}$$

where

$$\left(\frac{x}{\ell}\right)_{R_{max}} = 1 - \left(\frac{1}{n}\right)^{1/(n-1)} \tag{14}$$

while the equivalent bodies of the wing-elliptic body combinations are parabolic-arc bodies, i.e. equations (11) or (13) with $n = 2$. This work extends the class of equivalent body profiles for both the circular and elliptic body shapes to include arbitrarily specified functions subject to certain closure and continuity restrictions on the derivatives that are discussed in the appropriate section of the included user's manual.

Straight or Sweptforward Trailing Edge Planforms

Circular bodies.- For finite thickness wing-circular body combinations having wings with finite tip chords and trailing edges that are either straight or sweptforward, the complex potentials, $W_{2,\alpha}$, $W_{2,t}$, and $W_{2,B}$ can be readily determined from the work of reference 7.

$$W_{2,\alpha} = -iU_{\infty}\alpha \left\{ \left[\left(\sigma + \frac{R_b^{\,2}}{\sigma}\right)^2 - \left(s + \frac{R_b^{\,2}}{s}\right)^2 \right]^{1/2} - \sigma \right\} \tag{15}$$

$$\frac{W_{2,t}}{U_{\infty}} = \frac{1}{\pi} \int_{R_b}^{s} \frac{dZ_w(x,\xi)}{dx} \ln \left[\frac{(\sigma^2 - \xi^2)\left(\sigma^2 - \frac{R_b^{\,4}}{\xi^2}\right)}{\sigma^4} \right] d\xi$$

$$+ \frac{1}{2\pi} \left[S'_{eb}(x) + 4\, Z_w(x,R_b) \frac{dR_b}{dx} \right] \ln \sigma \tag{16}$$

$$\frac{W_{2,B}}{U_{\infty}} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma \tag{17}$$

where σ is the complex variable in the crossflow plane

$$\sigma = y + iz \tag{18}$$

R_b is the indented body radius, s - depending upon axial location - represents the local wing semispan of either the leading ($s = s_t$) or trailing ($s = s_t$) edge, and Z_w represents the wing ordinates. The velocity components associated with these potentials can be found by substituting those expressions into the general formulas:

$$u_{2,i} = \frac{\partial \phi_{2,i}}{\partial x} = \text{R.P.}\ \frac{\partial W_{2,i}}{\partial x} \tag{19}$$

$$(v - w)_{2,i} = (\phi_y - i\phi_z)_{2,i} = \frac{dW_{2,i}}{d\sigma} \tag{20}$$

where the subscript i depends upon the particular potential in question and R.P. signifies the real part of a complex quantity.

These operations have been carried out and the resulting expressions, which are quite lengthy, are given in reference 7. It should be noted that, in the evaluation of the velocity components associated with the thickness problem $(W_{2,t})$, different sets of expressions are necessary depending on whether the point of interest is (1) at a general location, (2) on the wing surface, or (3) at the wing-body junction. These distinctions are required in order to account properly for the Cauchy singularities which appear in several of the integrals associated with the thickness velocity components. No such distinctions are required for the lifting $(W_{2,\alpha})$ or equivalent thickness $(W_{2,B})$ problems.

Because of the symmetry of the class of wing-body combinations considered, nonlifting flows will produce no lateral forces or moments. The only force will be the longitudinal drag force which can be determined through the general formula,

$$D_{\alpha=o} = D_{eb} - \frac{\rho_\infty}{2}\left(\oint_{c_t} \phi_{2,t} \frac{\partial\phi_{2,t}}{\partial n}\, d\sigma_t - \oint_{c_B} \phi_{2,B} \frac{\partial\phi_{2,B}}{\partial n}\, d\sigma_B\right) \tag{21}$$

where D_{eb} represents the drag of the equivalent body while the other two terms involve the line integral along their respective contours (C_t is the contour defined by the cross section of the wing-body combination while C_B is the contour about the equivalent area circular cylinder) of the product of the appropriate velocity potential and the normal velocity associated with it. We note that the drag indicated by equation (21) refers to the inviscid drag of the configuration minus the base pressure drag. As pointed out in reference 8, there exist many shapes of aerodynamic interest for which the two integrals involved cancel. In particular, we note that if the equivalent body and the original body have the same shape and surface slope at the base, as is the case for configurations studied here, then since both integrals are carried out over the same contour along which $\phi_{2,t} = \phi_{2,B}$ and $\partial\phi_{2,t}/\partial n = \partial\phi_{2,B}/\partial n$, the integrals cancel and $D_{\alpha=o} = D_{eb}$. If we define a drag coefficient C_D based upon the maximum cross-sectional area of the equivalent body, S_m, then

$$C_{D_{\alpha=o}} = C_{D_{eb}} = \frac{D_{eb}}{\frac{\rho_\infty}{2} U_\infty^2 S_m} = \frac{1}{S_m} \int_o^{X_b} C_{p_{eb}} S'_{eb}(x)\, dx \tag{22}$$

where X_b is the axial location of the body base and $C_{p_{eb}}$ is the pressure coefficient on the surface of the nonlifting equivalent body and is equal to

$$C_{p_{eb}} = -2 \frac{u_B}{U_\infty} - \left(\frac{dR_{eb}(x)}{dx}\right)^2 \tag{23}$$

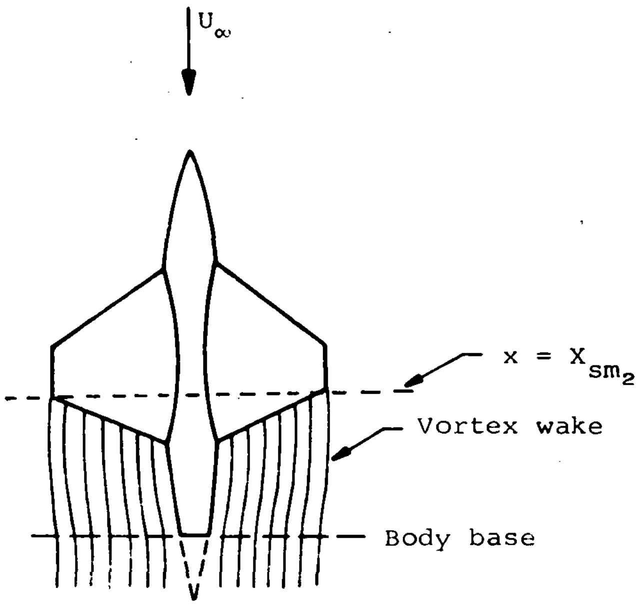
For the lifting situation, an exact analysis of the aerodynamic forces and moments, even within the framework of small disturbance theory, is quite formidable. The general formulas for determining the coefficients of lift, pitching moment, and drag are given by

$$C_L = \frac{L}{S_m q_\infty} = -\frac{2}{U_\infty} \oint_C \phi_{2,\alpha}\, d\sigma_C \tag{24}$$

$$C_m = \frac{M_y}{q_\infty \cdot S_m \cdot \ell} = \frac{-1}{q_\infty \cdot S_m \cdot \ell} \int_o^x \xi \frac{dL(\xi)}{d\xi}\, d\xi \tag{25}$$

$$C_{D_t} = \frac{D_t}{q_\infty S_m} = \frac{D_{eb}}{q_\infty S_m} - \frac{1}{S_m U_\infty^2}\left(\oint_C \phi_{2,\alpha} \frac{\partial \phi_{2,\alpha}}{\partial n}\, d\sigma_C + \oint_C \phi_{2,t} \frac{\partial \phi_{2,t}}{\partial n}\, d\sigma_t - \oint_B \phi_{2,B} \frac{\partial \phi_{2,B}}{\partial n}\, d\sigma_B\right) \tag{26}$$

where now the contour C, while still taken at the base of the body, must now account for the vortex wake which springs from the wing trailing edge and, as before, the drag given by equation (26) represents the inviscid drag minus the base pressure drag. Because the vortex lines near the body surface must follow the streamlines of the flow around the body, the vortex wake will not proceed parallel to the x axis, in general, as it does in many simpler cases considered in slender body theory; but will move away from or toward the body axis to follow the lateral expansion or contraction of the flow field near the body as shown below.



The resulting flow at the body base is influenced by the wake and, consequently, is no longer independent of the flow at cross sections preceding it. The solution of problems of this type is discussed briefly in reference 9. In general, they are quite difficult to solve and since they are by no means unique to transonic slender body flows, their exact solution is clearly beyond the scope of the present investigation. Because the analysis presented here, however, remains valid up to the axial location of the wing tip trailing edge $x = X_{sm2}$ (i.e. as long as the edge of the wing remains a leading edge) an estimate can be made of these quantities by making the assumption that beyond that point the vortex sheet remains parallel to the x axis and does not vary with x. With this premise in mind, we can proceed to evaluate equations (24), (25), and (26). Carrying through the indicated operations (see ref. 7 for details), we arrive at the result that the coefficients of lift, drag, and pitching moment are given by

$$C_L = \frac{2\pi\alpha}{S_m}\left(s_\ell^2 + \frac{R_b^4}{s_\ell^2} - R_{eb}^2\right)\Bigg|_{x = X_{sm_2}} \tag{27}$$

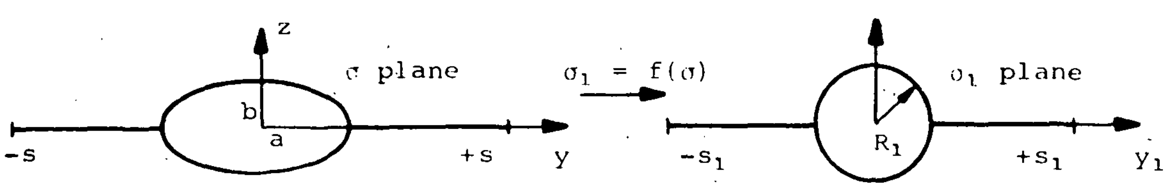
$$C_{D_t} = C_{D_{\alpha=o}} + \frac{\alpha}{2} C_L \tag{28}$$

$$C_m = \frac{2\pi\alpha}{S_m \cdot \ell}\left[- x\left(s_\ell^2 + \frac{R_b^4}{s_\ell^2} - R_{eb}^2\right)\Bigg|_{x = X_{sm_2}} + \int_o^{X_{r\ell e_1}} R_{eb}^2 \, d\xi\right.$$

$$\left. + \int_{X_{r\ell e_1}}^{X_{sm_2}} \left(s_\ell^2 + \frac{R_b^4}{s_\ell^2} - R_{eb}^2\right) d\xi\right] \tag{29}$$

where the drag coefficient at zero lift $C_{D_{\alpha=o}}$ is given by equation (22) and $X_{r\ell e_1}$ is the axial location of the point where the wing leading edge pierces the body surface.

Elliptic bodies.- The basic analysis of wing-body combinations composed of wings having finite tip chords with trailing edges that are either straight or sweptforward and bodies having indented elliptic cross sections such that the total cross-sectional area distribution equals the area of the original smooth body with elliptic cross section proceeds in a manner analogous to that used for the circular body shapes. Apparently, the most direct approach consists of reducing the elliptic cross section to a circular one by use of the appropriate Joukowski transformation and then applying the methods used for the circular shapes. The transformation required to take the ellipse into the circle shown below



is given by

$$\sigma_1 = \frac{\sigma + \sqrt{\sigma^2 - c^2}}{2} \tag{30}$$

where

$$c^2 = a^2 - b^2 \tag{31}$$

and

$$\sigma_1 = y_1 + iz_1 \tag{32}$$

This takes the ellipse into a circle of radius

$$R_1 = \frac{(a + b)}{2} \tag{33}$$

and the semispan s into the shortened semispan

$$s_1 = \frac{s + \sqrt{s^2 - c^2}}{2} \tag{34}$$

The potentials $W_{2,\alpha}$, $W_{2,t}$, and $W_{2,B}$ are then given by

$$\frac{W_{2,\alpha}}{U_\infty} = -i\alpha\left\{\left[\left(\sigma_1 + \frac{R_1{}^2}{\sigma_1}\right)^2 - \left(s_1 + \frac{R_1{}^2}{s_1}\right)^2\right]^{1/2} - \sigma\right\} \tag{35}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi}\int_{R_1}^{s_1} \frac{dZ_w\left(x,\xi_1 + \frac{c^2}{4\xi_1{}^2}\right)}{dx} \ln\left[\frac{(\sigma_1{}^2 - \xi_1{}^2)\left(\sigma_1{}^2 - \frac{R_1{}^4}{\xi_1{}^2}\right)}{\sigma_1{}^4}\right]\left(1 - \frac{c^2}{4\xi_1{}^2}\right) d\xi_1$$

$$+\left(\frac{S'_{eb}(x)}{2\pi} + 2\, Z_w(x,a)\,\frac{da}{dx}\right) \ln \sigma_1 \tag{36}$$

$$\frac{W_{2,B}}{U_\infty} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma \tag{37}$$

The velocity components associated with these potentials are found through the operations

$$u_{2,i} = \text{R.P.} \frac{\partial w_{2,i}}{dx} \tag{38}$$

$$(v - w)_{2,i} = \frac{dW_{2,i}}{d\sigma_1} \frac{d\sigma_1}{d\sigma} \tag{39}$$

These have been carried out and are given in reference 7 where it must be remembered in those formulas that depending upon axial location s_1 represents either the leading ($s_1 = s_{1_\ell}$) or trailing ($s_1 = s_{1_t}$) edge in the transformed plane. For the case of nonlifting flows about these classes of symmetric configurations, no lateral forces or moments exist so that the only force present is the longitudinal drag force. This can be determined through the use of equation (21) where now the contributions of the two line integrals do not cancel since the contour over which the product $\phi_{2,t}\ \partial\phi_{2,t}/\partial n$ is evaluated is the elliptic cross section at the base of the body whereas the contour for evaluating $\phi_{2,B}\ \partial\phi_{2,B}/\partial n$ is the circular cross section of the equivalent body. Carrying out the indicated operations, we find that the drag coefficient of this general class of nonlifting elliptic wing-body combinations is

$$C_{D_{\alpha=0}} = C_{D_{eb}} - \frac{1}{S_m}\left(\frac{S'_{eb}(x)}{2\pi}\right)^2 2\left[\frac{2}{\lambda}\ \ln\left(\frac{a(\lambda + 1)}{2\lambda}\right) K\left(\frac{\sqrt{\lambda^4 - 1}}{\lambda^2}\right) - \pi\ \ln R_{eb}\right] \tag{40}$$

where $C_{D_{eb}}$ is the drag coefficient of the nonlifting equivalent body and is given by equation (22), and $K(\xi)$ is the complete elliptic integral of the first kind.

For lifting flows at small angles of attack about these configurations, if we apply the same assumptions regarding the trailing vortex wake as were made for the circular body case, then the evaluation of equations (24), (25), and (26) provides the following results for the lift, pitching moment, and drag coefficients.

$$C_L = \frac{2\pi\alpha}{S_m}\left\{\left(\frac{s_\ell + \sqrt{s_\ell^2 - c^2}}{2}\right)^2 \left[1 + \frac{2c^2}{(s_\ell + \sqrt{s_\ell^2 - c^2})^2} + \left(\frac{a+b}{s_\ell + \sqrt{s_\ell^2 - c^2}}\right)^4\right] - R_{eb}^2\right\}\Bigg|_{x = X_{sm_2}}$$

(41)

$$C_m = \frac{2\pi\alpha}{S_m \cdot \ell}\Bigg(-x\left\{\left(\frac{s_\ell + \sqrt{s_\ell^2 - c^2}}{2}\right)^2 \left[1 + \frac{2c^2}{(s_\ell + \sqrt{s_\ell^2 - c^2})^2}\right.\right.$$

$$\left.\left. + \left(\frac{a+b}{s_\ell + \sqrt{s_\ell^2 - c^2}}\right)^4\right] - R_{eb}^2\right\}\Bigg|_{x = X_{sm_2}} + \lambda \int_0^{X_{r\ell e_1}} R_{eb}^2(x)\, dx$$

$$+ \int_{X_{r\ell e_1}}^{X_{sm_2}} \left\{\left(\frac{s_\ell + \sqrt{s_\ell^2 - c^2}}{2}\right)^2 \left[1 + \frac{2c^2}{(s_\ell + \sqrt{s_\ell^2 - c^2})^2} + \left(\frac{a+b}{s_\ell + \sqrt{s_\ell^2 - c^2}}\right)^4\right] - R_{eb}^2\right\} dx\Bigg)$$

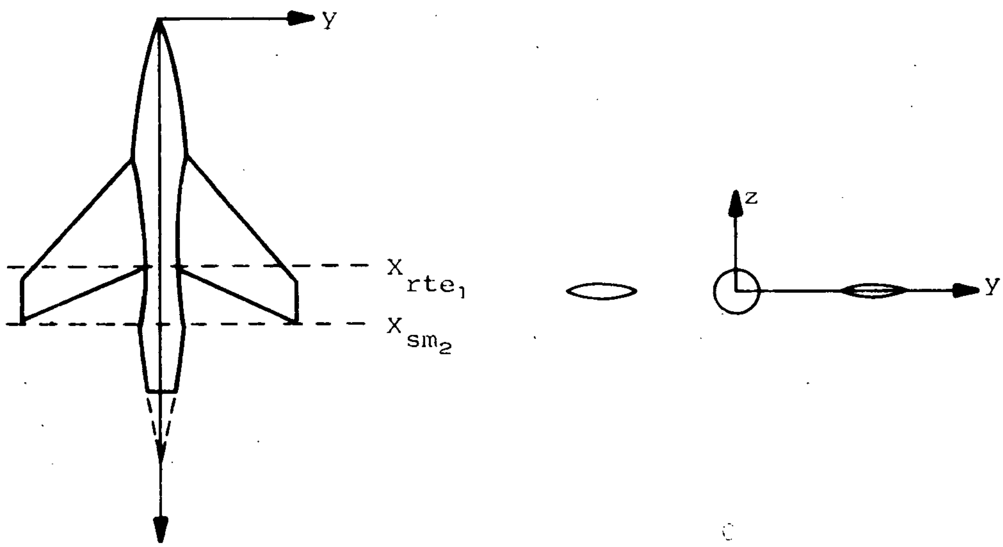
(42)

$$C_{D_t} = C_{D_{\alpha=o}} + \frac{\alpha}{2} C_L \qquad (43)$$

where $C_{D_{\alpha=o}}$ is the drag coefficient at zero lift and is given by equation (40).

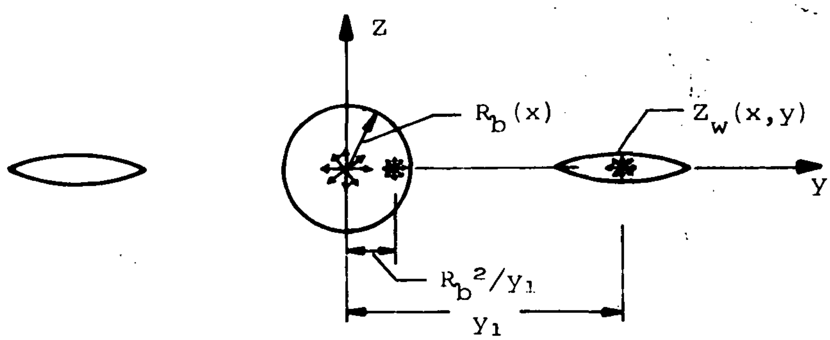
Sweptback Trailing Edge Planforms

Circular bodies.- For finite thickness wing-indented circular body combinations having sweptback trailing edges, as illustrated below,



new potential solutions are required to account for the gap between wing and body cross sections which appear in the crossflow plane between the axial locations $x = X_{rte_1}$ and $x = X_{sm_2}$. For lifting flows, the analysis is complicated even with the previously made assumption that the vortex sheet eminating from the trailing edge remains parallel to the x axis beyond the point $x = X_{sm_2}$. This is due to the presence of the vortex sheet in the gap between the wing and body from X_{rte_1} to X_{sm_2}. Consequently, the flow at any axial station beyond X_{rte_1} is influenced by the wake ahead of it and is no longer independent of the flow at preceding cross sections. Thus, the simplified analysis which was valid for lifting flows about wings with straight or sweptforward trailing edges - i.e. for cases where the edge of the wing always remained a leading edge - does not apply here. A new potential solution for $W_{2,\alpha}$ is required and this is beyond the work scope of the present investigation.

One of the primary goals of this study, however, is to determine the potential $W_{2,t}$ associated with the thickness problem for configurations of this type. For $x < X_{rte_1}$, equation (16) is valid. For $X_{rte_1} < x < X_{sm_2}$, a new potential must be developed to account for the gap between the wing and body. This is accomplished by using an extension of the method developed by Stocker in reference 10. That method is based upon the method of singularities and models the wing thickness by placing a continuous distribution of two-dimensional incompressible sources (or sinks) along the wing chordal plane together with their appropriate images within the body. The body itself is represented by a source (or sink) at the origin. Although originally developed for a wing attached to a body, this method can accommodate such a wing body as shown below



by distributing sources (or sinks) only along the wing and appropriately imaging them within the body. This method thus provides the following expression for the complex potential $W_{2,t}(x,y,z)$:

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} \ln \left[\frac{(\sigma^2 - \xi^2)(\sigma^2 - \frac{R_b{}^4}{\xi^2})}{\sigma^2}\right] d\xi + \frac{S_b'(x)}{2\pi} \ln \sigma$$

$$(X_{rte_1} < x < X_{sm_2}) \qquad (44)$$

Since for sweptback trailing edges with $X_{rte_1} < x < X_{sm_2}$, the equivalent body area distribution and actual body area distribution are related through the expression

$$S_{eb}(x) = S_b(x) + 4 \int_{s_t}^{s_\ell} Z_w(x,y)\, dy \tag{45}$$

we can write the alternate form

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} \ln \left[\frac{(\sigma^2 - \xi^2)(\sigma^2 - \frac{R_b{}^4}{\xi^2})}{\sigma^4}\right] d\xi + \frac{S'_{eb}(x)}{2\pi} \ln \sigma$$

$$(X_{rte_1} < x < X_{sm_2}) \tag{46}$$

The velocity components associated with this potential are determined through equations (19) and (20). As in the case of straight or sweptforward trailing edges, proper account must be taken of the familiar Cauchy singularity which appears in several of the integrals associated with these velocity components on the wing surface. If, because of the symmetry of these configurations, we restrict attention to the first quadrant of the crossflow plane ($y \geq 0$, $z \geq 0$), the following results are obtained. For points at any general location but not on the wing surface,

$$\frac{u_{2,t}}{U_\infty} = \frac{1}{2\pi}\left(\int_{s_t}^{s_\ell} \frac{d^2 Z_w(x,\xi)}{dx^2} \times\right.$$

$$\ell n\left\{\frac{\left[z^2 + (y-\xi)^2\right]\left[z^2 + (y+\xi)^2\right]\left[z^2 + (y-\frac{R_b{}^2}{\xi})^2\right]\left[z^2 + (y+\frac{R_b{}^2}{\xi})^2\right]}{\left(z^2 + y^2\right)^4}\right\} d\xi$$

$$+ 4R_b \frac{dR_b}{dx} \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} \frac{1}{\xi}\left[\frac{y + \frac{R_b{}^2}{\xi}}{z^2 + (y + \frac{R_b{}^2}{\xi})^2} - \frac{y - \frac{R_b{}^2}{\xi}}{z^2 + (y - \frac{R_b{}^2}{\xi})^2}\right] d\xi$$

$$+ \frac{dZ_w(x,s_\ell)}{dx} \frac{ds_\ell}{dx} \times$$

$$\ell n\left\{\frac{\left[z^2 + (y-s_\ell)^2\right]\left[z^2 + (y+s_\ell)^2\right]\left[z^2 + (y-\frac{R_b{}^2}{s_\ell})^2\right]\left[z^2 + (y+\frac{R_b{}^2}{s_\ell})^2\right]}{\left(z^2 + y^2\right)^4}\right\}$$

$$- \frac{dZ_w(x,s_t)}{dx} \frac{ds_t}{dx} \times$$

$$\ell n\left\{\frac{\left[z^2 + (y-s_t)^2\right]\left[z^2 + (y+s_t)^2\right]\left[z^2 + (y-\frac{R_b{}^2}{s_t})^2\right]\left[z^2 + (y+\frac{R_b{}^2}{s_t})^2\right]}{(z^2 + y^2)^4}\right\}$$

$$\left. + \frac{S''_{eb}(x)}{2} \ell n \left[z^2 + y^2\right]\right) \qquad (47)$$

$$\frac{v_{2,t}}{U_\infty} = \frac{1}{\pi}\left\{\int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx}\left[\frac{y-\xi}{z^2+(y-\xi)^2} + \frac{y+\xi}{z^2+(y+\xi)^2} + \frac{y-\frac{R_b^2}{\xi}}{z^2+\left(y-\frac{R_b^2}{\xi}\right)^2}\right.\right.$$

$$\left.\left. + \frac{y+\frac{R_b^2}{\xi}}{z^2+\left(y+\frac{R_b^2}{\xi}\right)^2}\right] d\xi + \frac{y}{z^2+y^2}\left(-4\int_{s_t}^{s_\ell}\frac{dZ_w(x,\xi)}{dx}\,d\xi + \frac{S'_{eb}(x)}{2}\right)\right\} \tag{48}$$

$$\frac{w_{2,t}}{U_\infty} = \frac{1}{\pi}\left\{z\int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx}\left[\frac{1}{z^2+(y-z)^2} + \frac{1}{z^2+(y+\xi)^2} + \frac{1}{z^2+\left(y-\frac{R_b^2}{\xi}\right)^2}\right.\right.$$

$$\left.\left. + \frac{1}{z^2+\left(y+\frac{R_b^2}{\xi}\right)^2}\right] d\xi + \frac{z}{z^2+y^2}\left(-4\int_{R_b}^{s}\frac{dZ_w(x,\xi)}{dx}\,d\xi + \frac{S'_{eb}(x)}{2}\right)\right\} \tag{49}$$

For points on the wing surface, i.e. $z = 0,\ s_t < y < s_\ell$:

$$\frac{u_{2,t}}{U_\infty} = \frac{1}{2\pi}\left\{\int_{s_t}^{s_\ell}\frac{d^2Z_w(x,\xi)}{dx^2}\,\ell n\left[\frac{(y+\xi)^2\left(y+\frac{R_b^2}{\xi}\right)^2\left(y-\frac{R_b^2}{\xi}\right)^2}{y^4}\right] d\xi\right.$$

$$+ \int_{s_t}^{s_\ell}\left(\frac{d^2Z_w(x,\xi)}{dx^2} - \frac{d^2Z_w(x,y)}{dx^2}\right)\ell n\left[\left(\frac{y-\xi}{y}\right)^2\right] d\xi$$

$$+ 4R_b\frac{dR_b}{dx}\frac{1}{y}\int_{s_t}^{s_\ell}\frac{dZ_w(x,\xi)}{dx}\left[\frac{1}{\xi+\frac{R_b^2}{y}} - \frac{1}{\xi-\frac{R_b^2}{y}}\right] d\xi$$

$$\left. + 2\frac{d^2Z_w(x,y)}{dx^2}\left\{(s_\ell - y)\left[\ell n\left(\frac{s_\ell - y}{y}\right) - 1\right] + (y - s_t)\left[\ell n\left(\frac{y - s_t}{y}\right) - 1\right]\right\}\right.$$

(Continued on next page)

$$+ \frac{dZ_w(x,s_\ell)}{dx} \frac{ds_\ell}{dx} \ell n \left[\frac{(s_\ell - y)^2 (s_\ell + y)^2 (y - \frac{R_b^2}{s_\ell})^2 (y + \frac{R_b^2}{s_\ell})^2}{y^8} \right]$$

$$- \frac{dZ_w(x,s_t)}{dx} \frac{ds_t}{dx} \ell n \left[\frac{(y - s_t)^2 (y + s_t)^2 (y - \frac{R_b^2}{s_t})^2 (y + \frac{R_b^2}{s_t})^2}{y^8} \right]$$

$$+ \left. \frac{S''_{eb}(x)}{2} \ell n(y^2) \right\} \tag{50}$$

$$\frac{v_{2,t}}{U_\infty} = \frac{1}{\pi} \left[\int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} \left(\frac{1}{y + \xi} + \frac{1}{y + \frac{R_b^2}{\xi}} + \frac{1}{y - \frac{R_b^2}{\xi}} \right) d\xi \right.$$

$$- \int_{s_t}^{s_\ell} \frac{\left(\frac{dZ_w(x,\xi)}{dx} - \frac{dZ_w(x,y)}{dx} \right)}{\xi - y} d\xi - \frac{dZ_w(x,y)}{dx} \ell n \left(\frac{s_\ell - y}{y - s_t} \right)$$

$$\left. + \frac{1}{y} \left(-4 \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} d\xi + \frac{S'_{eb}(x)}{2} \right) \right] \tag{51}$$

$$\frac{w_{2,t}}{U_\infty} = \frac{dZ_w(x,y)}{dx} \tag{52}$$

We note that in this case no singularities occur on the body surface or in the gap between the wing and body. At the leading and trailing edges of the wing, however, the characteristic logarithmic singularity associated with two-dimensional incompressible flow at a sharp edge appears.

As before, the thickness potential for the equivalent body cross section $W_{2,B}$ is given by equation (17). Because of the symmetry of these configurations, nonlifting flows will produce no lateral forces or moments but only a longitudinal drag force. Since, as is the usual case of realistic wing-body combinations, the configurations considered herein have the body base located aft of the trailing edge of the wing tip chord (i.e. $X_b > X_{sm_2}$), the drag coefficient at zero lift is given by

$$C_{D_{\alpha=o}} = C_{D_{eb}} \tag{53}$$

where $C_{D_{eb}}$ is the drag coefficient of the equivalent body of revolution and is given by equation (22).

Elliptic bodies.- For the wing-body combinations with indented elliptic cross section and sweptback trailing edges, equation (36) for $W_{2,t}$ applies for $x < X_{rte_1}$. For $X_{rte_1} < x < X_{sm_2}$, use of the Joukowski transformation, equation (30), provides the result that

$$\frac{W_{2,t}(\sigma_1)}{U_\infty} = \frac{1}{\pi}\int_{S_{1_t}}^{S_{1_\ell}} \frac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1^2})}{dx} \ln\left[\frac{(\sigma_1^2 - \xi_1^2)(\sigma_1^2 - \frac{R_1^4}{\xi_1^2})}{\sigma_1^4}\right]\left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ \frac{S'_{eb}(x)}{2\pi} \ln \sigma_1 \tag{54}$$

If we define the following quantities

$$q_1 + iq_2 = \frac{1}{\sigma_1^2 - \frac{c^2}{4}} \tag{55}$$

$$q_3 + iq_4 = \frac{\sigma_1}{\sigma_1^2 - \frac{c^2}{4}} \tag{56}$$

$$q_5 + iq_6 = \frac{\sigma_1^2}{\sigma_1^2 - \frac{c^2}{4}} \tag{57}$$

and again restrict attention to the first quadrant of the crossflow plane, equations (38) and (39) provide the following results for the velocity components. For a point at general location but not on the wing surface,

$$\frac{u_{2,t}}{U_\infty} = \frac{1}{2\pi}\left(\int_{s_{1_t}}^{s_{1_\ell}} \frac{d^2Z_w(x, \xi + \frac{c^2}{4\xi_1})}{dx^2} \times\right.$$

$$\ln\left\{\frac{\left[z_1{}^2 + (y_1 - z_1)^2\right]\left[z_1{}^2 + (y_1 + z_1)^2\right]\left[z_1{}^2 + (y_1 - \frac{R_1{}^2}{\xi_1})^2\right]\left[z_1{}^2 + (y_1 + \frac{R_1{}^2}{\xi_1})^2\right]}{(z_1{}^2 + y_1{}^2)^4}\right\} \times$$

$$(1 - \frac{c^2}{4\xi_1{}^2})\ d\xi_1$$

$$+ \left(\frac{\lambda^2 - 1}{\lambda^2}\right) a\ \frac{da}{dx}\left(-2\ q_1 \int_{s_t}^{s_\ell} \frac{dZ_w(x, \xi)}{dx}\ d\xi\right.$$

$$+ \left(\frac{\lambda + 1}{\lambda - 1}\right)\left[-q_5 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx}\ \frac{1}{\xi_1}\ \frac{(y_1 - \frac{R_1{}^2}{\xi_1})}{z_1{}^2 + (y_1{}^2 - \frac{R_1{}^2}{\xi_1})^2}\ d\xi_1\right.$$

$$- q_6\ z_1 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx}\ \frac{1}{\xi_1}\ \frac{d\xi_1}{z_1{}^2 + (y_1 - \frac{R_1{}^2}{\xi_1})^2}$$

$$+ q_5 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx}\ \frac{1}{\xi_1}\ \frac{y_1 + \frac{R_1{}^2}{\xi_1}}{z_1{}^2 + (y_1 + \frac{R_1{}^2}{\xi_1})^2}\ d\xi_1$$

$$\left. + q_6\ z_1 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx}\ \frac{1}{\xi_1}\ \frac{d\xi_1}{z_1{}^2 + (y_1 + \frac{R_1{}^2}{\xi_1})^2}\right]$$

$$+ \frac{c^2}{4}\left[q_3 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx}\ \frac{1}{\xi_1{}^2}\ \frac{(y_1 - \frac{R_1{}^2}{\xi_1})}{z_1{}^2 + (y_1 - \frac{R_1{}^2}{\xi_1})^2}\ d\xi_1\right.$$

(Continued on next page)

$$+ q_4 \, z_1 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx} \frac{1}{\xi_1^2} \frac{d\xi_1}{z_1^2 + (y_1 - \frac{R_1^2}{\xi_1})^2}$$

$$+ q_3 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx} \frac{1}{\xi_1^2} \frac{(y_1 + \frac{R_1^2}{\xi_1})}{z_1^2 + (y_1 + \frac{R_1^2}{\xi_1})^2} d\xi_1$$

$$\left. + q_4 \, z_1 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx} \frac{1}{\xi_1^2} \frac{d\xi_1}{z_1^2 + (y_1 + \frac{R_1^2}{\xi_1})^2} \right] \Bigg)$$

$$- \frac{\lambda^2 - 1}{\lambda^2} \, a \frac{da}{dx} q_1 \left(-4 \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} d\xi + \frac{S'_{eb}(x)}{2} \right) + \frac{dZ_w(x, s_\ell)}{dx} \frac{ds_\ell}{dx} \times$$

$$\ell n \left\{ \frac{\left[z_1^2 + (s_{1_\ell} - y_1)^2\right]\left[z_1^2 + (s_{1_\ell} + y_1)^2\right]\left[z_1^2 + (y_1 - \frac{R_1^2}{s_{1_\ell}})^2\right]\left[z_1^2 + (y_1 + \frac{R_1^2}{s_{1_\ell}})^2\right]}{(z_1^2 + y_1^2)^4} \right\}$$

$$- \frac{dZ_w(x, s_t)}{dx} \frac{ds_t}{dx} \times$$

$$\ell n \left\{ \frac{\left[z_1^2 + (y_1 - s_{1_t})^2\right]\left[z_1^2 + (y_1 + s_{1_t})^2\right]\left[z_1^2 + (y_1 - \frac{R_1^2}{s_{1_t}})^2\right]\left[z_1^2 + (y_1 + \frac{R_1^2}{s_{1_t}})^2\right]}{(z_1^2 + y_1^2)^4} \right\}$$

$$\left. + \frac{S''_{eb}(x)}{2} \, \ell n \, (z_1^2 + y_1^2) \right) \tag{58}$$

$$\frac{v_{2,t}}{U_\infty} = \frac{1}{\pi}\left\{ q_5 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx} \left[\frac{y_1 - \xi_1}{z_1^2 + (y_1 - \xi_1)^2} + \frac{y_1 + \xi_1}{z_1^2 + (y_1 + \xi_1)^2}\right.\right.$$

$$+ \frac{y_1 - \frac{R_1^2}{\xi_1}}{z_1^2 + (y_1 - \frac{R_1^2}{\xi_1})^2} + \left.\frac{y_1 + \frac{R_1^2}{\xi_1}}{z_1^2 + (y_1 + \frac{R_1^2}{\xi_1})^2}\right]\left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ q_6 z_1 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx} \left[\frac{1}{z_1^2 + (y_1 - \xi_1)^2} + \frac{1}{z_1^2 + (y_1 + \xi_1)^2}\right.$$

$$+ \frac{1}{z_1^2 + (y_1 - \frac{R_1^2}{\xi_1})^2} + \left.\frac{1}{z_1^2 + (y_1 + \frac{R_1^2}{\xi_1})^2}\right]\left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$\left.+ \frac{q_5 y_1 + q_6 z_1}{z_1^2 + y_1^2}\left(-4\int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx}\, d\xi + \frac{S'_{eb}(x)}{2}\right)\right\} \tag{59}$$

$$\frac{w_{2,t}}{U_\infty} = \frac{1}{\pi}\left\{ -q_6 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx} \left[\frac{y_1 - \xi_1}{z_1^2 + (y_1 - \xi_1)^2} + \frac{y_1 + \xi_1}{z_1^2 + (y_1 + \xi_1)^2}\right.\right.$$

$$+ \frac{y_1 - \frac{R_1^2}{\xi_1}}{z_1^2 + (y_1 - \frac{R_1^2}{\xi_1})^2} + \left.\frac{y_1 + \frac{R_1^2}{\xi_1}}{z_1^2 + (y_1 + \frac{R_1^2}{\xi_1})^2}\right]\left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ q_5 z_1 \int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx} \left[\frac{1}{z_1^2 + (y_1 - \xi_1)^2} + \frac{1}{z_1^2 + (y_1 + \xi_1)^2}\right.$$

(Continued on next page)

$$+ \frac{1}{z_1^2 + (y_1 - \frac{R_1^2}{\xi_1})^2} + \frac{1}{z_1^2 + (y_1 + \frac{R_1^2}{\xi_1})^2} \Bigg] \left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ \frac{q_5 z_1 - q_6 y_1}{z_1^2 + y_1^2} \left(-4 \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} d\xi + \frac{S'_{eb}(x)}{2} \right) \tag{60}$$

For points on the wing surface, i.e., $z = 0$, $s_t < y < s_\ell$ (or equivalently $z_1 = 0$, $s_{1_t} < y_1 < s_{1_\ell}$),

$$\frac{u_{2,t}}{U_\infty} =$$

$$\frac{1}{2\pi} \left\{ \int_{s_{1_t}}^{s_{1_\ell}} \frac{d^2Z_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx^2} \ell n \left[\frac{(y_1 + \xi_1)^2 (y_1 + \frac{R_1^2}{\xi_1})^2 (y_1 - \frac{R_1^2}{\xi_1})^2}{y_1^6} \right] \left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1 \right.$$

$$+ \int_{s_{1_t}}^{s_{1_\ell}} \left(\frac{d^2Z_w(x, \xi_1 + \frac{c^2}{4\xi_1})}{dx^2} - \frac{d^2Z_w(x, y_1 + \frac{c^2}{4y_1})}{dx^2} \right) \ell n \left[\left(\frac{y_1 - \xi_1}{y_1} \right)^2 \right] \left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ \frac{d^2Z_w(x, y_1 + \frac{c^2}{4y_1})}{dx^2} 2 \left((s_{1_\ell} - y_1) \left[\ell n \left(\frac{s_{1_\ell} - y_1}{y_1} \right) - 1 \right] + (y_1 - s_{1_t}) \left[\ell n \left(\frac{y_1 - s_{1_t}}{y_1} \right) - 1 \right] \right.$$

$$\left. - \frac{c^2}{4} \frac{1}{y_1} \left[\left(\frac{s_{1_\ell} - y_1}{s_{1_\ell}} \right) \ell n \left(\frac{s_{1_\ell} - y_1}{y_1} \right) + \left(\frac{y_1 - s_{1_t}}{s_{1_t}} \right) \ell n \left(\frac{y_1 - s_{1_t}}{y_1} \right) - \ell n \left(\frac{s_{1_\ell}}{s_{1_t}} \right) \right] \right)$$

$$+ \left(\frac{\lambda^2 - 1}{\lambda^2} \right) a \frac{da}{dx} \left[-2 q_1 \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} d\xi - q_1 \left(-4 \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} d\xi + \frac{S'_{eb}(x)}{2} \right) \right.$$

(Continued on next page)

$$+\left(\frac{\lambda + 1}{\lambda - 1}\right)\frac{q_5}{y_1}\left[-\int_{s_{1_t}}^{s_{1_\ell}} \frac{\dfrac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx}}{\xi_1 - \dfrac{R_1{}^2}{y_1}} d\xi_1 + \int_{s_{1_t}}^{s_{1_\ell}} \frac{\dfrac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx}}{\xi_1 + \dfrac{R_1{}^2}{y_1}} d\xi_1\right]$$

$$+\frac{c^2}{4}\frac{q_3}{y_1}\left[\left[\int_{s_{1_t}}^{s_{1_\ell}} \frac{\dfrac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx}}{\xi_1(\xi_1 - \dfrac{R_1{}^2}{y_1})} d\xi_1 + \int_{s_{1_t}}^{s_{1_\ell}} \frac{\dfrac{dZ_w(x,\xi_1 + \frac{c^2}{4\xi_1})}{dx}}{\xi_1(\xi_1 + \dfrac{R_1{}^2}{y_1})} d\xi_1\right]\right]$$

$$+\frac{dZ_w(x,s_\ell)}{dx}\frac{ds_\ell}{dx}\,\ell n\left[\frac{(s_{1_\ell} - y_1)^2(s_{1_\ell} + y_1)^2(y_1 - \frac{R_1{}^2}{s_{1_\ell}})^2(y_1 + \frac{R_1{}^2}{s_{1_\ell}})^2}{y_1{}^8}\right]$$

$$-\frac{dZ_w(x,s_t)}{dx}\frac{ds_t}{dx}\,\ell n\left[\frac{(y_1 - s_{1_t})^2(y_1 + s_{1_t})^2(y_1 - \frac{R_1{}^2}{s_{1_t}})^2(y + \frac{R_1{}^2}{s_{1_t}})^2}{y_1{}^8}\right]$$

$$+\frac{S''_{eb}(x)}{2}\,\ell n\,(y_1{}^2)\Bigg\} \qquad (61)$$

$$\frac{v_{2,t}}{U_\infty} = \frac{q_5}{\pi}\left[\int_{s_{1_t}}^{s_{1_\ell}} \frac{dZ_w\left(x, \xi_1 + \frac{c^2}{4\xi_1}\right)}{dx}\left(\frac{1}{y_1 + \xi_1} + \frac{1}{y_1 + \frac{R_1{}^2}{\xi_1}} + \frac{1}{y - \frac{R_1{}^2}{\xi_1}}\right)\left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1\right.$$

$$- \int_{s_{1_t}}^{s_{1_\ell}} \frac{\left(\frac{dZ_w\left(x,\xi_1 + \frac{c^2}{4\xi_1}\right)}{dx} - \frac{dZ_w\left(x,y_1 + \frac{c^2}{4y_1}\right)}{dx}\right)}{\xi_1 - y_1}\left(1 - \frac{c^2}{4\xi_1{}^2}\right) d\xi_1$$

$$- \frac{dZ_w\left(x,y_1 + \frac{c^2}{4y_1}\right)}{dx}\left\{\ell n\left(\frac{s_{1_\ell} - y_1}{y_1 - s_{1_t}}\right) + \frac{c^2}{4}\left[\frac{1}{y_1{}^2}\left[\ell n\left(\frac{s_{1_\ell}}{s_{1_t}}\right)\right.\right.\right.$$

$$\left.\left.\left.- \ell n\left(\frac{s_{1_\ell} - y_1}{y_1 - s_{1_t}}\right)\right] + \frac{1}{y_1}\left(\frac{s_{1_\ell} - s_{1_t}}{s_{1_\ell} \cdot s_{1_t}}\right)\right]\right\}$$

$$\left. + \left(-4 \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} d\xi + \frac{S'_{eb}(x)}{2}\right)\frac{1}{y_1}\right] \qquad (62)$$

$$\frac{w_{2,t}}{U_\infty} = \frac{dZ_w(x,y)}{dx} \qquad (63)$$

The drag coefficient at zero lift $C_{D_{\alpha=o}}$ of this class of configurations is given by equation (40).

RESULTS AND DISCUSSION

While experimental verification of the theory is considered essential, particularly at this stage of development, there is not available at the present time experimental data for transonic flows about wing-indented body combinations suitable for comparison with theory developed here. Although a parallel experimental program was originally considered for a typical member of the class of configurations described herein, that program has, unfortunately, been delayed. Consequently, experimental verification of the theory, particularly for pressure distribution comparisons which are vital in assessing the validity of the assumptions of the theory within the various regions (body surface, wing surface, wing-body junction, wing leading and trailing edges, etc.) of the near flow field of these configurations will have to be deferred.

In order to illustrate the general behavior of the theoretical results for transonic flows about the slender wing-body combinations considered here, the surface and flow field pressure distributions for several typical members of the classes of configurations described previously are given in figures 4 through 7. For example, in figure 4 pressure distributions are presented for a finite thickness wing-indented circular body combination with a nonzero taper ratio and straight trailing edge in which the equivalent body is a parabolic-arc of thickness ratio $D/\ell = 0.1$, the wing is a truncated delta wing with aspect ratio $AR = 1.7$, and taper ratio $TR = 0.2$ (so that $\beta_{\ell e} = 58^{\circ}$, $\beta_{te} = 0$), and has parabolic-arc profiles of thickness/chord ratio $t/c_w = 0.04$. The wing root chord is half the body length, i.e., $c_{R_T}/\ell = 0.5$, the root chord leading edge is located at $X_{r\ell e}/\ell = 0.25$, and the body base is at $X_{b/\ell} = 0.86$. The longitudinal pressure distributions given in figure 4 are for the free-stream conditions $M_\infty = 1$, $\alpha = 0^{\circ}$ and are presented at the two angular positions $\theta = 0^{\circ}, 90^{\circ}$ in the crossflow plane and at locations on the body surface and also along lines parallel to the body axis but removed laterally from it by distances of 1, 2, and 4 times the maximum equivalent body diameter D. Thus, the pressure distributions given for $\theta = 0^{\circ}$ and $r/D = 1, 2$ cut across the wing surface, intersecting the leading edge at the axial positions $x/\ell = 0.410$ and 0.570, respectively.

The wing-body surface pressure distributions shown in the first plot of figure 4, when compared to the pressure distribution on the equivalent body alone, demonstrate the large effect that the wing has upon the body pressure distribution. Moreover, it clearly shows the rapid variation of the pressure distributions caused by the singularities at the points (x/ℓ = 0.320, 0.75) where the wing leading and trailing edges pierce the body surface and also at x/ℓ = 0.65 where the leading edge of the wing tip chord is. The discontinuities at the points where the leading and trailing edges intersect the body surface are related to the characteristic logarithmic singularity associated with the two-dimensional thickness problem (i.e., $\phi_{2,t}$) of flow at a sharp edge. The discontinuity at the leading edge of the wing tip chord is due to the discontinuity in slope of the indented body at that point. This discontinuity occurs since in order that the equivalent body area distribution and its derivatives remain smooth, it is necessary for the indented body to have a slope discontinuity at x/ℓ = 0.65 to compensate for the one due to the wing. This discontinuity also occurs at the trailing edge of the wing tip chord and would be evident if the trailing edge were swept forward, so that the point ($x = X_{sm_1}$) would be separate and distinct from the axial location where the trailing edge pierces the body surface ($x = X_{rte_1}$). For the case of a straight trailing edge as in figure 4, those points, of course, coincide so that only one singularity is evident. The flow field distributions shown at r/D = 1, 2, and 4 illustrate several interesting features. The most prominent is the propagation into the flow field of the singularities which occur in the surface pressure distribution at the three points discussed above. This is a direct consequence of using the transonic equivalence rule to provide flow-field information based upon knowledge of flow properties at the body surface. Also evident in the distributions along the lines r/D = 1, 2, $\theta = 0^{\circ}$ are the logarithmic singularities at x/ℓ = 0.410 and 0.570, respectively, as those lines cross the wing leading edge. The longitudinal flow field pressure distributions provide insight into the rapidity with which the flow field becomes axisymmetric and equal to that about the equivalent body. At the lateral distance r/D = 4, the pressure distributions at $\theta = 0^{\circ}, 90^{\circ}$ at points ahead of the leading edge of the wing tip chord (X_{sm_2}/ℓ = 0.65) are virtually indistinguishable from that about about the equivalent body - except for the exponentially small region of influence of the logarithmic singularity propagating out from the body surface at the point where the leading edge pierces the body surface. However, within the axial region corresponding to the wing tip chord, $0.65 < x/\ell < 0.75$, the pressure distribution still shows some effect of the wing, although it is clearly diminishing. This is not surprising and could have been anticipated from the results from the delta wing with zero taper ratio given in figure 4 of reference 7 which also indicate that the effect of the wing on the flow field at lateral distances of several maximum body radii is negligible at all axial locations except those in the near vicinity of the wing maximum span. Knowledge of the region in which the flow about geometrically complex configurations of this type can be considered axisymmetric and equal to that about the equivalent body is quite significant and can provide, for example, useful information for a completely numerical finite difference solution in applying the far-field boundary condition. The drag coefficient for this configuration, which is provided by evaluating numerically the integral in equation (22), is found to be $C_{D_t} = 0.1044$.

Analogous results are given in figure 5 for a lifting flow about this same configuration for the free-stream conditions $M_\infty = 1$ and $\alpha = 2^\circ$. We note again that the singularities discussed with regard to the nonlifting case also appear here. Moreover, due to the nature of lifting flows near a sharp edge, the logarithmic singularities associated with the thickness problem are further reinforced by the inverse square root behavior associated with the two-dimensional lifting problem (i.e., $\phi_{2,\alpha}$) of flow around a sharp edge. The net result is the more rapid variation of pressure evident in those regions. Nevertheless, the flow field distributions again display the strong tendency to return to those generated by the equivalent body alone, as is most apparent in the flow field distributions at $r/D = 4$. At this angle of attack, equations (27), (28), and (29) provide the following results for the aerodynamic coefficients:

$$C_L = 1.7070, \quad C_{D_t} = 0.1342, \quad C_m = -0.8906$$

In order to demonstrate the pressure distribution behavior typical of the wing-body combinations considered here having swept-back trailing edges and non-zero taper ratios, results are given in figure 6 for a finite thickness wing-indented circular body combination in which the equivalent body is a parabolic-arc of thickness ratio $D/\ell = 0.10$, the wings have parabolic-arc profiles of thickness ratio $t/c_w = 0.04$, planform aspect ratio $AR = 2.8$, taper ratio $TR = 0.4$, root chord $C_{R_T}/\ell = 0.3$, with the root chord leading edge at $X_{r\ell e/\ell} = 0.25$ (so that $\beta_{\ell e} = 45^{\circ}$, $\beta_{te} = 23.75^{\circ}$). Analogous results are presented in figure 7 for a finite thickness wing-indented elliptic body combination composed of a parabolic-arc body of semimajor to semiminor axes $\lambda = 3$ and a wing essentially identical to the one described above for the circular body except that the trailing edge is swept at the angle $\beta_{te} = 25.05^{\circ}$. The trailing edge sweep angles of these configurations are such that the axial locations of leading edge of the wing tip chord and the point where the trailing edge pierces the body surface coincide, i.e., $X_{sm_1} = X_{rte_1}$.

In figure 6, we note that the general variation of both surface and flow field pressure distributions are essentially the same as those of the straight trailing edge configuration shown in figure 4 for points ahead of the leading edge of the wing tip chord ($X_{sm_1}/\ell = 0.571$). However, within the axial region containing the wing tip chord ($X_{sm_1} < x < X_{sm_2}$) and coincidentally, the wing trailing edge, the pressure distributions now indicate a much more rapid variation, while still exhibiting the same trend as that shown in figure 4. Apparently the gap between wing and body in the crossflow plane within this region influences the behavior of the surface pressures to a greater degree than in the case when there is no gap, and consequently an unbroken lateral distribution of sources along the wing from body to wing tip. The flow field distributions within the near flow field of this region maintain this rapid variation, with the distributions of C_p at $\theta = 0^{\circ}$, $r/D = 1, 2$ also exhibiting the characteristic influence of the logarithmic singularities at the points where these lines cross the trailing edge. Nevertheless, beyond the wing tip the flow still displays the characteristic tendency to return to that of the axisymmetric flow about the equivalent body, as is evident in the distribution at $r/D = 4$, a distance which is only slightly beyond the point of maximum span, $r/D = 3.2$.

The surface and flow field pressure distributions shown in figure 7 for the wing-indented elliptic body combination described above are essentially similar in behavior to those in figure 6 for the corresponding circular body and do not exhibit any new characteristic features. We note that the asymmetry introduced by ellipticity of the cross section alone (excluding the influence of the wing), while being evident at the body surface, rapidly dies out. In reference 6, it was shown that for a smooth elliptic body alone having a semimajor to semiminor axis ratio $\lambda = 3$ the flow field becomes essentially axisymmetric at $r/D = 1$.

Perhaps the most notable feature of the theoretical results presented here (and in ref. 7) for transonic flows about the classes of wing-indented body combinations being considered is the behavior of the pressure distributions caused by the singularities which occur at the following axial locations:

- $X_{r\ell e_1}$ -- leading edge pierces body surface
- X_{rte_1} -- trailing edge pierces body surface
- X_{sm_1} -- leading edge of wing tip chord
- X_{sm_2} -- trailing edge of wing tip chord

Although the singularities at general locations along the leading and trailing edges, for either nonlifting or lifting situations, could be included here, they are not, both since their character and origin are well known and also because they are local phenomena and, consequently, of restricted influence unlike the singularities delineated above.

It is important to realize that the basis of these singularities is essentially geometric in character, with the difficulty arising from either a discontinuity in first (at $x = X_{sm_1}, X_{sm_2}$) or second (at $x = X_{r\ell e_1}, X_{rte_1}$) derivative of the indented body area distribution, which causes, in turn, discontinuities in the surface velocity components. Then, because the transonic equivalence rule is used to provide flow field information based upon knowledge of flow properties on the body surface, these discontinuities are propagated laterally into the flow field making their presence even more evident. A direct method of alleviating this problem, while at the same time providing both a more general and realistic approximation would be to smooth these junction points with monotonically varying fillets. It appears, however, that a simple functional representation of fairing curves of this nature is not possible. Analytic (i.e., cubic), trigonometric, or exponential curves, while satisfying the end conditions of matching slope and ordinate at two points, fail to be continuously monotonic under boundary conditions typical of the configurations considered here. Thus, a wiggle would result in the faired curve and this is unacceptable. A means of eliminating this problem would be with piecewise continuous spline-fit functions. In any case, by whatever means the smoothing is accomplished, the result would be a more accurate representation of the actual solution in the vicinity of these points.

CONCLUDING REMARKS

Theoretical analysis and development of associated computer programs have been conducted in order to develop calculative techniques for predicting properties of transonic flows about certain classes of slender wing-body combinations. The theoretical analysis is based upon a combination of the transonic equivalence rule and uses either an arbitrarily specified solution or the local linearization method for determining the nonlifting transonic flow about the equivalent body.

Computational programs, which are documented in a general user's manual and included as part of this report, have been developed for finite thickness wing-body combinations in which the bodies are area-rule indented in such a manner that the resultant equivalent bodies remain smooth. The equivalent body profiles are either user-supplied subject to certain continuity and closure restrictions or program-supplied in which case the radius is of the general class $R \sim x/\ell - (x/\ell)^n$ or $1 - x/\ell - (1 - x/\ell)^n$. In addition, the body cross sectional shapes are either (1) circular or (2) elliptic and such that a constant ratio λ of semimajor to semiminor axes is maintained along the entire body length.

A general class of wings is considered which are symmetric in planform about the azimuthal body meridian (x-z plane) and consist of straight leading and trailing edges swept at arbitrary angles. The positions of the leading and trailing edges of the root chord are located at arbitrary locations on the body axis, and the profiles are described by $Z_w \sim \bar{x}/c_w - (\bar{x}/c_w)^m$ or $1 - \bar{x}/c_w - (1 - \bar{x}/c_w)^m$ where $\bar{x}$ is the axial distance from the leading edge and c_w is the local chord.

These programs provide longitudinal pressure distributions for both nonlifting and lifting situations, at arbitrary angular positions in the crossflow plane at points along the body and wing surface and also along lines parallel to the body axis but removed at arbitrarily specified lateral distances from it. In addition to the pressure distributions, the aerodynamic characteristics of lift, drag, and pitching moment are also provided.

The theoretical pressure distributions predicted by these programs for certain members of the class of configurations described above indicate quantitatively the relatively large effects of wing thickness and lift on both the body and flow field pressures, and also serve to point out the singularities inherent to the theory as it is presently constituted. In addition, they demonstrate the large influence that sweeping the trailing edge and introducing a finite tip chord has upon the pressure distributions.

In conclusion, we emphasize that the techniques employed here are quite fundamental and possess great generality so as to allow extension to even more complex configurations. Moreover, since the solutions to the various two-dimensional crossflow problems are independent of Mach number, they can be calculated once and for all once the geometry of the configuration is fixed and then combined with any one of a possible variety of solutions (experimental, numerical, etc.) for the transonic flow about the nonlifting equivalent body. We suggest, furthermore, that experimental work be conducted to determine surface and flow-field pressure distributions on selected wing-body combinations in order to define more clearly the extent to which the theory applies to configurations of this nature and, also, that consideration be given to developing methods to smooth the solutions in the vicinity of the various discontinuities in the area derivatives of the indented body shapes.

Nielsen Engineering & Research, Inc.
Mountain View, California
July 13, 1972

APPENDIX A

COMPUTER PROGRAM USER'S MANUAL

SUMMARY

An operating manual is given for the computer program developed in conjunction with the theoretical work presented in this report. The program computes the transonic surface and flow field pressure distributions and aerodynamic characteristics for various classes of wing-body combinations considered herein. Use is made of the transonic equivalence rule and either the local linearization method or a user-supplied solution for flow about the nonlifting equivalent body.

A description of the general operating procedure of the program is given, together with instructions for the preparation of input data, sample output of test cases, and a listing. The program is written in FORTRAM IV programming language and prepared specifically for use on an IBM 360/67 series computer. Typical running times are approximately 30 to 45 seconds for the equivalent body calculations using the local linearization method and about 2 minutes for the crossflow solution calculations involving approximately 1000 points located typically along the wing, wing-body junction, and flow field.

DESCRIPTION OF PROGRAM

The computer program presented here is applicable to several classes of finite thickness wing-body combinations discussed in the preceding report in which the bodies are area-rule indented along the wing-body junction in such a manner that the total cross-sectional area distribution is identical to that of a smooth body having a specified profile. The programs compute the surface and flow field pressure distributions, for both nonlifting and lifting situations for straight or swept forward trailing edge planforms and for nonlifting situations for swept back trailing edge planforms, at arbitrarily specified angular positions in the crossflow plane, at points along the body and wing surface, and also along lines parallel to the body axis but removed from it at specified lateral distances. In addition, the aerodynamic characteristics of lift, drag, and pitching moment are computed.

Wing and Body Geometry

The wing and body geometries of the configurations programmed are shown schematically in figures 2 and 3. Figure 2 illustrates two members of the class of wing-body combinations which have indented bodies that are circular in cross section, while figure 3 shows the corresponding members of the class having indented bodies with elliptic cross section.

Program-supplied equivalent body profiles.- Unless the user specifies to the contrary, the class of equivalent bodies of revolution of both types of the above configurations consist of profiles described by the equations

$$\frac{R_{eb}}{\ell} = \frac{\tau_{eb} n^{n/n(n-1)}}{2(n - 1)} \left[\frac{x}{\ell} - \left(\frac{x}{\ell}\right)^n\right] \tag{64}$$

or

$$\frac{R_{eb}}{\ell} = \frac{\tau_{eb} n^{n/(n-1)}}{2(n - 1)} \left[1 - \frac{x}{\ell} - \left(1 - \frac{x}{\ell}\right)^n\right] \tag{65}$$

with n = constant ≥ 2. In reference 7, the profiles of the elliptic bodies were restricted to parabolic arcs, i.e. equations (64) or (65) with $n = 2$. Thus, this work extends the elliptic body category to include the entire class of equivalent body profiles used for the case of the circular bodies.

User-supplied equivalent body profiles.- At the user's option, an arbitrarily specified equivalent body profile may be substituted in lieu of the above class of profiles. The modifications necessary to the program are detailed in the PROGRAM INPUT section. The restrictions on these profiles depend, in part, on the method used to calculate the solution for the nonlifting flow about the equivalent body, i.e. u_B. If the local linearization method, as presently constituted, is used to calculate u_B (see eqs. (5), (6), and (7)), then it is necessary that the profiles be closed, have sharp tips, and have continuous derivatives through the fourth. On the other hand, if the solution for u_B is user-supplied, then the requirement from the other portions of the solution, i.e. $\phi_{2,\alpha}$, $\phi_{2,t}$, and $\phi_{2,B}$, is that the equivalent body profiles have continuous derivatives through the second.

Indented-body profiles.- The ordinates of the indented body profiles are fixed once the equivalent body profile and wing profile are specified. For circular bodies with straight/sweptforward trailing edge planforms, the indented body radius R_b is found through a Newton-Raphson iteration procedure on the expression

$$\pi R_{eb}^2 = \pi R_b^2 + 4 \int_{R_b}^{s} Z_w(x,\xi)\, d\xi \tag{66}$$

while the derivatives dR_b/dx and d^2R_b/dx^2 are calculated by using an appropriate five-point difference formula. For sweptback trailing edge planforms, the above method applies up to $x = X_{rte_1}$. For $X_{rte_1} < x < X_{sm_2}$, R_b is found without iteration from the expression

$$\pi R_{eb}^2 = \pi R_b^2 + 4 \int_{s_t}^{s_\ell} Z_w(x,\xi)\, d\xi \tag{67}$$

Analogously, for elliptic bodies with straight/sweptforward trailing edges the semimajor axis a of the indented elliptic cross section is found by iteration on

$$\pi R_{eb}^2 = \frac{\pi a^2}{\lambda} + 4 \int_{a}^{s} Z_w(x,\xi)\, d\xi \tag{68}$$

with the derivatives da/dx and d^2a/dx being evaluated numerically by using the appropriate five-point difference formula. Fow sweptback trailing edge planforms with $X_{rte_1} < x < X_{sm_2}$, a is found directly from the expression

$$\pi R_{eb}^2 = \frac{\pi a^2}{\lambda} + 4 \int_{s_t}^{s_\ell} Z_w(x,\xi)\, d\xi \tag{69}$$

The general class of wings considered for both types of body shapes described above have wing planforms that consist of symmetric straight leading and trailing edges, swept at arbitrary angles $\beta_{\ell e}$ and β_{te}, respectively, to the y axis. Both $\beta_{\ell e}$ and β_{te}, are measured positive clockwise; thus, for β_{te} less than, equal to, or greater than zero, the trailing edge is correspondingly sweptforward, straight, or sweptback. The position of the leading edge of the wing root chord $X_{r\ell e}$ and its length C_{R_t} are arbitrary. The wing profiles are represented by expressions of the form

$$\frac{Z_w}{c_w} = \frac{\tau_w m^{(m/m-1)}}{2(m-1)}\left(\frac{\bar{x}}{c_w} - \left(\frac{\bar{x}}{c_w}\right)^m\right) \tag{70}$$

or

$$\frac{Z_w}{c_w} = \frac{\tau_w m^{(m/m-1)}}{2(m-1)}\left(1 - \frac{\bar{x}}{c_w} - \left(1 - \frac{\bar{x}}{c_w}\right)^m\right) \tag{71}$$

where c_w is the local chord, $\bar{x}$ the distance from the leading edge, m is a constant $\geq$ 2, and τ_w is the wing thickness-to-chord ratio. The wings are assumed to maintain a constant thickness-to-chord ratio across the span, with the consequence that the wing profiles at all spanwise locations are geometrically similar so that

$$\frac{\tau_w}{2} = \frac{(Z_w(x,y))_{max}}{c_w(y)} = \frac{(Z_w(x,o))_{max}}{C_{R_t}} \tag{72}$$

General

The coordinate system used in the program is a body-fixed Cartesian system centered at the body nose with the x axis directed rearward and aligned with the longitudinal axis of the body, the y axis directed to the right facing forward, and the z axis directed vertically upward, as shown in figure 1. Because the transonic equivalence rule allows the perturbation potential ϕ to be expressed in the form

$$\phi = \phi_{2,\alpha} + \phi_{2,t} - \phi_{2,B} + \phi_B \tag{73}$$

where each of the components has the meaning indicated in figure 1, and since $\phi_{2,\alpha}$, $\phi_{2,t}$, and $\phi_{2,B}$ satisfy the two-dimensional Laplace equation

$$(\phi_{2,i})_{yy} + (\phi_{2,i})_{zz} = 0 \tag{74}$$

they are independent of Mach number. Consequently, once the geometry of the configuration is fixed they can be calculated once and for all, stored, and used, for example, in a comparative study of a certain wing-body combination as the Mach number is varied systematically throughout the transonic range. An option for doing this is available and is discussed in the PROGRAM INPUT section. The only portion of the solution dependent upon M_∞ is ϕ_B and this term represents the solution to the full transonic equation (1) for flow about the nonlifting equivalent body.

Local linearization solution of u_B.- If the local linearization method is used to determine the solution for ϕ_B, or more conveniently, $u_B = (\phi_B)_x$, then according to whether M_∞ is near one, below the lower critical, or above the upper critical, equation (5), (6), or (7) must be integrated. Since these are all first order ordinary nonlinear differential equations, appropriate initial conditions are required. These are given at the point x_s, which is the positive root of the equation

$$S''_{eb}(x) = 0 \tag{75}$$

that is closest to the origin. The values of u_B/U_∞ at this point are, for accelerating transonic flows with $M_\infty \approx 1$ (eq. (5))

$$\frac{u_B}{U_\infty} = \frac{1 - M_\infty^2}{M_\infty^2(\gamma + 1)} + \frac{1}{4\pi}\int_0^x \frac{S''_{eb}(x) - S''_{eb}(\xi)}{x - \xi}\,d\xi \tag{76}$$

for purely subsonic flow (eq. (6))

$$\frac{u_B}{U_\infty} = \frac{1}{4\pi} \int_0^{\ell} \frac{S''_{eb}(x) - S''_{eb}(\xi)}{|x - \xi|} d\xi \tag{77}$$

and for purely subsonic flow (eq. (7))

$$\frac{u_B}{U_\infty} = \frac{1}{2\pi} \int_0^{x} \frac{S''_{eb}(x) - S''_{eb}(\xi)}{x - \xi} d\xi \tag{78}$$

The integrations start at x_s, proceed to a specified point near the nose, and upon reaching that point, return to x_s, restart the integration procedure, and then continue toward the tail. In each of these programs, the differential equations are integrated by using Hamming's modified predictor-corrector method described in the Scientific Subroutine Package (SSP) available from the IBM Corporation. The integrals involved in those differential equations are evaluated by using Simpson's rule.

User-supplied solution for u_B.- At the user's option, an arbitrarily-specified solution for u_B can be used in lieu of the local linearization solution. This solution can involve mixed transonic flows with imbedded shocks and can be determined in any of a variety of ways (numerical, experimental, etc.). Details regarding the manner of inputting this information to the program are discussed in the PROGRAM INPUT section.

Crossflow Potentials and Aerodynamic Characteristics

This section assembles for user convenience, the crossflow potentials and aerodynamic characteristics of all of the configurations considered in this report.

Straight/Sweptforward trailing-edge planforms.- For the classes of finite thickness wing-circular body combinations considered herein which have straight/sweptforward trailing edge planforms, (see fig. 2(a)), the following results are provided for $W_{2,\alpha}$, $W_{2,t}$, and $W_{2,B}$ at the indicated axial locations:

$$\frac{W_{2,\alpha}}{U_\infty} = \frac{i\alpha R_{eb}^2}{\sigma} \qquad (0 < x < X_{r\ell e_1}) \tag{79}$$

$$\frac{W_{2,\alpha}}{U_\infty} = -i\alpha\left\{\left[\left(\sigma + \frac{R_b{}^2}{\sigma}\right)^2 - \left(s_\ell + \frac{R_b{}^2}{s_\ell}\right)^2\right]^{1/2} - \sigma\right\}$$

$$(X_{r\ell e_1} < x < X_{sm_2}) \tag{80}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma \begin{pmatrix} 0 < x < X_{r\ell e_1} \\ X_{rte_1} < x < \ell \end{pmatrix} \tag{81}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{R_b}^{s} \frac{dZ_w(x,\xi)}{dx} \ln \left[\frac{(\sigma^2 - \xi^2)\left(\sigma^2 - \frac{R_b{}^4}{\xi^2}\right)}{\sigma^4}\right] d\xi$$

$$+ \frac{1}{2\pi}\left[S'_{eb}(x) + 4Z_w(x,R_b)\frac{dR_b}{dx}\right] \ln \sigma \qquad (X_{r\ell e_1} < x < X_{rte_1}) \tag{82}$$

$$\frac{W_{2,B}}{U_\infty} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma \qquad (0 < x < \ell) \tag{83}$$

where s in equation (82) denotes either the leading ($s = s_\ell$) or trailing ($s = s_t$) edge, depending upon the axial location. We note that for all of the configurations considered in this report, the solution for $W_{2,B}$ is given by equation (83).

The aerodynamic characteristics of this class of configurations are given as follows. The drag coefficient at zero lift is found through a numerical integration of the expression

$$C_{D_{\alpha=o}} = C_{D_{eb}} = \frac{1}{S_m} \int_o^{X_b} C_{p_{eb}} \frac{dS_{eb}(x)}{dx} dx \tag{84}$$

where $C_{D_{eb}}$ is the drag coefficient of the equivalent body alone, S_m is the maximum area of the equivalent body, and $C_{p_{eb}}$ is the pressure coefficient on the surface of the nonlifting equivalent body and is equal to

$$C_{p_{eb}} = - 2 \frac{u_B}{U_\infty} - \left(\frac{dR_{eb}(x)}{dx}\right)^2 \tag{85}$$

Because of the symmetry of these configurations no lateral forces or moments exist at $\alpha = 0$. For the lifting situation, the coefficients of lift, drag, and pitching moment are given by

$$C_L = \frac{2\pi\alpha}{S_m} \left(s_\ell{}^2 + \frac{R_b{}^4}{s_\ell{}^2} - R_{eb}^2\right)\Bigg|_{x = X_{sm_2}} \tag{86}$$

$$C_{D_t} = C_{D_{\alpha=o}} + \frac{\alpha}{2} C_L \tag{87}$$

$$C_m = \frac{2\pi\alpha}{S_m \cdot \ell} \left[-x \left(s_\ell{}^2 + \frac{R_b{}^4}{s_\ell{}^2} - R_{eb}^2\right)\Bigg|_{x = X_{sm_2}} + \int_o^{X_{r\ell e_1}} R_{eb}^2 \, d\xi \right.$$

$$\left. + \int_{X_{r\ell e_1}}^{X_{sm_2}} \left(s_\ell{}^2 + \frac{R_b{}^4}{s_\ell{}^2} - R_{eb}^2\right) d\xi \right] \tag{88}$$

where X_{sm_2} and $X_{r\ell e_1}$ are the axial locations, respectively, of the trailing edge of the wing tip-chord and the point where the wing leading edge pierces the body surface. The integrals involved in evaluating the pitching moment are calculated by using Simpson's rule.

The corresponding results for the wing-elliptic body combinations (see fig. 3(a)) are for $W_{2,\alpha}$, $W_{2,t}$:

$$\frac{W_{2,\alpha}}{U_\infty} = i\alpha \left\{\sigma_1 - \frac{R_1^{\,2}}{\sigma_1} - \sigma\right\} \quad (0 < x < X_{r\ell e_1}) \tag{89}$$

$$\frac{W_{2,\alpha}}{U_\infty} = -i\alpha\left\{\left[\left(\sigma_1 + \frac{R_1^{\,2}}{\sigma_1}\right)^2 - \left(s_1 + \frac{R_1^{\,2}}{s_1}\right)^2\right]^{1/2} - \sigma\right\}$$

$$(X_{r\ell e_1} < x < X_{sm_2}) \tag{90}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma_1 \quad \begin{pmatrix} 0 < x < X_{r\ell e_1} \\ X_{rte_1} < x < \ell \end{pmatrix} \tag{91}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{R_1}^{s_1} \frac{dZ_w\left(x, \xi_1 + \frac{c^2}{4\xi_1}\right)}{dx} \ln \left[\frac{(\sigma_1^{\,2} - \xi_1)\left(\sigma_1^{\,2} - \frac{R_1^{\,4}}{\xi_1^{\,2}}\right)}{\sigma_1^{\,4}}\right] \left(1 - \frac{c^2}{4\xi_1^{\,2}}\right) d\xi_1$$

$$+ \left(\frac{S'_{eb}(x)}{2\pi} + 2Z_w(x,a)\,\frac{da}{dx}\right) \ln \sigma_1 \qquad (X_{r\ell e_1} < x < X_{rte_1}) \tag{92}$$

where s_1 in equation (92) denotes either the leading ($s_1 = s_{1\ell}$) or trailing ($s_1 = s_{1t}$) edge in the transformed σ_1 plane.

The aerodynamic characteristics of these elliptic body configurations are given, for nonlifting flows, by

$$C_{D_{\alpha=o}} = C_{D_{eb}} - \frac{1}{S_m}\left(\frac{S'_{eb}(x)}{2\pi}\right)^2 2\left[\frac{2}{\lambda} \ln \left(\frac{a(\lambda + 1)}{2\lambda}\right) K\left(\frac{\sqrt{\lambda^4 - 1}}{\lambda^2}\right) - \pi \ln R_{eb}\right] \tag{93}$$

where $C_{D_{eb}}$ is given by equation (84) and $K(\xi)$ is the complete elliptic integral of the first kind; and for lifting flows by

$$C_L = \frac{2\pi\alpha}{S_m}\left\{\left(\frac{s_\ell + \sqrt{s_\ell^2 - c^2}}{2}\right)^2 \left[1 + \frac{2c^2}{(s_\ell + \sqrt{s_\ell^2 - c^2})^2} + \left(\frac{a + b}{(s_\ell + \sqrt{s_\ell^2 - c^2})}\right)^4\right] - R_{eb}^2\right\}\Bigg|_{x=X_{sm_2}} \tag{94}$$

$$C_{D_t} = C_{D_{\alpha=o}} + \frac{\alpha}{2} C_L \tag{95}$$

$$C_m = \frac{2\pi\alpha}{S_m \cdot \ell}\left(-x\left\{\left(\frac{s_\ell + \sqrt{s_\ell^2 - c^2}}{2}\right)^2 \left[1 + \frac{2c^2}{(s_\ell + \sqrt{s_\ell^2 - c^2})^2} + \left(\frac{a + b}{s_\ell + \sqrt{s_\ell^2 - c^2}}\right)^4\right] - R_{eb}^2\right\}\Bigg|_{x=X_{sm_2}} + \lambda \int_o^{X_{r\ell e_1}} R_{eb}^2(x)\, dx \right.$$
$$\left. + \int_{X_{r\ell e_1}}^{X_{sm_2}} \left\{\left(\frac{s_\ell + \sqrt{s_\ell^2 - c^2}}{2}\right)^2 \left[1 + \frac{2c^2}{(s_\ell + \sqrt{s_\ell^2 - c^2})^2} + \left(\frac{a + b}{s_\ell + \sqrt{s_\ell^2 - c^2}}\right)^4\right] - R_{eb}^2\right\} dx\right) \tag{96}$$

where the integrals involved in evaluating the pitching moment are calculated by using Simpson's rule.

Sweptback trailing edge planforms.- For the classes of finite thickness wing-circular body combinations having sweptback trailing edge planforms considered here (see fig. 2(b)), the following results for $W_{2,t}$ are provided:

$$\frac{W_{2,t}}{U_\infty} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma \qquad \begin{pmatrix} 0 < x < X_{r\ell e_1} \\ X_{sm_2} < x < \ell \end{pmatrix} \tag{97}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{R_b}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} \ln \left[\frac{(\sigma^2 - \xi^2)\left(\sigma^2 - \frac{R_b^4}{\xi^2}\right)}{\sigma^4}\right] d\xi$$

$$+ \frac{1}{2\pi}\left[S'_{eb}(x) + 4Z_w(x,R_b)\frac{dR_b}{dx}\right] \ln \sigma \qquad (X_{r\ell e_1} < x < X_{rte_1}) \tag{98}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{s_t}^{s_\ell} \frac{dZ_w(x,\xi)}{dx} \ln \left[\frac{(\sigma^2 - \xi^2)\left(\sigma^2 - \frac{R_b^4}{\xi^2}\right)}{\sigma^4}\right] d\xi + \frac{S'_{eb}(x)}{2\pi} \ln \sigma$$

$$(X_{rte_1} < x < X_{sm_2}) \tag{99}$$

and the drag coefficient at zero lift is given by

$$C_{D_{\alpha=0}} = C_{D_{eb}} \tag{100}$$

where $C_{D_{eb}}$ is given by equation (84).

The corresponding results for the wing-elliptic body combinations are

$$\frac{W_{2,t}}{U_\infty} = \frac{S'_{eb}(x)}{2\pi} \ln \sigma_1 \qquad \begin{pmatrix} 0 < x < X_{r\ell e_1} \\ X_{sm_2} < x < \ell \end{pmatrix} \tag{101}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{R_1}^{s_{1\ell}} \frac{dZ_w\left(x,\xi_1 + \frac{c^2}{4\xi_1}\right)}{dx} \ln \left[\frac{(\sigma_1^2 - \xi_1^2)\left(\sigma_1^2 - \frac{R_1^4}{\xi_1^2}\right)}{\sigma_1^4}\right] \left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ \left(\frac{S'_{eb}(x)}{2\pi} + 2Z_w(x,a)\frac{da}{dx}\right) \ln \sigma_1 \qquad (X_{r\ell e_1} < x < X_{rte_1}) \tag{102}$$

$$\frac{W_{2,t}}{U_\infty} = \frac{1}{\pi} \int_{R_1}^{s_{1\ell}} \frac{dZ_w\left(x, \xi_1 + \frac{c^2}{4\xi_1}\right)}{dx} \ln\left[\frac{(\sigma_1^2 - \xi_1^2)\left(\sigma_1^2 - \frac{R_1^4}{\xi_1^2}\right)}{\sigma_1^4}\right]\left(1 - \frac{c^2}{4\xi_1^2}\right) d\xi_1$$

$$+ \frac{S'_{eb}(x)}{2\pi} \ln \sigma_1 \qquad (X_{rte_1} < x < X_{sm_2}) \qquad (103)$$

with the drag coefficient at zero lift $C_{D_{\alpha=o}}$ given by equation (93).

Operating Procedure

The basic characteristics and general operating procedure of the computer program developed herein are straightforward and can be outlined as follows. After reading the input data (which is detailed in a subsequent section) and checking it for obvious errors, the program proceeds to calculate certain required geometrical and flow-field constants. Then, if the user selects the equivalent body profile to be of the class described by equations (64) or (65), the program proceeds to calculate the exponent n from information regarding the point of maximum thickness (see equations (12), (14)). The point x_s is next found by solving equation (75). If however, the equivalent body profile is user supplied, the calculation of n and x_s are omitted. Next, the axial locations $X_{r\ell e_1}$ and X_{rte_1}, which represent, respectively, the points where the wing leading and trailing edges pierce the body surface, are calculated. The exponent m describing the wing ordinates (see eqs. (70), (71)) is then calculated in a manner similar to that used to determine n. The calculation of n, x_s, $X_{r\ell e_1}$, X_{rte_1}, and m are all performed in an iterative fashion by using the standard Newton-Raphson iteration scheme.

With the calculation of the above parameters complete, the program prints a number of geometrical and flow-field characteristics for the case at hand. If the solution of u_B is user-supplied, the program begins at a point close to the nose $(x/\ell = 0.005)$ and proceeds toward the tail. If the local linearization method is used to determine u_B, then the appropriate initial value (eqs. (76), (77), or (78)) for the local linearization equation at hand (eqs. (5), (6), or (7)) is calculated at $x = x_s$ and the numerical integration begun. In the case of purely sub- or supersonic (eqs. (6), (7)) flow, it is convenient to redefine the dependent variables (see eqs. (80) through (88), ref. 6) and integrate a simplified differential equation. For the $M_\infty \approx 1$ case, it is more advantageous to integrate equation (5) as it stands. Because of the special character of that equation at $x = x_s$, it is necessary to use a Taylor series for u_B/U_∞ in the neighborhood of that point in order to avoid a singularity in the numerical integration. Consequently, for that case in addition to u_B/U_∞ several derivatives are also required and are calculated by the program. Details are given in reference 6. The numerical integrations then continue toward the nose and stop at a point ($x/\ell = 0.005$) close to it. The integrations are not carried directly to the nose because, although this is possible for the purely supersonic case, the local linearization method predicts a logarithmic singularity at $x = 0$ for a sharp-tipped body, much like that indicated by linearized theory. With the integration to the nose complete, the program returns to x_s, restarts the numerical integration, and continues toward the tail. As these calculations progress (using either a user-supplied or local linearization solution for u_B), the surface and flow-field pressure distributions are calculated from equations (2) and (3) and the output printed at specified axial locations. Until the point $x = X_{r\ell e_1}$ is reached, the appropriate crossflow solutions for determining these pressure distributions are those for the smooth body alone (see eqs. (79), (81), (83), (89), (91), and (101)). Beyond $X_{r\ell e_1}$, for the case of nonlifting flows, the crossflow solutions are calculated from equation (82), (83), or (92) and (83) for $X_{r\ell e_1} < x < X_{rte_1}$. Beyond X_{rte_1}, for planforms with straight/sweptforward trailing edges, the crossflow solutions revert to those for the smooth body alone; while for planforms with sweptback trailing edges, the appropriate crossflow solutions for $X_{rte_1} < x < X_{sm_2}$ are given by equations (99), (83), or (103), (83) and beyond X_{sm_2} the solutions revert to those for the smooth body alone. The calculations continue until the body base is reached, i.e. $x = X_b$; the calculation then returns to the mainline program, prints the value of the drag coefficient, and reads the input data for the next case. For lifting flows about planforms with straight/sweptforward trailing edges, the calculations proceed in a similar fashion to that of the nonlifting case (with the additional output of surface and flow-field pressure distributions at the angular locations $\pm\,\theta$ rather than just $+\,\theta$) until the axial location of the trailing edge of the wing tip chord is reached, i.e. $x = X_{sm_2}$. Beyond that point, for reasons given in reference 7, no further pressure distributions are given. However, the calculation of flow about the equivalent body, i.e. u_B, continues to the body base in order that the drag coefficient at zero lift $C_{D_{\alpha=o}}$ can be determined. When $x = X_b$, the calculation returns to the mainline program, determines the coefficients of lift, drag, and pitching moment from equation (86), (87), and (88), or (94), (95), and (96), prints these values, and then proceeds to read the input data for the next case.

PROGRAM INPUT

The variables that are input to the program are described in the following list:

Dictionary of Input Variables

AL — ratio of semimajor to semiminor axis (a/b) of elliptic cross section: program default value AL = 1

ALPHA — angle of attack, in degrees; program default value, ALPHA = 0

AMACH — free-stream Mach number

ANGLE — sweep angle, in degrees, of wing leading edge (measured positive clockwise from y axis and restricted to values 0 < ANGLE < 90, see figures 2, 3)

CRT — wing root chord normalized by complete body length, c_{R_t}/ℓ

ICOPY — integer index for program option for using previously-stored values of crossflow solutions; equal to 0 or 1; program default value, ICOPY = 0

MAREA — integer index for program option for using user-supplied or program-supplied subroutines for equivalent body area and derivatives, equal to 0 or 1; program default value, MAREA = 0

MOPT — integer index for program option for using user-supplied distribution for u_B or program-supplied local linearization solution; equal to 1, 2, 3, or 4.

NTHETA	integer indicating number of angular positions in crossflow plane at which output is desired; $1 \leq$ NTHETA ≤ 5
NXEB	integer indicating number of table entries of the user-supplied distribution of u_B; NXEB ≤ 201
RF(I)	six-dimensional vector representing values of r/D (the radial distance in the crossflow plane normalized by the maximum equivalent body diameter D) at which flow-field pressure distributions are to be calculated
SSMAX	maximum wing semispan normalized by complete body length, s_{max}/ℓ
TAUB	thickness ratio of equivalent body, D/ℓ
TAUW	thickness-to-chord ratio of wing (see eq. (10))
THETA(I)	NTHETA-dimensional vector representing values of the angle θ (in degrees) in the crossflow plane; $0 \leq$ THETA(I) ≤ 90
TR	wing planform taper ratio, c_{tip}/c_{R_t}; $0 \leq$ TR ≤ 1
XEB(I)	NXEB-dimensional vector representing values of the axial locations (normalized by the complete body length) where values for the user-supplied distribution u_B are given
XLBASE	axial location of body base normalized by the complete body length, x_b/ℓ
XLOUTP	interval size, as fraction of complete body length, between output stations for pressure distribution print-out
XRLE	axial location of leading edge of wing root chord normalized by complete body length, $x_{r\ell e}/\ell$
XMTB	axial location of position of maximum thickness of equivalent body of revolution normalized by complete body length (see eqs. (12), (14))
XMTW	location, as fraction of distance from wing leading edge to local chord length $(\overline{x}/c_w)$, of position of wing maximum thickness (see eqs. (8), (9))
XS2EB	user-supplied value of the axial location, normalized by the complete body length, where the user-supplied equivalent body profile satisfies $S''_{eb}(x) = 0$; only necessary as input when user supplies equivalent body profile and also uses local linearization method to calculate u_B

UEB(I) NXEB-dimensional vector representing values of u_B/u_∞ for the user-supplied distribution of u_B.

Input Format and Options

All of the input variables are entered into the program under a NAMELIST format (the one exception being the vectors UEB(I), XEB(I) which represent the ordinates and abscissas, respectively, of the user-supplied velocity distribution u_B and the format of these quantities is discussed below). The name of the NAMELIST data block is TRANIN.

Default values.- The following input variables have default values that are indicated below and unless the user wishes to change them, it is not necessary to enter them in the input data block

Variable Name	Default Value
AL	1.
ALPHA	0.
RF(I), I=1,2,3, 4,5,6	1.,2.,3.,4.,5.,6.,
NTHETA	2
THETA(I), I=1,2	0.,90.
MAREA	0
ICOPY	0
XLOUTP	.01

It is important to realize that the above variables assume their default values each time the program is run. If the user wishes any of the above variables to be different from its default value, this must be specified in the data statement for each run. All other input variables, once specified, remain unchanged by the program; thus, it is unnecessary to respecify them in subsequent runs if their values are to remain constant.

Local lineariztion option.- To use the local linearization method to determine u_B, it is necessary to specify in the input data the appropriate value for the integer index MOPT. Depending on the free stream Mach number, the proper value of MOPT to use the local linearization method is

Free Stream Mach No.	MOPT
$M_\infty \approx 1$ (near sonic flow)	1
$M_\infty \leq M_{cr,\ell}$ (below lower critical)	2
$M_{cr,u} \leq M_\infty$ (above upper critical)	3

User-supplied u_B option.- If the user wishes to supply the solution for u_B, then the program will bypass the local linearization calculations by specifying MOPT = 4. The solution for u_B is read into the program in the form of a tabular input of values of u_B/U_∞ - vs - x/ℓ immediately after the NAMELIST input block. Provision has been made for inputting ordinate and abscissa values up to a maximum number of 201 each (UEB(201), XEB(201)), i.e. values at each half percent of the body length if equally spaced. It is assumed that a sufficient number entries are made that linear interpolation in the table is appropriate.

User-supplied equivalent body profile.- If a class of equivalent body profiles not included in equation (64) or (65) is desired, then the user must set the integer index MAREA = 1, remove the following function subroutines from the program,

- FUNCTION SEBPI(DZ)
- FUNCTION S1EBPI(DZ)
- FUNCTION S2EBPI(DZ)
- FUNCTION S3EBPI(DZ)
- FUNCTION S4EBPI(DZ)

and replace them with his own. The above subroutines which are non-dimensionalized by normalizing them with respect to the body length, are defined in the following fashion.

$$\frac{S_{eb}(x/\ell)}{4\pi\ell^2} = \mathrm{SEBPI}(x/\ell)$$

$$\frac{S'_{eb}(x/\ell)}{4\pi\ell^2} = \frac{1}{4\pi\ell^2}\frac{dS_{eb}(x/\ell)}{d(x/\ell)} = \mathrm{S1EBPI}(x/\ell)$$

$$\frac{S''_{eb}(x/\ell)}{4\pi\ell^2} = \frac{d^2 S_{eb}(x/\ell)}{4\pi\ell^2 d(x/\ell)^2} = \text{S2EBPI}(x/\ell)$$

$$\frac{S'''_{eb}(x/\ell)}{4\pi\ell^2} = \frac{1}{4\pi\ell^2}\frac{d^3 S_{eb}(x/\ell)}{d(x/\ell)^3} = \text{S3EBPI}(x/\ell)$$

$$\frac{S^{IV}(x/\ell)}{4\pi\ell^2} = \frac{1}{4\pi\ell^2}\frac{d^4 S_{eb}(x/\ell)}{d(x/\ell)^4} = \text{S4EBPI}(x/\ell)$$

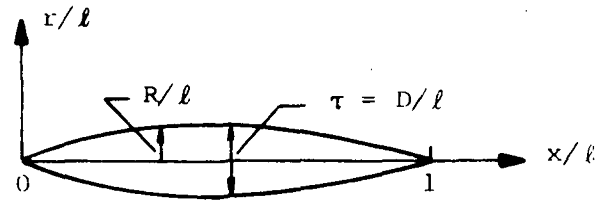
We note again the requirements that if the local linearization method is to be used with these subroutines, then the functions must be such that the complete profiles are closed, sharp-tipped, and have continuous area derivatives through the fourth. In addition, the user must supply the point x/ℓ = XS2EB, i.e. the point closest to the origin where

$$S''_{eb}(x/\ell) = 0$$

If, however, the user supplies <u>both</u> the equivalent body profiles <u>and</u> the distribution of u_B/U_∞ - vs - x/ℓ, then it is only necessary that the equivalent body area derivatives be continuous through the second. Also, for this case, it is unnecessary to specify XS2EB. These requirements are summarized below when user-specifying the equivalent body profile:

Value of MOPT	Requirements
MOPT = 1,2,3	(1) S_{eb}, S'_{eb}, S''_{eb}, S'''_{eb}, S^{IV}_{eb} continuous for $0 < x < \ell$
	(2) User must input value of x/ℓ = XS2EB where S''_{eb}(XS2EB) = 0
	(3) Set index MAREA = 1 in input
MOPT = 4	(1) S_{eb}, S'_{eb}, S''_{eb} continuous for $0 < x < \ell$
	(2) Set index MAREA = 1 in input

An illustrative example of a user-supplied equivalent body profile and the corresponding area and derivative subroutines to be used with, say, the local linearization method (MOPT = 1, 2, or 3) can be given as follows. Consider an equivalent body profile formed by the top half of the sinusoidal curve given by



$$\frac{R_{eb}(x/\ell)}{\ell} = \frac{\tau_{eb}}{2} \sin\left(\pi \left(\frac{x}{\ell}\right)\right) \qquad \text{with say} \quad \tau = .1$$

so that

$$\frac{S_{eb}(x/\ell)}{4\pi \ell^2} = \frac{\tau_{eb}^2}{16} \left(1 - \cos\left(2\pi \left(x/\ell\right)\right)\right)$$

$$\frac{S'_{eb}(x/\ell)}{4\pi \ell^2} = \frac{\pi \tau_{eb}^2}{8} \sin\left(2\pi (x/\ell)\right)$$

$$\frac{S''_{eb}(x/\ell)}{4\pi \ell^2} = \frac{\pi^2 \tau_{eb}^2}{4} \cos\left(2\pi (x/\ell)\right)$$

$$\frac{S'''_{eb}(x/\ell)}{4\pi \ell^2} = -\frac{\pi^3 \tau_{eb}^2}{2} \sin\left(2\pi (x/\ell)\right)$$

$$\frac{S^{IV}(x/\ell)}{4\pi \ell^2} = -\pi^4 \tau_{eb}^2 \cos\left(2\pi (x/\ell)\right)$$

The function subroutines for SEBPI(x/ℓ) and S1EBPI(x/ℓ) with $\tau = .1$ are given by

```
FUNCTION SEBPI(DZ)
TAU = .1
PI = 3.1415927
SEBPI = TAU*TAU*(1. - COS(2.*PI*DZ))/16.
```

```
RETURN
END
FUNCTION S1EBPI(DZ)
TAU = .1
PI = 3.1415927
S1EBPI = PI*TAU*TAU*SIN (2.*PI*DZ)/8.
RETURN
END
```

The FUNCTION subroutines S2EBPI, S3EBPI, and S4EBPI are given in analogous fashion. The point where

$$S''_{eb}(x/\ell) = 0$$

is given by

$$XS2EB = 0.25$$

and must be included in the NAMELIST input data block.

<u>Repetitive calculation storage option.</u>- If the user wishes to undertake a systematic study of a wing-body configuration of the classes considered herein in which the geometry of the configuration is frozen and the Mach number and/or angle of attack are varied, a provision is included in the program whereby the velocity components associated with the crossflow solution for $W_{2,t}$ - which is independent of M_∞ and α, and is by far the most time consuming crossflow potential to calculate - is stored at the user-specified output locations for later use. These velocity components are then provided for the remainder of the cases to be run rather than recalculated unnecessarily.

In order to activate the repetitive calculation option and make use of previously-stored results from a base run, it is necessary to set the integer index ICOPY = 1. The default value for ICOPY is ICOPY = 0 and this default value instructs the program to perform the crossflow calculations at the user-indicated axial locations and then automatically store these results in anticipation of use with the next case. If it is not desired to use those stored results for the next run, (i.e. ICOPY = 0 for the next case) the program simply replaces the previously-stored results with the ones being currently calculated.

Because for lifting situations, the crossflow calculations only proceed to $x/\ell = X_{sm_2}$ (as opposed to the $\alpha = 0$ case, which carries the calculation to $x/\ell = X_b$) unless all of the cases in the study are for $\alpha \neq 0$, the initial or base run which stores the crossflow results should be made for $\alpha = 0$.

Finally, when using the storage option, it is necessary, because of the different starting conditions involved, to use the same method for calculating u_B, i.e. (1) local linearization or (2) user-supplied distribution of u_B.

Data Format

The data format is most easily demonstrated by an example. Consider the case of a wing-elliptic body combination composed of a parabolic-arc equivalent body of thickness ratio 1/10, the ratio of semimajor to semiminor axis of the elliptic cross section is 3, the body base is at 85 percent of the complete body length, the wing profiles are parabolic arcs having a thickness/chord ratio of 0.04, the wing root chord is 40 percent of the complete body length with the leading edge of the wing root chord located at $x/\ell = 0.3$, the leading edge swept at 45 degrees, a taper ratio of 0.3, and a wing semispan being 28 percent of the complete body length (this implies the trailing edge is straight). The pressure distributions are required to be output at every 2 percent of the complete body length at angular locations $\theta = 0^\circ, 45^\circ, 90^\circ$, at radial distances $r/D = 1, 1.5, 2, 2.5, 3$, and 3.5 in the crossflow plane, at sonic free-stream conditions and 2 degrees angle of attack, by using the local linearization method to determine the flow about the equivalent body.

Thus, the input data cards would read (note that with a NAMELIST format, input variable sequencing is arbitrary):

CARD NO. 1

COLUMN NO.	2 9	80
	& TRANIN	AL=3.,AMACH=1.,MOPT=1,TAUB=.1,TAUW=.04,XMTB=.5,

CARD NO. 2

COLUMN NO.	2 80
	XMTW=.5,XRLE=.3,CRT=.4,ANGLE=45.,TR=.3,SSMAX=.28,NTHETA=3,

CARD NO. 3

COLUMN NO.	2 ... 80
	THETA(1)=0.,THETA(2)=40.,THETA(3)=90.,RF(1)=1.,RF(2)=1.5,

CARD NO. 4

COLUMN NO.	2 ... 80
	RF(3)=2.,RF(4)=2.5,RF(5)=3.,RF(6)=3.5,ALPHA=2.,XLOUTP=.02,

CARD NO. 5

COLUMN NO.	2 ... 80
	&END

If for this case the user wished to supply his own distribution of u_B/U_∞ - vs. - x/ℓ, this would have been done by specifying MOPT = 4 and also the integer NXEB representing the number (say, for example, 101) of values of u_B/U_∞ being entered (i.e. NXEB = 101) in the NAMELIST input data block. Next, all of the NXEB (101, in this example) values of u_B/U_∞ = UEB(I) would be read in under the card format 8F10.0, with each successive value of u_B/U_∞ occupying a space of 10 columns (including decimal point) with 8 values per card. Finally, the NXEB values (101, in this example) of the corresponding axial locations x/ℓ = XEB(I) of the above values of u_B/U_∞ would be read in under the same format. Thus,

Card format for UEB(I): Format (8F10.0), decimal point required

COLUMN NO.	10	20	70	80
Data for	UEB(1)	UEB(2)		UEB(8)

⋮ etc.

COLUMN NO.	10	20	30	40	50
Data for	UEB(97)	UEB(98)	UEB(99)	UEB(100)	UEB(101)

Card format for XEB(I): Format (8F10.0), decimal point required

COLUMN NO.	10	20	70	80
Data for	XEB(1)	XEB(2)		XEB(8)

⋮ etc.

COLUMN NO.	10	20	30	40	50
Data for	XEB(97)	XEB(98)	XEB(99)	XEB(100)	XEB(101)

MESSAGES PRINTED BY THE PROGRAMS

This section lists the messages printed by the programs and indicates what to do when they are encountered. The first group of messages (1 to 15) are concerned with errors in input quantities and are self-explanatory.

(1) INTERVAL SIZE FOR PRESSURE DISTRIBUTION PRINT-OUT MUST BE GREATER THAN 0 AND LESS THAN 1

This message indicates that the condition $0 < XLOUTP/\ell < 1$ has been violated.

(2) XMTB MUST BE GREATER THAN 0 AND LESS THAN 1

This message indicates that the condition $0 < (x/\ell)_{R_{max}} < 1$ has been violated.

(3) EQUIVALENT BODY THICKNESS RATIO MUST BE GREATER THAN ZERO.

This message indicates that the condition $D/\ell > 0$ has been violated.

(4) XMTW MUST BE GREATER THAN 0 AND LESS THAN 1

This message indicates that the condition $0 < \left(\frac{\bar{x}}{c_w}\right)_{Z_{max}} < 1$ has been violated.

(5) WING THICKNESS/CHORD RATIO MUST BE GREATER THAN ZERO

This message indicates that the condition $Z_{max}/c_w > 0$ has been violated.

(6) XRLE MUST BE GREATER THAN 0 AND LESS THAN 1

This message indicates that the condition $0 < X_{r\ell e}/\ell < 1$ has been violated.

(7) CRT MUST BE GREATER THAN 0 AND LESS THAN 1

This message indicates that the condition $0 < C_{R_t}/\ell < 1$ has been violated.

(8) XRLE MUST BE LESS THAN XRTE

This message indicates that the condition $X_{r\ell e} < X_{rte}$ has been violated.

(9) ANGLE MUST BE BETWEEN 0 DEGREES AND 90 DEGREES

This message indicates that the condition $0 < \text{ANGLE} < 90$ has been violated.

(10) AXIAL LOCATION OF BODY BASE MUST BE AT OR BEHIND POINT WHERE WING TRAILING EDGE PIERCES BODY SURFACE

This message indicates that the condition $X_{r\ell e_1} < X_b$ has been violated.

(11) AXIAL LOCATION OF BODY BASE MUST BE AT OR BEHIND TRAILING EDGE OF WING TIP CHORD

This message indicates that the condition $X_{sm_2} < X_b$ has been violated.

(12) RATIO OF MAJOR TO MINOR AXIS MUST BE GREATER THAN 0

This message indicates that the condition $\lambda(=a/b) > 0$ has been violated.

(13) TAPER RATIO MUST BE BETWEEN 0 AND 1

This message indicates that the condition $0 \leq \text{TR} \leq 1$ has been violated.

The following error messages, numbers (14) through (18) should not occur in the present programs. If they do, they are probably caused by an error in reproducing the source decks.

(14) EXECUTION TERMINATED BECAUSE EXPONENT N CANNOT BE DETERMINED TO WITHIN .01 PERCENT IN 20 ITERATIONS

This message is printed by the circular body programs when the exponent n describing the equivalent body ordinates (see eqs. (11) to (14)) cannot be found accurately from information regarding the point of maximum radius (eqs. (12) or (16)) by using a Newton-Raphson procedure 20 times.

(15) EXECUTION TERMINATED BECAUSE POINT WHERE WING LEADING EDGE PIERCES BODY CANNOT BE FOUND TO WITHIN .01 PERCENT IN 20 ITERATIONS

This message is printed when the point $X_{r\ell e_1}/\ell$ cannot be found accurately by using a Newton-Raphson iteration procedure 20 times.

(16) EXECUTION TERMINATED BECAUSE POINT WHERE WING TRAILING EDGE PIERCES BODY CANNOT BE FOUND TO WITHIN .01 PERCENT IN 20 ITERATIONS

This message is printed when the point X_{rte_1}/ℓ cannot be found accurately by using a Newton-Raphson iteration procedure 20 times.

(17) EXECUTION TERMINATED BECAUSE SEB"(X)=0 POINT CANNOT BE DETERMINED TO WITHIN SUFFICIENT ACCURACY IN 10 ITERATIONS

This message is printed by the circular body programs when the point x_s/ℓ cannot be found accurately by using a Newton-Raphson iteration procedure 10 times.

(18) INTEGRATION TERMINATED BECAUSE ACCUMULATED ERRORS HAVE CAUSED INTEGRATION SUBROUTINE TO BISECT ORIGINAL STEP SIZE (.001) 10 TIMES

This message is printed when the integration subroutine used (Hammings modified predictor-corrector scheme as described in the Scientific Subroutine Package from IBM Corporation) cannot achieve the integration accuracy (DPRMT(4)) desired even though the original step size (DPRMT(3) = 0.001) has been bisected 10 times.

(19) PROGRAM TERMINATED BECAUSE INDENTED BODY RADIUS HAS BECOME LESS THAN ZERO AT x/ℓ =

This message is printed by the program when the cross-sectional area of the wing is larger than that of the equivalent body, so that the indented body radius is less than zero. A wing with smaller thickness/chord ratio or smaller span must be used.

(20) PROGRAM TERMINATED BECAUSE INDENTED BODY MAJOR AXIS HAS BECOME LESS THAN ZERO AT x/ℓ =

This message is printed by the program for the same reason as message (20).

(21) CALCULATION TERMINATED BECAUSE FLOW ABOUT EQUIVALENT BODY HAS BECOME SUPERSONIC AT x/ℓ = . INPUT MACH NUMBER GREATER THAN LOWER CRITICAL

This message is printed when using the local linearization method for purely subsonic flow to calculate u_B and indicates that the free stream Mach number is greater than the lower critical. There exists a region of supersonic flow and the local linearization method does not apply.

(22) CALCULATION TERMINATED BECAUSE FLOW ABOUT EQUIVALENT BODY HAS BECOME SUBSONIC AT x/ℓ =

This message is printed using the local linearization method for purely supersonic flow to calculate u_B and it indicates that a region of subsonic flow has been encountered. This always occurs near the tail of the equivalent bodies considered here and the phenomenon is discussed fully in reference 6. However, if this region occurs at or near the nose, the local linearization method does not apply. In this case, the following error message is also printed,

INPUT MACH NUMBER LESS THAN UPPER CRITICAL

indicating that the condition $M_{cr,u} < M_\infty$ has been violated.

(23) START OF SUPERSONIC CALCULATION
SUPERSONIC CALCULATION STARTS AT x/ℓ =

This message is printed when using the local linearization method for $M_\infty \approx 1$ flows to calculate u_B and indicates where transfer is made from the parabolic (eq. (5)) to the hyperbolic (eq. (7)) differential equation. See reference 6 for details.

(24) START OF SUBSONIC CALCULATION
SUBSONIC CALCULATION STARTS AT x/ℓ =

This message is printed when using the local linearization method for $M_\infty \approx 1$ flows to calculate u_B and indicates where transfer is made from the hyperbolic (eq. (7)) to the elliptic (eq. (6)) differential equation. See reference 6 for details.

APPENDIX A

NUMERICAL EXAMPLES

General Description of the Output

The output format of the program developed is as follows. On the top of the first page, a heading is printed describing the Mach number range, the general class of body (circular or elliptic) and wing being considered, and the theory used. Next, the wing-body geometry and flow field characteristics are printed. Then, if the local linearization method is being used, the program prints the fact that the integrations are starting at $x = x_s$ and proceeding to the nose. If the user supplies the distribution of u_B, this is omitted. A heading of independent and dependent variables is printed next which contains, from left to right, the axial location x/ℓ at which output is to be given, the actual body radius R_b/ℓ (or actual semimajor axis a/ℓ in the case of elliptic bodies) at that axial location, the angles θ (in degrees) in the crossflow plane at which output is desired, the surface pressure coefficient C_p(body), and six flow-field pressure coefficients $C_p(r/D = \ )$ at the indicated distances r/D in the crossflow plane. For the case when the user supplies the u_B distribution, the calculation begins at a point close to the nose and proceeds toward the body base with the values of the above quantities being printed out in the indicated tabular form at specified axial locations. When the local linearization method is used to determine u_B, the calculation begins at $x = x_s$ and proceeds to a point close to the nose, with the quantities described being printed at the specified axial locations, then, when the point close to the nose is reached, the program returns to $x = x_s$, prints the fact that the integration is restarting at that point and proceeding to the tail, prints the independent and dependent variable heading described above, and proceeds with the calculation to the body base. If it should happen at some point that the radial distance in the crossflow plane at which output is desired is less than the body radius (i.e. the point is inside the body), or if an output point falls on the wing leading or trailing edge, the pressure coefficient at that point is set equal to 1.E + 6 and the program continues. Also, because of the discontinuity in the second derivative of the indented body d^2R_b/dx^2 (or d^2a/dx^2) at the points $x/\ell = X_{r\ell e_1}$ and $x/\ell = X_{rte_1}$

and the discontinuity in the slope of the wing span at the points $x/\ell = X_{sm_1}$, and $x/\ell = X_{sm_2}$, output is not printed within a band ($\Delta x/\ell = 0.005$) of those points. When the calculations are successfully completed, the pertinent aerodynamic coefficients are calculated and printed and the program then proceeds to read the data for the next case.

Sample Cases

In order to provide checks on the programs, sample test cases have been run for each program and the results are provided in figures 8 through 12. In each case, the input data is provided together with the corresponding output.

Finally, we note that in order to improve their accuracy, changes have been made in the subroutines, as given in reference 6, which compute the derivatives of the indented circular and elliptic body area distributions. Consequently, the test case results that appear in reference 6 will differ slightly from the results which the current program will produce for those same wing-body geometries.

LISTING OF COMPUTER PROGRAM

```
C     PROGRAM FOR DETERMINING THE SURFACE AND FLOW FIELD PRESSURE DISTRI
C     BUTIONS AT TRANSONIC SPEEDS ON NONLIFTING AND LIFTING WING-INDENTE
C     D BODY COMBINATIONS WITH THE BODY BEING EITHER (1) CIRCULAR IN CRO
C     SS SECTION AND SUCH THAT THE EQUIVALENT BODY OF REVOLUTION HAS ORD
C     INATES R THAT ARE EITHER USER-SUPPLIED OR ARE PROPORTIONAL TO X/L-
C     (X/L)**N OR 1-X/L-(1-X/L)**N OR (2) ELLIPTIC IN CROSS SECTION AND
C     SUCH THAT THE CROSS SECTIONS MAINTAIN A CONSTANT RATIO OF MAJOR TO
C     MINOR AXES ALONG THE ENTIRE BODY LENGTH WITH THE EQUIVALENT BODY O
C     F REVOLUTION HAVING ORDINATES R DESCRIBED ABOVE,THE WING HAVING A
C     CONSTANT THICKNESS/CHORD RATIO, TAPER RATIO BETWEEN 0 AND 1, WITH
C     THE ROOT CHORD EXTENDING FROM X/L =XRLE TO X/L = XRTE, AND WITH OR
C     DINATES Z PROPORTIONAL TO XBAR/C-(XBAR/C)**M OR 1-XBAR/C-(1-XBAR/C
C     )**M BY USING THE TRANSONIC EQUIVALENCE RULE AND EITHER A USER-SUP
C     PLIED VELOCITY DISTRIBUTION UEB(X) ON THE EQUIVALENT BODY SURFACE
C     OR DETERMINING UEB(X) BY THE LOCAL LINEARIZATION METHOD.  FOR REFE
C     RENCE SEE SPREITER,J.R. AND STAHARA,S.S.,NASA CR-1722, FEB.1971.
C
C
C*********************************************************************
C
C
C     AL=RATIO OF SEMIMAJOR/SEMINOR AXIS OF ELLIPTIC CROSS SECTION
C     AMACH=FREE STREAM MACH NUMBER
C     ALPHA=ANGLE OF ATTACH IN DEGREES
C     XLOUTP=INTERVAL SIZE AS FRACTION OF COMPLETE BODY LENGTH FOR
C     PRESSURE DISTRIBUTION PRINTOUT
C     NTHETA=NUMBER OF ANGULAR POSITIONS IN THE CROSSFLOW PLANE AT WHICH
C     PRESSURE DISTRIBUTION PRINTOUT IS DESIRED
C     THETA(I)=NTHETA DIMENSIONAL VECTOR REPRESENTING NTHETA VALUES OF
C     THE AZIMUTHAL ANGLE THETA IN THE CROSSFLOW PLANE (IN DEGREES)
C     RF(I)=SIX DIMENSIONAL VECTOR REPRESENTING SIX VALUES OF THE RADIAL
C     DISTANCE R IN THE CROSSFLOW PLANE NORMALIZED BY THE MAXIMUM EQUIVA
C     LENT BODY DIAMETER D AT WHICH PRESSURE DISTRIBUTION PRINTOUT IS
C     DESIRED
C     XMTB=LOCATION AS FRACTION OF COMPLETE BODY LENGTH OF POSITION OF
C     MAXIMUM EQUIVALENT BODY THICKNESS
C     TAUB=THICKNESS RATIO OF EQUIVALENT BODY
C     XMTW=LOCATION AS FRACTION OF DISTANCE FROM LEADING EDGE TO LOCAL
C     CHORD LENGTH (XBAR/C) OF POSITION OF WING MAXIMUM THICKNESS
C     TAUW=THICKNESS/CHORD RATIO OF WING
C     XRLE=LOCATION AS FRACTION OF COMPLETE BODY LENGTH OF POSITION OF
C     LEADING EDGE OF WING ROOT CHORD
C     XLBASE=LOCATION AS FRACTION OF COMPLETE BODY LENGTH OF POSITION OF
C     BODY BASE
C     ANGLE=LEADING EDGE SWEEP ANGLE IN DEGREES
C     CRT=WING ROOT CHORD AS FRACTION OF COMPLETE BODY LENGTH
C     TR=WING PLANFORM TAPER RATIO
C     SSMAX=WING SEMISPAN AS FRACTION OF COMPLETE BODY LENGTH
C     ICOPY=INTEGER INDEX FOR PROGRAM OPTION FOR USING PREVIOUSLY-STORED
C     CROSSFLOW CALCULATIONS
C     ICOPY=0--PROGRAM CALCULATES CROSSFLOW SOLUTIONS AND STORES THEM
C     ICOPY=1--PROGRAM USES PREVIOUSLY-STORED CROSSFLOW SOLUTIONS
C     MAREA=INTEGER INDEX FOR PROGRAM OPTION FOR EQUIVALENT BODY PROFILE
C     MAREA=0--PROGRAM SUPPLIES EQUIVALENT BODY AREA SUBROUTINES
C     MAREA=1--USER SUPPLIES EQUIVALENT BODY AREA SUBROUTINES
C     XS2EB=AXIAL LOCATION AS FRACTION OF COMPLETE BODY LENGTH WHERE USE
C     R-SUPPLIED EQUIVALENT BODY AREA DISTRIBUTION SATISFIES SEB(X)=0
C     NXEB=INTEGER INDICATING NUMBER OF TABLE ENTRIES OF THE USER-SUPPLI
C     ED DISTRIBUTION OF UB.  MAX VALUE OF NXEB = 201
C     XEB(I)=NXEB-DIMENSIONAL VECTOR REPRESENTING VALUES OF THE AXIAL LO
C     CATIONS AS FRACTIONS OF THE COMPLETE BODY LENGTH WHERE VALUES FOR
C     THE USER-SUPPLIED DISTRIBUTION UB ARE GIVEN
C     UEB(I)=NXEB-DIMENSIONAL VECTOR REPRESENTING VALUES OF THE USER-SUP
C     PLIED DISTRIBUTION UB
C     MOPT=INTEGER INDEX FOR PROGRAM OPTION FOR CALCULATION OF LONGITUDI
C     NAL PERTURBATION VELOCITY COMPONENT, UB(X), ON NONLIFTING EQUIVALE
C     NT BODY SURFACE
C     MOPT=1--PROGRAM USES LOCAL LINEARIZATION METHOD (LLM) FOR FREE STR
C     EAM MACH NUMBERS AT OR NEAR ONE TO CALC. UEB(X)
C     MOPT=2--PROGRAM USES LLM FOR PURELY SUBSONIC FLOW TO CALC. UEB(X)
C     MOPT=3--PROGRAM USES LLM FOR PURELY SUPERSONIC FLOW TO CALC.UEB(X)
C     MOPT=4--PROGRAM USES USER-SUPPLIED DISTRIBUTION OF UEB(X)
C
C
C*********************************************************************
C
C
C     MAIN PROGRAM
      DIMENSION DPRMT(5),DY(1),DDERY(1),DAUX(16,1)
      DIMENSION XMTKS(19),DNEXP(19),DZS1(19)
      DIMENSION THETA(5),THETA1(5),THETA2(5),RF(7),YN1(5),ZN1(5),SCS(5),
     1RF1(7),RFF(7),THET(5),XEB(201),UEB(201)
      NAMELIST /TRANIN/ AMACH,ALPHA,XLOUTP,NTHETA,THETA,RF,AL,XMTB,XMTW,
     1TAUB,TAUW,XRLE,XLBASE,ANGLE,CRT,TR,SSMAX,XS2EB,MOPT,MAREA,NXEB,
     2ICOPY
      EXTERNAL FCT,OUTP
      COMMON /BLK1/ DM,DK,D1,D1M,D2M,D2S,DU1,D2A0,DV,DV3,DTAU2
      COMMON /BLK2/ DX11,DX12,DUU1,DUU2,DCD,ALPHA1,THET,THETA1,THETA2,
     1RFF,YN1,ZN1,SCS,DWRTE,DMRM,MMM,M11,NTHET,ICL,ICOP
      COMMON /BLK3/ DNW,DNW1,DNW2,XTEMLE,XRL,XRT,TBLE,TBTE,TBLMTE,YCMT
     1,XLSM1,ALSM2,SMAX,DAIR,MTE,MTR
      COMMON /BLK4/ ALL,AL12,AL1,AL2,AL3,AL4,M,MOP,MELLPT
      COMMON /BLK5/ XRLE1,XRTE1,HH,H3,NNN
      COMMON /BLK6/ PI,PI2,TOL,NIT11
      COMMON /BLK7/ DCTAU2,DN,K
      COMMON /BLK8/ DTOL
      COMMON /BLK9/ MM
      COMMON /BLK19/ XEB,UEB,NXEB,ITEST
      DATA XMTKS/0.05,.10,.15,.20,.25,.30,.35,.40,.45,.50,.55,.60,.65,
     1           .70,.75,.80,.85,.90,.95/
      DATA DNEXP/88.731,34.649,19.173,12.215,8.396,6.044,4.482,3.389,
     1           2.595,2.0,2.595,3.389,4.482,6.044,8.396,12.215,19.173,
     2           34.649,88.731/
      DATA DZS1/0.007713,0.019162,0.033578,0.050724,0.070564,0.093097,
     1          0.118356,0.146406,0.177347,0.211325,0.272310,0.337355,
     2          0.406294,0.479059,0.555659,0.636159,0.720673,0.809353,
     3          0.902386/
   10 AL=1.
      RF(1)=1.
      DO 333 I=2,7
  333 RF(I)=RF(I-1)+1.
      ALPHA=0.
      XLOUTP=.01
      NTHETA=2
      THETA(1)=0.
      THETA(2)=90.
      ICOPY=0
      MAREA=0
      ICOPY=0
      READ (5,TRANIN)
      MOP=MOPT
      NTHET=NTHETA
      XRL=XRLE
      ALL=AL
      ICOP=ICOPY
      MELLPT=0
      IF(ABS(AL-1.).GT.1.E-2) MELLPT=1
      IF(MELLPT.EQ.1) GO TO 590
      IF(MOPT.NE.4) WRITE (6,600)
      IF(MOPT.EQ.4) WRITE (6,599)
  600 FORMAT (1H1,35X60HCALCULATION OF SURFACE AND FLOW FIELD PRESSURE D
     1ISTRIBUTIONS / 36X58HFOR FLOW AT FREE STREAM MACH NUMBERS AT OR NE
     2AR ONE, BELOW /36X62HTHE LOWER CRITICAL,OR ABOVE THE UPPER CRITICA
     3L ABOUT A FINITE / 36X58HTHICKNESS WING-INDENTED CIRCULAR BODY COM
     4BINATION WITH THE / 36X61HEQUIVALENT BODY OF REVOLUTION EITHER USE
     5R-SPECIFIED OR HAVING / 36X61HORDINATES R PROPORTIONAL TO X/L-(X/L
     6)**N OR 1-X/L-(1-X/L)**N, / 36X61HTHE WING HAVING A CONSTANT THICK
     7NESS/CHORD RATIO, TAPER RATIO / 36X60HBETWEEN 0 AND 1, AND WITH OR
     8DINATES Z PROPORTIONAL TO XBAR/C / 36X61H-(XBAR/C)**M OR 1-XBAR/C-
     9(1-XBAR/C)**M BY USING THE TRANSONIC / 36X51HEQUIVALENCE RULE AND
     1THE LOCAL LINEARIZATION METHOD ///)
  599 FORMAT (1H1,35X60HCALCULATION OF SURFACE AND FLOW FIELD PRESSURE D
     1ISTRIBUTIONS / 36X57HFOR TRANSONIC FLOW ABOUT A FINITE THICKNESS W
     2ING-INDENTED / 36X61HCIRCULAR BODY COMBINATION WITH THE EQUIVALENT
```

```
3 BODY OF REVOLU-/ 36X61HTION EITHER USER-SPECIFIED OR HAVING ORDINMAIN1145
4ATES R PROPORTIONAL / 36X59HTO X/L-(X/L)**N OR 1-X/L-(1-X/L)**N, TMAIN1146
5HE WING HAVING A CON- / 36X61HSTANT THICKNESS/CHORD RATIO, TAPER RMAIN1147
6ATIO BETWEEN 0 AND 1, AND / 36X61HWITH ORDINATES Z PROPORTIONAL TOMAIN1148
7 XBAR/C-(XBAR/C)**M OR 1-XBAR/ 36X60H/C-(1-XBAR/C)**M BY USING THEMAIN1149
8 TRANSONIC EQUIVALENCE RULE AND/ 36X62HA USER SUPPLIED VELOCITY DIMAIN1150
9STRIBUTION UEB(X) ON THE EQUIVALENT / 36X12HBODY SURFACE ///)      MAIN1151
  GO TO 336                                                         MAIN1152
590 CONTINUE                                                        MAIN1153
    IF(MOPT.NE.4) WRITE (6,597)                                     MAIN1154
    IF(MOPT.EQ.4) WRITE (6,596)                                     MAIN1155
597 FORMAT (1H1,35X60HCALCULATION OF SURFACE AND FLOW FIELD PRESSURE DMAIN1156
   1ISTRIBUTIONS / 36X58HFOR FLOW AT FREE STREAM MACH NUMBERS AT OR NEMAIN1157
   2AR ONE, BELOW / 36X62HTHE LOWER CRITICAL, OR ABOVE THE UPPER CRITIMAIN1158
   3CAL ABOUT A FINITE / 36X61HTHICKNESS WING-INDENTED BODY COMBINATIOMAIN1159
   4N WITH THE BODY HAVING / 36X60HAN ELLIPTIC CROSS SECTION THAT MAINMAIN1160
   5TAINS A CONSTANT RATIO OF / 36X60HMAJOR/MINOR AXES ALONG THE ENTIRMAIN1161
   6E BODY LENGTH WITH THE EQUI- / 36X61HVALENT BODY OF REVOLUTION EITMAIN1162
   7HER USER-SPECIFIED OR HAVING OR- / 36X60HDINATES R PROPORTIONAL TOMAIN1163
   8 X/L-(X/L)**N OR 1-X/L-(1-X/L)**N, / 36X61HTHE WING HAVING A CONSTMAIN1164
   9ANT THICKNESS/CHORD RATIO, TAPER RATIO / 36X60HBETWEEN 0 AND 1, ANMAIN1165
   1D WITH ORDINATES Z PROPORTIONAL TO XBAR/C / 36X61H-(XBAR/C)**M OR MAIN1166
   21-XBAR/C-(1-XBAR/C)**M BY USING THE TRANSONIC / 36X51HEQUIVALENCE MAIN1167
   3RULE AND THE LOCAL LINEARIZATION METHOD ///)                    MAIN1168
596 FORMAT (1H1,35X60HCALCULATION OF SURFACE AND FLOW FIELD PRESSURE DMAIN1169
   1ISTRIBUTIONS / 36X57HFOR TRANSONIC FLOW ABOUT A FINITE THICKNESS WMAIN1170
   2ING-INDENTED / 36X60HBODY COMBINATION WITH THE BODY HAVING AN ELLIMAIN1171
   3PTIC CROSS SEC- / 36X62HTION THAT MAINTAINS A CONSTANT RATIO OF MAMAIN1172
   4JOR/MINOR AXES ALONG / 36X61HTHE ENTIRE BODY LENGTH WITH THE EQUIVMAIN1173
   5ALENT BODY OF REVOLUTION / 36X59HEITHER USER-SPECIFIED OR HAVING OMAIN1174
   6RDINATES R PROPORTIONAL TO / 36X60HX/L-(X/L)**N OR 1-X/L-(1-X/L)**MAIN1175
   7N, THE WING HAVING A CONSTANT / 36X60HTHICKNESS/CHORD RATIO, TAPERMAIN1176
   8 RATIO BETWEEN 0 AND 1, AND WITH / 36X59HORDINATES Z PROPORTIONAL MAIN1177
   9TO XBAR/C-(XBAR/C)**M OR 1-XBAR/C- / 36X59H(1-XBAR/C)**M BY USING MAIN1178
   1THE TRANSONIC EQUIVALENCE RULE AND A / 36X60HUSER-SUPPLIED VELOCITMAIN1179
   2Y DISTRIBUTION UEB(X) ON THE EQUIVALENT / 36X12HBODY SURFACE ///) MAIN1180
336 DM=AMACH                                                        MAIN1181
    AX=XLOUIP                                                       MAIN1182
    XMT=XMTB                                                        MAIN1183
    XCMT=XMIN                                                       MAIN1184
    SMAX=SSMAX                                                      MAIN1185
    XRTE=CRT+XRLE                                                   MAIN1186
    XHT=XRTE                                                        MAIN1187
    IF(DX.LE..0.OR.DX.GE.1.) GO TO 307                              MAIN1188
    IF(AL.LE.0.) GO TO 315                                          MAIN1189
    IF(XMT.GE.1..OR.XMT.LE.0.) GO TO 303                            MAIN1190
    IF(TAUB.LT.0.) GO TO 304                                        MAIN1191
    IF(XCMT.GE.1..OR.XCMT.LE.0.) GO TO 308                          MAIN1192
    IF(TAUW.LT.0.) GO TO 309                                        MAIN1193
    IF(XRLE.LT.0..OR.XRLE.GT.1.) GO TO 310                          MAIN1194
    IF(CRT.LT.0..OR.CRT.GT.1.) GO TO 311                            MAIN1195
    IF(XRTE.LE.XRLE) GO TO 312                                      MAIN1196
    IF(ANGLE.LT.0..OR.ANGLE.GT.90.) GO TO 313                       MAIN1197
    IF(TR.LT..0.OR.TR.GT.1.) GO TO 314                              MAIN1198
    PI=3.1415927                                                    MAIN1199
    PI2=6.2831854                                                   MAIN1200
    TOL=1.E-2                                                       MAIN1201
    NIT11=10                                                        MAIN1202
    HH=.001                                                         MAIN1203
    H3=3.*HH                                                        MAIN1204
    NNN=0                                                           MAIN1205
    M11=0                                                           MAIN1206
    DTAU=TAUB                                                       MAIN1207
    DTAU2=DTAU*DTAU                                                 MAIN1208
    AL12=SQRT(AL)                                                   MAIN1209
    GAMMA=1.4                                                       MAIN1210
    DX=DM*DM*2.4                                                    MAIN1211
    D1=DK*1.7010724                                                 MAIN1212
    D1M=(DM*DM-1.00)                                                MAIN1213
    D2M=D1M/DK                                                      MAIN1214
    DTOL=1.E-3                                                      MAIN1215
    DZ0=.005                                                        MAIN1216
    DZF=XLBASE                                                      MAIN1217
    DX12=0.                                                         MAIN1218
    DUU2=0.                                                         MAIN1219
    IC=1                                                            MAIN1220
    ICC=1C                                                          MAIN1221
    M=1                                                             MAIN1222
    MM=1                                                            MAIN1223
    MTR=0                                                           MAIN1224
    IF(TR.GT.1.E-2) MTR=1                                           MAIN1225
    ANGLE1=ANGLE*PI/180.                                            MAIN1226
    TBLE=TAN(ANGLE1)                                                MAIN1227
    CTIP=TR*CRT                                                     MAIN1228
    XLSM1=SSMAX*TBLE+XRLE                                           MAIN1229
    XLSM2=XLSM1+CTIP                                                MAIN1230
    IF(XLBASE.LT.XLSM2) GO TO 563                                   MAIN1231
    XLDET=XLSM2-XRTE                                                MAIN1232
    IF(ABS(XLDET).LE.1.E-5) GO TO 1000                              MAIN1233
    IF(XLDET.LT.0.) GO TO 1001                                      MAIN1234
    TBTE=XLDET/SSMAX                                                MAIN1235
    ANGTE1=ATAN(TBTE)                                               MAIN1236
    ANGTE=180.*ANGTE1/PI                                            MAIN1237
    MTE=1                                                           MAIN1238
    ALPHA=0.                                                        MAIN1239
    GO TO 1002                                                      MAIN1240
1000 ANGTE1=0.                                                      MAIN1241
    ANGTE=0.                                                        MAIN1242
    MTE=0                                                           MAIN1243
    TBTE=0.                                                         MAIN1244
    GO TO 1002                                                      MAIN1245
1001 TBTE=-XLDET/SSMAX                                              MAIN1246
    ANGTE1=-ATAN(TBTE)                                              MAIN1247
    ANGTE=180.*ANGTE1/PI                                            MAIN1248
    MTE=2                                                           MAIN1249
1002 XTEHLE=CRT                                                     MAIN1250
    TBLMTE=TBLE-TBTE                                                MAIN1251
    ALPHA1=ALPHA*PI/180.                                            MAIN1252
    DMM=DX                                                          MAIN1253
    IF(MAREA.EQ.1) GO TO 3                                          MAIN1254
    K=1                                                             MAIN1255
    IF (XMT.GT.0.5) K=0                                             MAIN1256
    DO 913 I=1,19                                                   MAIN1257
    J=I-1                                                           MAIN1258
    IF (XMTKS(I)-XMT) 913,914,951                                   MAIN1259
913 CONTINUE                                                        MAIN1260
914 DN=DNEXP(I)                                                     MAIN1261
    DZS1=DZST(I)                                                    MAIN1262
    GO TO 950                                                       MAIN1263
951 IF (I.EQ.1) GO TO 914                                           MAIN1264
    DN=DNEXP(J)+(DNEXP(I)-DNEXP(J))/(XMTKS(I)-XMTKS(J))*(XMT-XMTKS(J))MAIN1265
    DZS1=DZST(J)+(DZST(I)-DZST(J))/(XMTKS(I)-XMTKS(J))*(XMT-XMTKS(J)) MAIN1266
950 CONTINUE                                                        MAIN1267
    A=XMT                                                           MAIN1268
    IF(XMT.LT..5) A=1.-XMT                                          MAIN1269
    NIT=0                                                           MAIN1270
  5 DN1=DN-1.                                                       MAIN1271
    NIT=NIT+1                                                       MAIN1272
    IF(NIT.GT.20) GO TO 6                                           MAIN1273
    AN1=A**DN1                                                      MAIN1274
    DNUM=1.-DN*AN1                                                  MAIN1275
    DENUM=-AN1*(DN*ALOG(A)+1.)                                      MAIN1276
    DNEWN=DN-(DNUM/DENUM)                                           MAIN1277
    ERR=ABS(DN/DNEWN-1.)                                            MAIN1278
    DN=DNEWN                                                        MAIN1279
    IF(ERR.GT.1.E-4) GO TO 5                                        MAIN1280
    GO TO 8                                                         MAIN1281
  6 WRITE (6,7)                                                     MAIN1282
  7 FORMAT (4X49HEXECUTION TERMINATED BECAUSE EXPONENT N CANNOT BE /5XMAIN1283
   149HDETERMINED TO WITHIN .01 PERCENT IN 20 ITERATIONS)           MAIN1284
    GO TO 10                                                        MAIN1285
  8 DC=(DN**(DN/(DN-1.)))/(DN-1.)                                   MAIN1286
    DCTAU=DC*DTAU                                                   MAIN1287
    DCTAU2=DCTAU*DCTAU                                              MAIN1288
```

```
C
C     CALCULATION OF THE POINT WHERE SEB''(X) = 0
C
      N=1
    1 DZS=DZS1-S2EBPI(DZS1)/S3EBPI(DZS1)
      IF(ABS(DZS-DZS1).LT.1.E-4) GO TO 3
      DZS1=DZS
      N=N+1
      IF(N.GT.11) GO TO 2
      GO TO 1
    2 WRITE (6,100)
  100 FORMAT (1H0,4X57HEXECUTION TERMINATED BECAUSE SEB''(X) = 0 POINT C
     1ANNOT BE /5X57HDETERMINED TO WITHIN SUFFICIENT ACCURACY IN 10 ITE
     2RATIONS )
      GO TO 10
    3 IF(MELLPT.NE.1) GO TO 569
      AL1=(AL*AL-1.)/(AL*AL)
      AL2=(AL+1.)/(AL-1.)
      AL3=(AL-1.)*(AL-1.)/(AL*AL)
      AL4=.5*(1.+1./AL)
  569 XI=XRLE
      DO 20 I=1,20
      RBXI=AL12*SQRT(4.*SEBPI(XI))
      DRBXI=AL*2.*S1EBPI(XI)/RBXI
      XIN=XI-(SPANLE(XI)-RBXI)/(DSPLE(XI)-DRBXI)
      ERR=ABS(XIN/XI-1.)
      XI=XIN
      IF(ERR.LE.1.E-4) GO TO 555
   20 CONTINUE
  555 XRLE1=XI
      IF(I.GT.19) GO TO 556
      IF(MTE.EQ.0) GO TO 559
      XI=XRTE
      DO 22 I=1,20
      RBXI=AL12*SQRT(4.*SEBPI(XI))
      DRBXI=AL*2.*S1EBPI(XI)/RBXI
      XIN=XI-(SPANTE(XI)-RBXI)/(DSPTE(XI)-DRBXI)
      ERR=ABS(XIN/XI-1.)
      XI=XIN
      IF(ERR.LE.1.E-4) GO TO 557
   22 CONTINUE
  557 XRTE1=XI
      IF(I.GT.19) GO TO 558
      GO TO 559
  556 WRITE (6,560)
  560 FORMAT (1H0,4X66HEXECUTION TERMINATED BECAUSE POINT WHERE WING LEA
     1DING EDGE PIERCES / 5X59HBODY CANNOT BE FOUND TO WITHIN .01 PERCEN
     2T IN 20 ITERATIONS )
      GO TO 10
  558 WRITE (6,561)
  561 FORMAT (1H0,4X67HEXECUTION TERMINATED BECAUSE POINT WHERE WING TRA
     1ILING EDGE PIERCES / 5X59HBODY CANNOT BE FOUND TO WITNIN .01 PERCE
     2NT IN 20 ITERATIONS )
      GO TO 10
  559 CONTINUE
      IF(MTE.EQ.0) XRTE1=XLSM2
      DO 699 I=1,NTHETA
      THETA1(I)=THETA(I)
      THETA(I)=THETA1(I)*PI/180.
      THET(I)=THETA(I)
      THETA2(I)=-THETA1(I)
      YN1(I)=COS(THETA(I))
      IF(ABS(YN1(I)).LE.1.E-4) YN1(I)=0.
      ZN1(I)=SIN(THETA(I))
      IF(ABS(ZN1(I)).LE.1.E-4) ZN1(I)=0.
      SCS(I)=SQRT(YN1(I)*YN1(I)+AL*AL*ZN1(I)*ZN1(I))
  699 CONTINUE
      DO 813 I=1,19
      J=I-1
      IF(XMTKS(I)-XMTW) 813,814,851
  813 CONTINUE
  814 DNW=DNEXP(I)
      GO TO 850
  851 IF(I.EQ.1) GO TO 814
      DNW=DNEXP(J)+(DNEXP(I)-DNEXP(J))/(XMTKS(I)-XMTKS(J))*(XMT-XMTKS(J
     1)
  850 CONTINUE
      AW=XMTW
      IF(XMTW.LT..5) AW=1.-XMTW
      NIT=0
   51 DNW1=DNW-1.
      NIT=NIT+1
      IF(NIT.GT.20) GO TO 61
      ANW1=AW**DNW1
      DNUMW=1.-DNW*ANW1
      DENOMW=-ANW1*(DNW*ALOG(AW)+1.)
      DNEWNW=DNW-(DNUMW/DENOMW)
      ERRW=ABS(DNW/DNEWNW-1.)
      DNW=DNEWNW
      IF(ERRW.GT.1.E-4) GO TO 51
      GO TO 81
   61 WRITE (6,71)
   71 FORMAT (4X49HEXECUTION TERMINATED BECAUSE EXPONENT M CANNOT BE/
     15X49HDETERMINED TO WITHIN .01 PERCENT IN 20 ITERATIONS)
      GO TO 10
   81 DNW1=DNW-1.
      DNW2=DNW1-1.
      DAIR=DNW**(DNW/DNW1)
      DAIR=DAIR*TAUW/(2.*DNW1)
      IF(MAREA.EQ.1.AND.MOPT.NE.4) DZS=XSZEB
      WRITE (6,601)
      IF(MELLPT.EQ.1) WRITE (6,622) AL
      WRITE (6,603) TAUB
      IF(MAREA.EQ.1) GO TO 1050
      WRITE (6,602) XMTB
      WRITE (6,604) DN
 1050 IF(MOPT.EQ.4) GO TO 1051
      WRITE (6,605) DZS
 1051 WRITE (6,608) XMTW
      WRITE (6,609) TAUW
      WRITE (6,610) DNW
      WRITE (6,611) XRLE
      WRITE (6,612) XRTE
      WRITE (6,653) TR
      WRITE (6,616) XRLE1
      WRITE (6,617) XRTE1
      WRITE (6,618) XLBASE
      WRITE (6,613) ANGLE
      WRITE (6,614) ANGTE
      IF(TR.GT.1.E-2) GO TO 1003
      WRITE (6,651) XLSM1
      GO TO 1004
 1003 WRITE (6,654) XLSM1
      WRITE (6,655) XLSM2
 1004 WRITE (6,652) SSMAX
      WRITE (6,615) ALPHA
      WRITE (6,606) GAMMA
      WRITE (6,607) DM
      IF(XLBASE.LT.XRTE1) GO TO 382
      DO 701 I=1,6
      RF1(I)=RF(I)
  701 RF(I)=TAUB*RF(I)
      DO 786 I=1,6
      J=8-I
      RF(J)=RF(J-1)
      RFF(J)=RF(J)
  786 CONTINUE
      IF(MOPT.EQ.4) GO TO 542
      DMRM=0.0
      IF (DMM-0.001) 920,920,921
  921 DWRTE=DZS/DMM + 1.0
      MRM=DWRTE
      DWRTE=MRM
      DWRTE=DWRTE*DMM + .001
```

```
      DMRM=-DMM                                                          MAIN1433
  920 CONTINUE                                                           MAIN1434
      IF(MOPT.EQ.2.OR.MOPT.EQ.3) GO TO 496                               MAIN1435
C                                                                        MAIN1436
C     CALCULATION OF AREA AND DERIVATIVES OF AREA AT XS                  MAIN1437
C                                                                        MAIN1438
      DA=SEBPI(DZS)                                                      MAIN1439
      DA1=S1EBPI(DZS)                                                    MAIN1440
      DA2=S2EBPI(DZS)                                                    MAIN1441
      DA3=S3EBPI(DZS)                                                    MAIN1442
      DAM=ALOG(U1*DA/DZS)                                                MAIN1443
      D1N=DA1/DA                                                         MAIN1444
C                                                                        MAIN1445
C     CALCULATION OF INITIAL VALUE OF U                                  MAIN1446
C                                                                        MAIN1447
      DU=-D1N+DINT(DZS)                                                  MAIN1448
C                                                                        MAIN1449
C     CALCULATION OF THE 1ST DERIVATIVE OF U                             MAIN1450
C                                                                        MAIN1451
      D2A0=S2EBPI(0.)                                                    MAIN1452
      D1I=D1INT(DZS)-D2A0/DZS                                            MAIN1453
      DU1=.5                                                             MAIN1454
      N=1                                                                MAIN1455
   13 DU11=DU1-DU1*(DU1-DA3*(ALOG(DU1)+DAM)-D1I)/(DU1-DA3)               MAIN1456
      IF(ABS(DU11-DU1).LT.1.E-4) GO TO 14                                MAIN1457
      DU1=DU11                                                           MAIN1458
      N=N+1                                                              MAIN1459
      IF(N.GT.11) GO TO 14                                               MAIN1460
      GO TO 13                                                           MAIN1461
   14 DU1=DU11                                                           MAIN1462
  496 CONTINUE                                                           MAIN1463
C                                                                        MAIN1464
C     START OF INTEGRATION PROCEDURE                                     MAIN1465
      WRITE (6,101)                                                      MAIN1466
  101 FORMAT (1H0,46HSTART OF INTEGRATION FROM SEB''(X) = 0 TO NOSE //) MAIN1467
      WRITE (6,112) (RFI(I),I=1,6)                                       MAIN1468
  112 FORMAT (4X,3HX/L,3X,7HRBODY/L,3X,10HTHETA(DEG),4X,8HCP(BODY),     MAIN1469
     13X,6(7HCP(R/D=,F5.2,1H)),//)                                       MAIN1470
      IF(MOPT.EQ.2.OR.MOPT.EQ.3) GO TO 495                               MAIN1471
      DY(1)=DU                                                           MAIN1472
      DDERY(1)=1.                                                        MAIN1473
      DPRMT(1)=DZS                                                       MAIN1474
      DPRMT(2)=DZ0                                                       MAIN1475
      DPRMT(3)=-1.E-3                                                    MAIN1476
      DPRMT(4)=1.E-3                                                     MAIN1477
      NDIM=1                                                             MAIN1478
      MMM=0                                                              MAIN1479
      CALL DHPCG(DPRMT,DY,DDERY,NDIM,IHLF,FCT,OUTP,DAUX)                 MAIN1480
      DRAG=DCD                                                           MAIN1481
      IF(IHLF.GT.10) GO TO 10                                            MAIN1482
      IF(MM.GT.1) GO TO 10                                               MAIN1483
      WRITE (6,103)                                                      MAIN1484
  103 FORMAT (1H0,46HSTART OF INTEGRATION FROM SEB''(X) = 0 TO TAIL  //)MAIN1485
      WRITE (6,112) (RFI(I),I=1,6)                                       MAIN1486
      NNN=0                                                              MAIN1487
      DPRMT(1)=DZS                                                       MAIN1488
      DPRMT(2)=DZF                                                       MAIN1489
      DY(1)=DU                                                           MAIN1490
      IF(DMRM.EQ.0.) GO TO 922                                           MAIN1491
      MRM=MRM-1                                                          MAIN1492
      DWRTE=MRM                                                          MAIN1493
      DWRTE=DWRTE*DMM                                                    MAIN1494
      DMRM=DMM                                                           MAIN1495
  922 CONTINUE                                                           MAIN1496
   27 DPRMT(3)=1.E-3                                                     MAIN1497
      DDERY(1)=1.                                                        MAIN1498
      NDIM=1                                                             MAIN1499
      MMM=0                                                              MAIN1500
      CALL DHPCG(DPRMT,DY,DDERY,NDIM,IHLF,FCT,OUTP,DAUX)                 MAIN1501
      IF(IHLF.GT.10) GO TO 10                                            MAIN1502
      IF(MM.EQ.2) GO TO 29                                               MAIN1503
      IF(MM.EQ.3) GO TO 30                                               MAIN1504
  517 DRAG=DRAG+DCD                                                      MAIN1505
      ADRAG=16.*DRAG/(DTAU*DTAU)                                         MAIN1506
      IF(MELLPT.EQ.0) GO TO 587                                          MAIN1507
      REBBAS=SQRT(4.*SEBPI(XLBASE))                                      MAIN1508
      ABB=AL12*REBBAS                                                    MAIN1509
      DS122=2.*S1EBPI(XLBASE)                                            MAIN1510
      DS122=DS122*DS122                                                  MAIN1511
      AL10=SQRT(AL**4-1.)/(AL*AL)                                        MAIN1512
      CALL CEL1(RES,AL10,IER)                                            MAIN1513
      CDF1=8.*DS122*(2.*ALOG(AL4*ABB)*RES/(AL*PI)-ALOG(REBBAS))/(DTAU*   MAIN1514
     1DTAU)                                                              MAIN1515
      ADRAG=ADRAG-CDF1                                                   MAIN1516
  587 IF(M11.EQ.1) GO TO 381                                             MAIN1517
      WRITE (6,386) ADRAG                                                MAIN1518
      GO TO 10                                                           MAIN1519
  381 ALIFT=8.*ALPHA1*DLIFT(XLSM2)/(DTAU*DTAU)                           MAIN1520
      ADRAG=ADRAG+.5*ALPHA1*ALIFT                                        MAIN1521
      AMOMNT=-XLSM2*ALIFT+8.*ALPHA1*DMOMNT(XLSM2)/(DTAU*DTAU)            MAIN1522
      WRITE (6,383) ADRAG                                                MAIN1523
      WRITE (6,384) ALIFT                                                MAIN1524
      WRITE (6,385) AMOMNT                                               MAIN1525
      GO TO 10                                                           MAIN1526
   29 DY(1)=DV                                                           MAIN1527
      DPRMT(1)=DV3                                                       MAIN1528
      DRAG=DRAG+DCD                                                      MAIN1529
      MM=1                                                               MAIN1530
      IF(M11.EQ.1) GO TO 27                                              MAIN1531
      WRITE (6,199)                                                      MAIN1532
  199 FORMAT (1H0,31HSTART OF SUPERSONIC CALCULATION)                    MAIN1533
      WRITE (6,198) DV3                                                  MAIN1534
  198 FORMAT (1H0, 39HSUPERSONIC CALCULATION STARTS AT X/L = ,E12.5//)   MAIN1535
   41 WRITE (6,112) (RFI(I),I=1,6)                                       MAIN1536
      GO TO 27                                                           MAIN1537
   30 DSL=(DX12-DX11)/(DUU1-DUU2)                                        MAIN1538
      DSH=DSL*(DUU2+D2M)                                                 MAIN1539
      DX3=DX12+DSH                                                       MAIN1540
      DX4=2.D0*DX3-DX12                                                  MAIN1541
      DY(1)=-DUU2-S2EBPI(DX4)*ALOG(SEBPI(DX4)/(DX4*(1.-DX4)))-DINT(DX4) MAIN1542
     1-DINT1(DX4)-2.D0*D2M                                               MAIN1543
      DPRMT(1)=DX4                                                       MAIN1544
      MM=1                                                               MAIN1545
      DRAG=DRAG+DCD                                                      MAIN1546
      IF(M11.EQ.1) GO TO 27                                              MAIN1547
      WRITE(6,205)                                                       MAIN1548
  205 FORMAT(1H0,29HSTART OF SUBSONIC CALCULATION)                       MAIN1549
      WRITE (6,510) DX4                                                  MAIN1550
  510 FORMAT (1H0,37HSUBSONIC CALCULATION BEGINS AT X/L = ,E12.5/)       MAIN1551
      WRITE (6,112) (RFI(I),I=1,6)                                       MAIN1552
      GO TO 27                                                           MAIN1553
  495 CONTINUE                                                           MAIN1554
      DY(1)=0.0                                                          MAIN1555
      DDERY(1)=1.0                                                       MAIN1556
      DPRMT(1)=DZS                                                       MAIN1557
      DPRMT(2)=DZ0                                                       MAIN1558
      DPRMT(3)=-1.E-3                                                    MAIN1559
      DPRMT(4)=1.E-3                                                     MAIN1560
      NDIM=1                                                             MAIN1561
      MMM=0                                                              MAIN1562
      CALL DHPCG(DPRMT,DY,DDERY,NDIM,IHLF,FCT,OUTP,DAUX)                 MAIN1563
      IF(IHLF.GT.10) GO TO 10                                            MAIN1564
      IF(MM.GT.1) GO TO 10                                               MAIN1565
      DRAG=DCD                                                           MAIN1566
      IF(DMRM.EQ.0.0) GO TO 923                                          MAIN1567
      MRM=MRM-1                                                          MAIN1568
      DWRTE=MRM                                                          MAIN1569
      DWRTE=DWRTE*DMM                                                    MAIN1570
      DMRM=DMM                                                           MAIN1571
  923 DY(1)=0.                                                           MAIN1572
      DDERY(1)=1.0                                                       MAIN1573
      DPRMT(1)=DZS                                                       MAIN1574
      DPRMT(2)=DZF                                                       MAIN1575
      DPRMT(3)=1.E-3                                                     MAIN1576
```

```
      MMM=0
      NNN=0
      WRITE (6,103)
      WRITE (6,112) (HF1(I),I=1,6)
      CALL DHPCG(DPRMT,DY,DDERY,NDIM,IHLF,FCT,OUTP,DAUX)
      IF(IHLF.GT.10) GO TO 10
      IF(MM.GT.1) GO TO 10
      GO TO 517
  542 DMRM=0.
      IF(DMM-.001) 1920,1920,1921
 1921 DWRTE=DMM-.001
      DMRM=DMM
 1920 CONTINUE
      READ (5,2001) (XEB(J),J=1,NXEB)
      READ (5,2001) (DEB(J),J=1,NXEB)
 2001 FORMAT (8F10.5)
      WRITE (6,2004)
 2004 FORMAT (1H0)
      WRITE (6,112) (HF1(I),I=1,6)
      ITEST=0
      NXB=NXEB
      XOUT=.005
      NUM=1000.*(XLBASE-XOUT)+2.
      MMM=0
      IHLF=1
      DO 2002 I=1,NUM
      CALL OUTP1(XOUT,DY,DDERY,IHLF,NDIM,DPRMT)
      XOUT=XOUT+.001
      IF(XOUT.GT.XLBASE) GO TO 2003
 2002 CONTINUE
 2003 DRAG=0.
      GO TO 517
  383 FORMAT (1H0,30HDRAG COEFFICIENT =              ,E12.5)
  384 FORMAT (1H ,30HLIFT COEFFICIENT =              ,E12.5)
  385 FORMAT (1H ,30HPITCHING MOMENT COEFFICIENT = ,E12.5)
  386 FORMAT (1H0,19HDRAG COEFFICIENT = ,E12.5)
  601 FORMAT (1H 61HWING-BODY COMBINATION GEOMETRY AND FLOW FIELD CHARAC
     1TERISTICS //)
  602 FORMAT (1H ,43HEQUIVALENT BODY MAXIMUM THICKNESS AT X/L = ,E12.5)
  603 FORMAT (1H ,43HEQUIVALENT BODY THICKNESS RATIO =           ,E12.5)
  604 FORMAT (1H ,43HEXPONENT N FOR EQUIVALENT BODY ORDINATES = ,E12.5)
  605 FORMAT (1H ,43HSEB''(X) = 0 AT X/L =                      ,E12.5)
  606 FORMAT (1H ,43HRATIO OF SPECIFIC HEATS =                  ,E12.5)
  607 FORMAT (1H ,43HFREE STREAM MACH NUMBER =                  ,E12.5)
  608 FORMAT (1H ,43HWING MAX. THICKNESS AT YBAR/C =            ,E12.5)
  609 FORMAT (1H ,43HWING THICKNESS/CHORD RATIO =               ,E12.5)
  610 FORMAT (1H ,43HEXPONENT M FOR WING ORDINATES =            ,E12.5)
  611 FORMAT (1H ,43HLEADING EDGE OF WING ROOT CHORD AT X/L =   ,E12.5)
  612 FORMAT (1H ,43HTRAILING EDGE OF WING ROOT CHORD AT X/L =  ,E12.5)
  613 FORMAT (1H ,43HLEADING EDGE SWEEP ANGLE (DEG) =           ,E12.5)
  614 FORMAT (1H ,43HTRAILING EDGE SWEEP ANGLE (DEG) =          ,E12.5)
  615 FORMAT (1H ,43HANGLE OF ATTACK ALPHA (DEG) =              ,E12.5)
  616 FORMAT (1H ,43HLEADING EDGE PIERCES BODY AT X/L =         ,E12.5)
  617 FORMAT (1H ,43HTRAILING EDGE PIERCES BODY AT X/L =        ,E12.5)
  618 FORMAT (1H ,43HBODY BASE AT X/L =                         ,E12.5)
  622 FORMAT (1H ,43HRATIO OF SEMIMAJOR/SEMIMINOR AXIS =        ,E12.5)
  651 FORMAT (1H ,43HLOCATION OF MAX. SPAN AT X/L =             ,E12.5)
  652 FORMAT (1H ,43HNORMALIZED MAX. SEMISPAN SSMAX/L =         ,E12.5)
  653 FORMAT (1H ,43HPLANFORM TAPER RATIO =                     ,E12.5)
  654 FORMAT (1H ,43HLOCATION OF WINGTIP LEADING EDGE AT X/L =  ,E12.5)
  655 FORMAT (1H ,43HLOCATION OF WINGTIP TRAILING EDGE AT X/L = ,E12.5)
  303 WRITE (6,400)
  400 FORMAT (1H0,43HXMTB MUST BE GREATER THAN 0 AND LESS THAN 1)
      GO TO 10
  304 WRITE (6,401)
  401 FORMAT (1H0,57HEQUIVALENT BODY THICKNESS RATIO MUST BE GREATER THA
     1N ZERO )
      GO TO 10
  307 WRITE (6,403)
  403 FORMAT (/69H INTERVAL SIZE FOR PRESSURE DISTRIBUTION PRINT-OUT MUS
     1T BE GREATER THAN 0 AND LESS THAN 1  )
      GO TO 10
  308 WRITE (6,404)
  404 FORMAT (1H0,43HXMTW MUST BE GREATER THAN 0 AND LESS THAN 1 )
      GO TO 10
  309 WRITE (6,405)
  405 FORMAT (1H0,49HWING THICKNESS/CHORD RATIO MUST BE GREATER THAN 0 )
      GO TO 10
  310 WRITE (6,406)
  406 FORMAT (1H0,43HXRLE MUST BE GREATER THAN 0 AND LESS THAN 1 )
      GO TO 10
  311 WRITE (6,407)
  407 FORMAT (1H0,42HCPT MUST BE GREATER THAN 0 AND LESS THAN 1 )
      GO TO 10
  312 WRITE (6,408)
  408 FORMAT (1H0,27HXRLE MUST BE LESS THAN XRTE )
      GO TO 10
  313 WRITE (6,409)
  409 FORMAT (1H0,40HANGLE MUST BE BETWEEN 0 DEGS AND 90 DEGS )
      GO TO 10
  314 WRITE (6,410)
  410 FORMAT (1H0,35HTAPER RATIO MUST BE BETWEEN 0 AND 1 )
      GO TO 10
  315 WRITE (6,411)
  411 FORMAT (1H0,59HRATIO OF SEMIMAJOR/SEMIMINOR AXES MUST BE GREATER T
     1HAN ZERO )
      GO TO 10
  382 WRITE (6,482)
  482 FORMAT (1H0,60HAXIAL LOCATION OF BODY BASE MUST BE AT OR BEHIND PO
     1INT WHERE / 1X,39HWING TRAILING EDGE PIERCES BODY SURFACE )
      GO TO 10
  583 WRITE (6,483)
  483 FORMAT (1H0,62HAXIAL LOCATION OF BODY BASE MUST BE AT OR BEHIND TR
     1AILING EDGE / 1X,17HOF WING-TIP CHORD )
      GO TO 10
      END


      SUBROUTINE OUTP1(DX,DY,DDERY,IHLF,NDIM,DPRMT)
      DIMENSION DY(1),DDERY(1),DPRMT(5)
      CALL OUTP(DX,DY,DDERY,IHLF,NDIM,DPRMT)
      RETURN
      END


      SUBROUTINE DHPCG(PRMT,Y,DERY,NDIM,IHLF,FCT,OUTP,AUX)
C
C
      DIMENSION PRMT(1),Y(1),DERY(1),AUX(16,1)
      COMMON /BLK9/ MM
      N=1
      IHLF=0
      X=PRMT(1)
      H=PRMT(3)
      PRMT(5)=0.
      DO 1 I=1,NDIM
      AUX(16,I)=0.
      AUX(15,I)=DERY(I)
    1 AUX(1,I)=Y(I)
      IF(H*(PRMT(2)-X))3,2,4
C
C     ERROR RETURNS
    2 IHLF=12
      GO TO 4
    3 IHLF=13
C
C     COMPUTATION OF DERY FOR STARTING VALUES
    4 CALL FCT(X,Y,DERY)
```

```
      IF(MM.GT.1) RETURN                                                DHPCG024
C                                                                       DHPCG025
C     RECORDING OF STARTING VALUES                                      DHPCG026
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                DHPCG027
      IF(PRMT(5))6,5,6                                                  DHPCG028
    5 IF(IHLF)7,7,6                                                     DHPCG029
    6 RETURN                                                            DHPCG030
    7 DO 8 I=1,NDIM                                                     DHPCG031
    8 AUX(8,I)=DERY(I)                                                  DHPCG032
C                                                                       DHPCG033
C     COMPUTATION OF AUX(2,I)                                           DHPCG034
      ISW=1                                                             DHPCG035
      GO TO 100                                                         DHPCG036
C                                                                       DHPCG037
    9 X=X+H                                                             DHPCG038
      DO 10 I=1,NDIM                                                    DHPCG039
   10 AUX(2,I)=Y(I)                                                     DHPCG040
C                                                                       DHPCG041
C     INCREMENT H IS TESTED BY MEANS OF BISECTION                       DHPCG042
   11 IHLF=IHLF+1                                                       DHPCG043
      X=X-H                                                             DHPCG044
      DO 12 I=1,NDIM                                                    DHPCG045
   12 AUX(4,I)=AUX(2,I)                                                 DHPCG046
      H=.5D0*H                                                          DHPCG047
      N=1                                                               DHPCG048
      ISW=2                                                             DHPCG049
      GO TO 100                                                         DHPCG050
C                                                                       DHPCG051
   13 X=X+H                                                             DHPCG052
      CALL FCT(X,Y,DERY)                                                DHPCG053
      IF(MM.GT.1) RETURN                                                DHPCG054
      N=2                                                               DHPCG055
      DO 14 I=1,NDIM                                                    DHPCG056
      AUX(2,I)=Y(I)                                                     DHPCG057
   14 AUX(9,I)=DERY(I)                                                  DHPCG058
      ISW=3                                                             DHPCG059
      GO TO 100                                                         DHPCG060
C                                                                       DHPCG061
C     COMPUTATION OF TEST VALUE DELT                                    DHPCG062
   15 DELT=0.                                                           DHPCG063
      DO 16 I=1,NDIM                                                    DHPCG064
   16 DELT=DELT+AUX(15,I)*ABS(Y(I)-AUX(4,I))                            DHPCG065
      DELT=.06666667*DELT                                               DHPCG066
      IF(DELT-PRMT(4))19,19,17                                          DHPCG067
   17 IF(IHLF-10)11,18,18                                               DHPCG068
C                                                                       DHPCG069
C     NO SATISFACTORY ACCURACY AFIER 10 BISECTIONS. ERROR MESSAGE.      DHPCG070
   18 IHLF=11                                                           DHPCG071
      X=X+H                                                             DHPCG072
      GO TO 4                                                           DHPCG073
C                                                                       DHPCG074
C     THERE IS SATISFACTORY ACCURACY AFIER LFSS THAN 11 BISECTIONS.     DHPCG075
   19 X=X+H                                                             DHPCG076
      CALL FCT(X,Y,DERY)                                                DHPCG077
      IF(MM.GT.1) RETURN                                                DHPCG078
      DO 20 I=1,NDIM                                                    DHPCG079
      AUX(3,I)=Y(I)                                                     DHPCG080
   20 AUX(10,I)=DERY(I)                                                 DHPCG081
      N=3                                                               DHPCG082
      ISW=4                                                             DHPCG083
      GO TO 100                                                         DHPCG084
C                                                                       DHPCG085
   21 N=1                                                               DHPCG086
      X=X+H                                                             DHPCG087
      CALL FCT(X,Y,DERY)                                                DHPCG088
      IF(MM.GT.1) RETURN                                                DHPCG089
      X=PRMT(1)                                                         DHPCG090
      DO 22 I=1,NDIM                                                    DHPCG091
      AUX(11,I)=DERY(I)                                                 DHPCG092
   22 Y(I)=AUX(1,I)+H*(.375*AUX(8,I)+.7916667*AUX(9,I)-.2083333*AUX(10DHPCG093
     1,I)+.04166667*DERY(I))                                            DHPCG094
   23 X=X+H                                                             DHPCG095
      N=N+1                                                             DHPCG096
      CALL FCT(X,Y,DERY)                                                DHPCG097
      IF(MM.GT.1) RETURN                                                DHPCG098
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                DHPCG099
      IF(PRMT(5))6,24,6                                                 DHPCG100
   24 IF(N-4)25,200,200                                                 DHPCG101
   25 DO 26 I=1,NDIM                                                    DHPCG102
      AUX(N,I)=Y(I)                                                     DHPCG103
   26 AUX(N+7,I)=DERY(I)                                                DHPCG104
      IF(N-3)27,29,200                                                  DHPCG105
C                                                                       DHPCG106
   27 DO 28 I=1,NDIM                                                    DHPCG107
      DELT=AUX(9,I)+AUX(9,I)                                            DHPCG108
      DELT=DELT+DELT                                                    DHPCG109
   28 Y(I)=AUX(1,I)+.33333333*H*(AUX(8,I)+DELT+AUX(10,I))               DHPCG110
      GO TO 23                                                          DHPCG111
C                                                                       DHPCG112
   29 DO 30 I=1,NDIM                                                    DHPCG113
      DELT=AUX(9,I)+AUX(10,I)                                           DHPCG114
      DELT=DELT+DELT+DELT                                               DHPCG115
   30 Y(I)=AUX(1,I)+.375*H*(AUX(8,I)+DELT+AUX(11,I))                    DHPCG116
      GO TO 23                                                          DHPCG117
C                                                                       DHPCG118
C                                                                       DHPCG119
C     ******************************************************************DHPCG120
C     THE FOLLOWING PART OF SUBROUTINE DHPCG COMPUTES BY MEANS OF       DHPCG121
C     RUNGE-KUTTA METHOD STARTING VALUES FOR THE NOT SELF-STARTING      DHPCG122
C     PREDICTOR-CORRECTOR METHOD.                                       DHPCG123
  100 DO 101 I=1,NDIM                                                   DHPCG124
      Z=H*AUX(N+7,I)                                                    DHPCG125
      AUX(5,I)=Z                                                        DHPCG126
  101 Y(I)=AUX(N,I)+.4*Z                                                DHPCG127
C     Z IS AN AUXILIARY STORAGE LOCATION                                DHPCG128
C                                                                       DHPCG129
      Z=X+.4*H                                                          DHPCG130
      CALL FCT(Z,Y,DERY)                                                DHPCG131
      IF(MM.GT.1) RETURN                                                DHPCG132
      DO 102 I=1,NDIM                                                   DHPCG133
      Z=H*DERY(I)                                                       DHPCG134
      AUX(6,I)=Z                                                        DHPCG135
  102 Y(I)=AUX(N,I)+.29697761*AUX(5,I)+.15875964*Z                      DHPCG136
C                                                                       DHPCG137
      Z=X+.45573725*H                                                   DHPCG138
      CALL FCT(Z,Y,DERY)                                                DHPCG139
      IF(MM.GT.1) RETURN                                                DHPCG140
      DO 103 I=1,NDIM                                                   DHPCG141
      Z=H*DERY(I)                                                       DHPCG142
      AUX(7,I)=Z                                                        DHPCG143
  103 Y(I)=AUX(N,I)+.21810039*AUX(5,I)-3.05096515*AUX(6,I)+3.83286476*Z DHPCG144
C                                                                       DHPCG145
      Z=X+H                                                             DHPCG146
      CALL FCT(Z,Y,DERY)                                                DHPCG147
      IF(MM.GT.1) RETURN                                                DHPCG148
      DO 104 I=1,NDIM                                                   DHPCG149
  104 Y(I)=AUX(N,I)+.17476028*AUX(5,I)-.55148066*AUX(6,I)+1.2055356*AUX(DHPCG150
     17,I)+.17118478*H*DERY(I)                                          DHPCG151
      GO TO(9,13,15,21),ISW                                             DHPCG152
C     ******************************************************************DHPCG153
C                                                                       DHPCG154
C                                                                       DHPCG155
C     POSSIBLE BREAK-POINT FOR LINKAGE                                  DHPCG156
C                                                                       DHPCG157
C                                                                       DHPCG158
C     STARTING VALUES ARE COMPUTED.                                     DHPCG159
C     NOW START HAMMINGS MODIFIED PREDICTOR-CORRECTOR METHOD.           DHPCG160
  200 ISTEP=3                                                           DHPCG161
  201 IF(N-8)204,202,204                                                DHPCG162
C                                                                       DHPCG163
C     N=8 CAUSES THE ROWS OF AUX TO CHANGE THEIR STORAGE LOCATIONS      DHPCG164
  202 DO 203 N=2,7                                                      DHPCG165
      DO 203 I=1,NDIM                                                   DHPCG166
      AUX(N-1,I)=AUX(N,I)                                               DHPCG167
```

```
  203 AUX(N+6,I)=AUX(N+7,I)                                             DHPC6168
      N=7                                                               DHPC6169
C                                                                       DHPC6170
C     N LESS THAN 8 CAUSES N+1 TO GET N                                 DHPC6171
  204 N=N+1                                                             DHPC6172
C                                                                       DHPC6173
C     COMPUTATION OF NEXT VECTOR Y                                      DHPC6174
      DO 205 I=1,NDIM                                                   DHPC6175
      AUX(N-1,I)=Y(I)                                                   DHPC6176
  205 AUX(N+6,I)=DERY(I)                                                DHPC6177
      X=X+H                                                             DHPC6178
  206 ISTEP=ISTEP+1                                                     DHPC6179
      DO 207 I=1,NDIM                                                   DHPC6180
      DELT=AUX(N-4,I)+1.333333333*H*(AUX(N+6,I)+AUX(N+6,I)-AUX(N+5,I)+AUXDHPC6181
     1(N+4,I)+AUX(N+4,I))                                               DHPC6182
      Y(I)=DELT-.9256198³*AUX(16,I)                                     DHPC6183
  207 AUX(16,I)=DELT                                                    DHPC6184
C     PREDICTOR IS NOW GENERATED IN ROW 16 OF AUX, MODIFIED PREDICTOR   DHPC6185
C     IS GENERATED IN Y. DELT MEANS AN AUXILIARY STORAGE.               DHPC6186
C                                                                       DHPC6187
      CALL FCT(X,Y,DERY)                                                DHPC6188
      IF (MM.GT.1) RETURN                                               DHPC6189
C     DERIVATIVE OF MODIFIED PREDICTOR IS GENERATED IN DERY             DHPC6190
C                                                                       DHPC6191
      DO 208 I=1,NDIM                                                   DHPC6192
      DELT=.125U0*(9.U0*AUX(N-1,I)-AUX(N-3,I)+3.00*H*(DERY(I)+AUX(N+6,I)DHPC6193
     1+AUX(N+6,I)-AUX(N+5,I)))                                          DHPC6194
      AUX(16,I)=AUX(16,I)-DELT                                          DHPC6195
  208 Y(I)=DELT+.07438U165*AUX(16,I)                                    DHPC6196
C                                                                       DHPC6197
C     TEST WHETHER H MUST BE HALVED OR DOUBLED                          DHPC6198
      DELT=0.                                                           DHPC6199
      DO 209 I=1,NDIM                                                   DHPC6200
  209 DELT=DELT+AUX(15,I)*ABS(AUX(16,I))                                DHPC6201
      IF(DELT-PRMT(4))210,222,222                                       DHPC6202
C                                                                       DHPC6203
C     H MUST NOT BE HALVED. THAT MEANS Y(I) ARE GOOD.                   DHPC6204
  210 CALL FCT(X,Y,DERY)                                                DHPC6205
      IF(MM.GT.1) RETURN                                                DHPC6206
      CALL OUTP(X,Y,DERY,IHLF,NDIM,PRMT)                                DHPC6207
      IF(PRMT(5))212,211,212                                            DHPC6208
  211 IF(IHLF-11)213,212,212                                            DHPC6209
  212 RETURN                                                            DHPC6210
  213 IF(H*(X-PRMT(2)))214,212,212                                      DHPC6211
  214 IF(ABS(X-PRMT(2))-.1*ABS(H))212,215,215                           DHPC6212
  215 IF(DELT-.02*PRMT(4))216,216,201                                   DHPC6213
C                                                                       DHPC6214
C                                                                       DHPC6215
C     H COULD BE DOUBLED IF ALL NECESSARY PRECEDING VALUES ARE          DHPC6216
C     AVAILABLE                                                         DHPC6217
  216 IF(IHLF)201,201,217                                               DHPC6218
  217 IF(N-7)201,218,218                                                DHPC6219
  218 IF(ISTEP-4)201,219,219                                            DHPC6220
  219 IMOD=ISTEP/2                                                      DHPC6221
      IF(ISTEP-IMOD-IMOD)201,220,201                                    DHPC6222
  220 H=H+H                                                             DHPC6223
      IHLF=IHLF-1                                                       DHPC6224
      ISTEP=0                                                           DHPC6225
      DO 221 I=1,NDIM                                                   DHPC6226
      AUX(N-1,I)=AUX(N-2,I)                                             DHPC6227
      AUX(N-2,I)=AUX(N-4,I)                                             DHPC6228
      AUX(N-3,I)=AUX(N-6,I)                                             DHPC6229
      AUX(N+6,I)=AUX(N+5,I)                                             DHPC6230
      AUX(N+5,I)=AUX(N+3,I)                                             DHPC6231
      AUX(N+4,I)=AUX(N+1,I)                                             DHPC6232
      DELT=AUX(N+6,I)+AUX(N+5,I)                                        DHPC6233
      DELT=DELT+DELT+DELT                                               DHPC6234
  221 AUX(16,I)=8.962962962*(Y(I)-AUX(N-3,I))-3.361111111*H*(DERY(I)+DELT+DHPC6235
     1AUX(N+4,I))                                                       DHPC6236
      GO TO 201                                                         DHPC6237
C                                                                       DHPC6238
C                                                                       DHPC6239
C     H MUST BE HALVED                                                  DHPC6240
  222 IHLF=IHLF+1                                                       DHPC6241
      IF(IHLF-10)223,223,210                                            DHPC6242
  223 H=.5*H                                                            DHPC6243
      ISTEP=0                                                           DHPC6244
      DO 224 I=1,NDIM                                                   DHPC6245
      Y(I)=.390625E-2*(8.E1*AUX(N-1,I)+135.00*AUX(N-2,I)+4.E1*AUX(N-3,I)DHPC6246
     1+AUX(N-4,I))-.1171875*(AUX(N+6,I)-6.*AUX(N+5,I)-AUX(N+4,I))*H     DHPC6247
      AUX(N-4,I)=.390625E-2*(12.00*AUX(N-1,I)+135.00*AUX(N-2,I)+        DHPC6248
     1108.00*AUX(N-3,I)+AUX(N-4,I))-.0234375*(AUX(N+6,I)+               DHPC6249
     218.00*AUX(N+5,I)-9.00*AUX(N+4,I))*H                               DHPC6250
      AUX(N-3,I)=AUX(N-2,I)                                             DHPC6251
  224 AUX(N+4,I)=AUX(N+5,I)                                             DHPC6252
      X=X-H                                                             DHPC6253
      DELT=X-(H+H)                                                      DHPC6254
      CALL FCT(DELT,Y,DERY)                                             DHPC6255
      IF(MM.GT.1) RETURN                                                DHPC6256
      DO 225 I=1,NDIM                                                   DHPC6257
      AUX(N-2,I)=Y(I)                                                   DHPC6258
      AUX(N+5,I)=DERY(I)                                                DHPC6259
  225 Y(I)=AUX(N-4,I)                                                   DHPC6260
      DELT=DELT-(H+H)                                                   DHPC6261
      CALL FCT(DELT,Y,DERY)                                             DHPC6262
      IF(MM.GT.1) RETURN                                                DHPC6263
      DO 226 I=1,NDIM                                                   DHPC6264
      DELT=AUX(N+5,I)+AUX(N+4,I)                                        DHPC6265
      DELT=DELT+DELT+DELT                                               DHPC6266
      AUX(16,I)=8.962962962*(AUX(N-1,I)-Y(I))                           DHPC6267
     1-3.361111111*H*(AUX(N+6,I)+DELT+DERY(I))                          DHPC6268
  226 AUX(N+3,I)=DERY(I)                                                DHPC6269
      GO TO 206                                                         DHPC6270
      END                                                               DHPC6271
```

```
      SUBROUTINE PHITKS(X,Y,Z,VTX,VTY,VTZ)                              PHTKS001
      IMPLICIT COMPLEX*8(C)                                             PHTKS002
      DIMENSION THETA(5),THETA1(5),THETA2(5),RF(7),YN1(5),ZN1(5),SCS(5) PHTKS003
      EXTERNAL FUNX1,FUNX2,FUNX3,FUNX4,FUNX5,FUNX6,FUNX7,FUNY1,FUNY2,   PHTKS004
     2FUNY3,FUNY4,FUNZ1,DZWDX,FUNX8,FUNX9,FUNX10,FUNY5                  PHTKS005
      EXTERNAL FUNX11,FUNX21,FUNX22,FUNX23,FUNX25,FUNX26,FUNX27,FUNX28, PHTKS006
     1FUNX13,FUNX14,FUNX15,FUNX16,FUNX17,FUNX61,FUNX71,FUNY11,FUNY12,   PHTKS007
     2FUNY13,FUNY14,FUNZ11,FUNX24,FUNX16,FUNX19,FUNX10,FUNY15           PHTKS008
      COMMON /BLK2/ DX11,DX12,DUU1,DUU2,DCD,ALPHA,THETA,THETA1,THETA2,  PHTKS009
     1RF,YN1,ZN1,SCS,DWRTE,DMRM,MMM,M11,NTHETA,IC,ICOPY                 PHTKS010
      COMMON /BLK3/ DN,UN1,UN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  PHTKS011
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    PHTKS012
      COMMON /BLK4/ DL,DL12,DL1,DL2,DL3,DL4,M,MUPT,MELLPT               PHTKS013
      COMMON /BLK5/ XRLE1,XRTE1,HH,H3,NNN                               PHTKS014
      COMMON /BLK6/ PI,PI2,TOL,NII                                      PHTKS015
      COMMON /BLK11/ RF1,II                                             PHTKS016
      COMMON /BLK13/ YY,Z2,R2,Z2,Y22,ZY8,R2Y,A2MB24                     PHTKS017
      COMMON /BLK14/ DZWXY,D2ZWXY,DZR,D2ZR,D2ZMXY,D2ZMXR                PHTKS018
      COMMON /BLK16/ AZWXY,A2ZWXY,DZA,D2ZA,A2ZMXY,D2ZMXA                PHTKS019
      IF(MELLPT.EQ.1) GO TO 300                                         PHTKS020
      IF(X.LT.XRLE1) GO TO 600                                          PHTKS021
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 600                             PHTKS022
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 600                             PHTKS023
      IF(MTE.EQ.1.AND.X.GT.XRTE1) GO TO 500                             PHTKS024
      XL=RB(X)                                                          PHTKS025
      XU=SPANLE(X)                                                      PHTKS026
      DSDX=DSPLE(X)                                                     PHTKS027
      DRDX=DRBDX(X)                                                     PHTKS028
      D2RDX=D2RBDX(X)                                                   PHTKS029
      DZR=DZWDX(X,XL)                                                   PHTKS030
      D2ZR=D2ZWDX(X,XL)                                                 PHTKS031
      Z2=Z*Z                                                            PHTKS032
      Y2=Y*Y                                                            PHTKS033
      Y22=Y2*Y2                                                         PHTKS034
      ZY2=Z2+Y2                                                         PHTKS035
```

```
      ZY4=ZY2*ZY2                                                       PHTKS036
      ZY8=ZY4*ZY4                                                       PHTKS037
      R2=XL*XL                                                          PHTKS038
      R2S=R2/XU                                                         PHTKS039
      YY=Y                                                              PHTKS040
      ZZ=Z                                                              PHTKS041
      R22=R2                                                            PHTKS042
      SR2=XU+R2S                                                        PHTKS043
      RS=XL/XU                                                          PHTKS044
      A1=Z2+(Y-XU)*(Y-XU)                                               PHTKS045
      A2=Z2+(Y+XU)*(Y+XU)                                               PHTKS046
      A3=Z2+(Y-R2S)*(Y-R2S)                                             PHTKS047
      A4=Z2+(Y+R2S)*(Y+R2S)                                             PHTKS048
      A5=Z2+(Y-XL)*(Y-XL)                                               PHTKS049
      A6=Z2+(Y+XL)*(Y+XL)                                               PHTKS050
      VTXS=DZWDX(X,XU)*DSDX*ALOG(A1*A2*A3*A4/ZY8)                       PHTKS051
      VTXB=PI2*S2CHPI(X)+2.*((DZR+DZWDY(X,XL)*DRDX)*DRDX+ZW(X,XL)*D2RBDXPHTKS052
     1(X))                                                              PHTKS053
      VTXB=VTXB*ALOG(ZY2)                                               PHTKS054
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,DZWDX,X,VTYZ)                       PHTKS055
   31 VTYZ=(-4.*VTYZ+PI2*S1EBPI(X)+2.*ZW(X,XL)*DRDX)/ZY2                PHTKS056
      IF(ABS(Z).LE.1.E-4.AND.ABS(Y).LT.XU) GO TO 1                      PHTKS057
      VTXR=-2.*DZR*DRDX*ALOG(A5*A6/ZY4)                                 PHTKS058
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX1,X,VTX1)                       PHTKS059
   20 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX2,X,VTX2)                       PHTKS060
   21 VTX2=4.*XL*DRDX*VTX2                                              PHTKS061
      VTX=(VTX1+VTX2+VTXS+VTXR+VTXB)/PI2                                PHTKS062
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNY1,X,VTY1)                       PHTKS063
      VTYYZ=Y*VTYZ                                                      PHTKS064
      VTY=(VTY1+VTYYZ)/PI                                               PHTKS065
      IF(ABS(Z).LE.1.E-4) GO TO 400                                     PHTKS066
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNZ1,X,VTZ1)                       PHTKS067
   23 VTZ1=Z*VTZ1                                                       PHTKS068
      VTZYZ=Z*VTYZ                                                      PHTKS069
      VTZ=(VTZ1+VTZYZ)/PI                                               PHTKS070
      RETURN                                                            PHTKS071
  400 VTZ=0.                                                            PHTKS072
      RETURN                                                            PHTKS073
    1 DZWXY=DZWDX(X,Y)                                                  PHTKS074
      D2ZWXY=D2ZWDX(X,Y)                                                PHTKS075
      D2ZMXY=D2ZDXY(X,Y)                                                PHTKS076
      D2ZMXR=D2ZDXY(X,XL)                                               PHTKS077
      R=XL                                                              PHTKS078
      XUL=XU-XL                                                         PHTKS079
      R2Y=R2/Y                                                          PHTKS080
      SRY=XU-R2Y                                                        PHTKS081
      RRY=R-R2Y                                                         PHTKS082
      YR=Y-XL                                                           PHTKS083
      SY=XU-Y                                                           PHTKS084
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX3,X,VTX3)                       PHTKS085
   24 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX4,X,VTX4)                       PHTKS086
   25 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX5,X,VTX5)                       PHTKS087
   26 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX6,X,VTX6)                       PHTKS088
      VTX6=4.*XL*DRDX*VTX6/Y                                            PHTKS089
   27 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX7,X,VTX7)                       PHTKS090
      VTX7=-4.*XL*DRDX*VTX7/Y                                           PHTKS091
   28 IF(ABS(YR).LE.1.E-4) GO TO 50                                     PHTKS092
      VTX8=2.*D2ZWXY*(SY*(ALOG(SY/Y)-1.)+YR*(ALOG(YR/Y)-1.))            PHTKS093
      VTX9=2.*D2ZR*(XU*ALOG(SRY/XU)-XL*ALOG(RRY/XL)-R2Y*ALOG(SRY/RRY))  PHTKS094
      VTX10=-4.*XL*DRDX*DZR*ALOG(SRY/RRY)/Y                             PHTKS095
      VTXR=-2.*DZR*DRDX*ALOG(A5*A6/ZY4)                                 PHTKS096
   51 VTX=(VTX3+VTX4+VTX5+VTX6+VTX7+VTX8+VTX9+VTX10+VTXS+VTXR+VTXB)/PI2 PHTKS097
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNY2,X,VTY2)                       PHTKS098
   29 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNY3,X,VTY3)                       PHTKS099
      VTY3=-VTY3                                                        PHTKS100
   30 CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNY4,X,VTY4)                       PHTKS101
      VTY4=VTY4/Y                                                       PHTKS102
   32 IF(ABS(YR).LE.1.E-4) GO TO 52                                     PHTKS103
      VTY5=-DZWXY*ALOG(SY/YR)                                           PHTKS104
      VTY6=DZR*(XUL+R2Y*ALOG(SRY/RRY))/Y                                PHTKS105
   54 VTYYZ=Y*VTYZ                                                      PHTKS106
      VTY=(VTY2+VTY3+VTY4+VTY5+VTY6+VTYYZ)/PI                           PHTKS107
      VTZ=DZWXY                                                         PHTKS108
      RETURN                                                            PHTKS109
   50 VTX8=2.*D2ZR*XUL*(ALOG(XUL/R)-1.)                                 PHTKS110
      VTX9=2.*D2ZR*(XU*ALOG(XUL/XU)-R*ALOG(XUL/R))                      PHTKS111
      VTX10=0.                                                          PHTKS112
      VTXR=-4.*DZR*DRDX*ALOG(2.*XUL/R)                                  PHTKS113
      GO TO 51                                                          PHTKS114
   52 VTY5=0.                                                           PHTKS115
      VTY6=DZR*XUL/R                                                    PHTKS116
      GO TO 54                                                          PHTKS117
  500 STE=SPANTE(X)                                                     PHTKS118
      SLE=SPANLE(X)                                                     PHTKS119
      DSLEDX=DSPLE(X)                                                   PHTKS120
      DSTEDX=DSPTE(X)                                                   PHTKS121
      RBB=RB(X)                                                         PHTKS122
      DRDX=DRBDX(X)                                                     PHTKS123
      DZLE=DZWDX(X,SLE)                                                 PHTKS124
      DZTE=DZWDX(X,STE)                                                 PHTKS125
      YY=Y                                                              PHTKS126
      ZZ=Z                                                              PHTKS127
      R2=RBB**2                                                         PHTKS128
      Z2=Z**2                                                           PHTKS129
      Y2=Y**2                                                           PHTKS130
      Y22=Y2*Y2                                                         PHTKS131
      ZY2=Z2+Y2                                                         PHTKS132
      ZY4=ZY2*ZY2                                                       PHTKS133
      ZY8=ZY4*ZY4                                                       PHTKS134
      R2SLE=R2/SLE                                                      PHTKS135
      R2STE=R2/STE                                                      PHTKS136
      A1=Z2+(Y-SLE)**2                                                  PHTKS137
      A2=Z2+(Y+SLE)**2                                                  PHTKS138
      A3=Z2+(Y-R2SLE)**2                                                PHTKS139
      A4=Z2+(Y+R2SLE)**2                                                PHTKS140
      A5=Z2+(Y-STE)**2                                                  PHTKS141
      A6=Z2+(Y+STE)**2                                                  PHTKS142
      A7=Z2+(Y-R2STE)**2                                                PHTKS143
      A8=Z2+(Y+R2STE)**2                                                PHTKS144
      VTXSLE=DZLE*DSLEDX*ALOG(A1*A2*A3*A4/ZY8)                          PHTKS145
      VTXSTE=-DZTE*DSTEDX*ALOG(A5*A6*A7*A8/ZY8)                         PHTKS146
      VTXB=PI2*S2CBPI(X)*ALOG(ZY2)                                      PHTKS147
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,DZWDX,X,VTYZ)                     PHTKS148
      VTYZ=(-4.*VTYZ+PI2*S1EBPI(X))/ZY2                                 PHTKS149
      IF(ABS(Z).LE.1.E-4) GO TO 99                                      PHTKS150
      GO TO 98                                                          PHTKS151
   99 IF(ABS(Y).LT.SLE.AND.ABS(Y).GT.STE) GO TO 100                     PHTKS152
   98 CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNX1,X,VTX1)                     PHTKS153
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNX2,X,VTX2)                     PHTKS154
      VTX2=4.*RBB*DRDX*VTX2                                             PHTKS155
      VTX=(VTX1+VTX2+VTXSLE+VTXSTE+VTXB)/PI2                            PHTKS156
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNY1,X,VTY1)                     PHTKS157
      VTYYZ=Y*VTYZ                                                      PHTKS158
      VTY=(VTY1+VTYYZ)/PI                                               PHTKS159
      IF(ABS(Z).LE.1.E-4) GO TO 97                                      PHTKS160
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNZ1,X,VTZ1)                     PHTKS161
      VTZ1=Z*VTZ1                                                       PHTKS162
      VTZYZ=Z*VTYZ                                                      PHTKS163
      VTZ=(VTZ1+VTZYZ)/PI                                               PHTKS164
      RETURN                                                            PHTKS165
   97 VTZ=0.                                                            PHTKS166
      RETURN                                                            PHTKS167
  100 DZWXY=DZWDX(X,Y)                                                  PHTKS168
      D2ZWXY=D2ZWDX(X,Y)                                                PHTKS169
      D2ZMXY=D2ZDXY(X,Y)                                                PHTKS170
      R2Y=R2/Y                                                          PHTKS171
      SLEY=SLE-Y                                                        PHTKS172
      YSTE=Y-STE                                                        PHTKS173
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNX8,X,VTX11)                    PHTKS174
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNX4,X,VTX4)                     PHTKS175
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNX9,X,VTX12)                    PHTKS176
      VTX12=4.*RBB*DRDX*VTX12/Y                                         PHTKS177
      VTX13=2.*D2ZWXY*(SLEY*(ALOG(SLEY/Y)-1.)+YSTE*(ALOG(YSTE/Y)-1.))   PHTKS178
      VTX=(VTX4+VTX11+VTX12+VTX13+VTXSLE+VTXSTE+VTXB)/PI2               PHTKS179
```

```
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNY3,X,VTY3)
      VTY3=-VTY3
      CALL SIMP1(STE,SLE,NIT,NIT1,TOL,FUNY5,X,VTY5)
      VTY6=-DZWXY*ALOG(SLEY/YSTE)
      VTYYZ=Y*VTYZ
      VTY=(VTY3+VTY5+VTY6+VTYYZ)/PI
      VTZ=DZWXY
      RETURN
  600 VTX=2.*S2EHPI(X)*ALOG(RF1)
      VTFAC=2.*S1EHPI(X)/RF1
      VTY=VTFAC*YN1(I1)
      VTZ=VTFAC*ZN1(I1)
      RETURN
  500 DX=X
      IF(X.LT.XRLE1) GO TO 6000
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 6000
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 6000
      A=AB(X)
      A2MB2=DL1*A*A
      A2MB24=A2MB2/4.
      IF(ABS(Z).LE.1.E-4) Z=0.
      CSIG=CMPLX(Y,Z)
      CA=CSQRT(CSIG*CSIG-A2MB2)
      C1=(CSIG+CA)/2.
      Y1=REAL(C1)
      Z1=AIMAG(C1)
      YY=Y1
      ZZ=Z1
      Z2=Z1*Z1
      Y2=Y1*Y1
      Y22=Y2*Y2
      ZY2=Z2+Y2
      AZY2=ZY2
      ZY4=ZY2*ZY2
      ZY8=ZY4*ZY4
      RR=DL4*A
      XL=RR
      R2=XL*XL
      CQ12=1./(C1*C1-A2MB24)
      CQ34=C1*CQ12
      CQ56=C1*CQ34
      Q1=REAL(CQ12)
      Q2=AIMAG(CQ12)
      Q3=REAL(CQ34)
      Q4=AIMAG(CQ34)
      Q5=REAL(CQ56)
      Q6=AIMAG(CQ56)
      DADX=DABDX(X)
      AFACQ5=DL1*A*DADX*DL2*Q5
      AFACQ6=DL1*A*DADX*DL2*Q6*Z1
      AFACQ3=DL1*A*DADX*A2MB24*Q3
      AFACQ4=DL1*A*DADX*A2MB24*Q4*Z1
      IF(MTE.EQ.1.AND.X.GT.XRTE1) GO TO 1500
      S=SPANLE(X)
      SCM212=SQRT(S*S-A2MB2)
      DSDX=DSPLE(X)
      D2ADX=D2ABDX(X)
      DZA=DZWDX(X,A)
      D2ZA=D2ZWDX(X,A)
      SS=(S+SCM212)/2.
      XU=SS
      R2S=R2/XU
      R22=R2
      AH22=R22
      SR2=XU+R2S
      RS=XL/XU
      A1=Z2+(Y1-XU)**2
      A2=Z2+(Y1+XU)**2
      A3=Z2+(Y1-R2S)**2
      A4=Z2+(Y1+R2S)**2
      A5=Z2+(Y1-XL)**2
      A6=Z2+(Y1+XL)**2
      VTXS=DZWDX(X,S)*DSDX*ALOG(A1*A2*A3*A4/ZY8)
      VTXB=PI2*S2EBPI(DX)+2.*((DZA+DZWDY(X,A)*DADX)*DADX+ZW(X,A)*D2ADX)
      VTXB=VTXB*ALOG(ZY2)
      CALL SIMP1(A,S,NIT,NIT1,TOL,DZWDX,X,VTYZ1)
      VTYZ=(-4.*VTYZ1+PI2*S1EBPI(DX)+2.*ZW(X,A)*DADX)/ZY2
      VTX11=-DL1*A*DADX*Q1*(2.*VTYZ1+VTYZ*ZY2)
      IF(ABS(Z1).LE.1.E-4.AND.ABS(Y1).LT.XU) GO TO 11
      VTXA=-2.*DZA*DADX*ALOG(A5*A6/ZY4)
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX11,X,VTX1)
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX21,X,VTX21)
      VTX21=-AFACQ5*VTX21
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX22,X,VTX22)
      VTX22=-AFACQ6*VTX22
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX23,X,VTX23)
      VTX23=AFACQ5*VTX23
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX24,X,VTX24)
      VTX24=AFACQ6*VTX24
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX25,X,VTX25)
      VTX25=AFACQ3*VTX25
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX26,X,VTX26)
      VTX26=AFACQ4*VTX26
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX27,X,VTX27)
      VTX27=AFACQ3*VTX27
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX28,X,VTX28)
      VTX28=AFACQ4*VTX28
      VTX2=VTX11+VTX21+VTX22+VTX23+VTX24+VTX25+VTX26+VTX27+VTX28
      VTX=(VTX1+VTX2+VTXS+VTXA+VTXB)/PI2
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNY11,X,VTY1)
      VTYY1=Q5*VTY1
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNZ11,X,VTZ1)
      VTZ1=Z1*VTZ1
      VTYYZ=(Q5*Y1+Q6*Z1)*VTYZ
      IF(ABS(Z1).LE.1.E-4) GO TO 1400
      VTYZ1=Q6*VTZ1
      VTY=(VTYY1+VTYZ1+VTYYZ)/PI
      VTZY1=-Q6*VTY1
      VTZZ1=Q5*VTZ1
      VTZYZ=(Q5*Z1-Q6*Y1)*VTYZ
      VTZ=(VTZY1+VTZZ1+VTZYZ)/PI
      RETURN
 1400 VTZ=0.
      VTY=(VTYY1+VTYYZ)/PI
      RETURN
   11 DZWXY=DZWDX(X,Y)
      D2ZWXY=D2ZWDX(X,Y)
      D2ZMXY=D2ZDXY(X,Y)
      D2ZMXA=D2ZDXY(X,A)
      AZWXY=DZWXY
      A2ZWXY=D2ZWXY
      A2ZMXY=D2ZMXY
      R=XL
      XUL=XU-XL
      R2Y=R2/Y1
      SRY=XU-R2Y
      RRY=R-R2Y
      YR=Y1-XL
      YPR=Y1+XL
      SY=XU-Y1
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX13,X,VTX3)
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX14,X,VTX4)
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX15,X,VTX5)
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX16,X,VTX6)
      VTX6=AFACQ5*VTX6/Y1
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX17,X,VTX7)
      VTX7=-AFACQ5*VTX7/Y1
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX71,X,VTX71)
      VTX71=AFACQ3*VTX71/Y1
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,FUNX61,X,VTX61)
      VTX61=AFACQ3*VTX61/Y1
      IF(ABS(YR).LE.1.E-4) GO TO 150
      VTX8=2.*D2ZWXY*(SY*(ALOG(SY/Y1)-1.)+YR*(ALOG(YR/Y1)-1.)-A2MB24*
     1((SY/XU)*ALOG(SY/Y1)+(YR/R)*ALOG(YR/Y1)-ALOG(XU/R))/Y1)
```

```
      VTX9=2.*DZZA*(XU*ALOG(SRY/XU)-XL*ALOG(RRY/YL)-R2Y*ALOG(SRY/RRY)
     1-A2MB24*(((SRY/R2Y)*ALOG(SRY/XU)+1.)/XU-((RRY/R2Y)*ALOG(RRY/R)+1.)
     2/R))
      VTXA10=DZA*DADX*(-4.*ALOG(YR*TPR/Y2)-4.*A*Q3*ALOG(SRY/RRY)/DL-
     1DL3*A*Q3*ALOG(XU/R))
  151 VTX=(VTX3+VTX4+VTX5+VTX6+VTX61+VTX7+VTX71+VTX8+VTX9+VTX11+VTXA10+
     1VTX5+VTX8)/PI2
      CALL SIMPI(XL,XU,NIT,NIT1,TOL,FUNY12,X,VTY2)
      VTYY2=Q5*VTY2
      CALL SIMPI(XL,XU,NIT,NIT1,TOL,FUNY13,X,VTY3)
      VTY3=-Q5*VTY3
      CALL SIMPI(XL,XU,NIT,NIT1,TOL,FUNY14,X,VTY4)
      VTY4=Q5*VTY4/Y1
      IF(ABS(YR).LE.1.E-4) GO TO 152
      VTY5=-Q5*DZWXY*(ALOG(SY/YR)+A2MB24*((ALOG(XU/R)-ALOG(SY/YR))/Y2+
     1(XUL/(XU*R))/Y1))
      VTY6=Q5*DZA*(XUL+R2Y*ALOG(SRY/RRY)+A2MB24*(ALOG(XU/R)-ALOG(SRY/RRY
     1))/R2Y)/Y1
  154 VTYYZ=Q5*Y1*VTYZ
      VTY=(VTYY2+VTY3+VTY4+VTY5+VTY6+VTYYZ)/PI
      VTZ=DZWXY
      RETURN
  150 VTX8=2.*DZZA*(XUL*(ALOG(XUL/R)-1.)-A2MB24*((XUL/XU)*ALOG(XUL/R)-
     1ALOG(XU/R))/R)
      VTX9=2.*DZZA*(XU*ALOG(XUL/XU)-R*ALOG(XUL/R)-A2MB24*(((XUL/R)*ALOG(
     1XUL/XU)+1.)/XU-1./R))
      VTXA10=DZA*DADX*(-4.*ALOG(2.*XUL/R)-DL*DL3*ALOG(XU/R))
      GO TO 151
  152 VTY5=0.
      VTY6=Q5*DZA*(XUL/R-A2MB24*XUL/(R2*XU))
      GO TO 154
 1500 STE=SPANTE(X)
      SLE=SPANLE(X)
      DSLEDX=DSPLE(X)
      DSTEDX=DSPTE(X)
      DZLE=DZWDX(X,SLE)
      DZTE=DZWDX(X,STE)
      STE1=(STE+SQRT(STE*STE-A2MB2))/2.
      SLE1=(SLE+SQRT(SLE*SLE-A2MB2))/2.
      R2SLE1=R2/SLE1
      R2STE1=R2/STE1
      A1=Z2+(Y1-SLE1)**2
      A2=Z2+(Y1+SLE1)**2
      A3=Z2+(Y1-R2SLE1)**2
      A4=Z2+(Y1+R2SLE1)**2
      A5=Z2+(Y1-STE1)**2
      A6=Z2+(Y1+STE1)**2
      A7=Z2+(Y1-R2STE1)**2
      A8=Z2+(Y1+R2STE1)**2
      VTXSLE=DZLE*DSLEDX*ALOG(A1*A2*A3*A4/ZY8)
      VTXSTE=-DZTE*DSTEDX*ALOG(A5*A6*A7*A8/ZY8)
      VTX6=PI2*S2EBPI(DX)*ALOG(ZY2)
      CALL SIMPI(STE,SLE,NIT,NIT1,TOL,DZWDX,X,VTYZ1)
      VTYZ=(-4.*VTYZ1+PI2*SIEBPI(DX))/ZY2
      VTX11=-DL1*A*DADX*Q1*(2.*VTYZ1+VTY7*ZY2)
      IF(ABS(Z1).LE.1.E-4) GO TO 1199
      GO TO 1198
 1199 IF(ABS(Y).LT.SLE.AND.ABS(Y).GT.STE) GO TO 1100
 1198 CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX11,X,VTX1)
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX21,X,VTX21)
      VTX21=-AFACQ5*VTX21
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX22,X,VTX22)
      VTX22=-AFACQ6*VTX22
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX23,X,VTX23)
      VTX23=AFACQ5*VTX23
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX24,X,VTX24)
      VTX24=AFACQ6*VTX24
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX25,X,VTX25)
      VTX25=AFACQ3*VTX25
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX26,X,VTX26)
      VTX26=AFACQ4*VTX26
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX27,X,VTX27)
      VTX27=AFACQ3*VTX27
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX28,X,VTX28)
      VTX28=AFACQ4*VTX28
      VTX2=VTX11+VTX21+VTX22+VTX23+VTX24+VTX25+VTX26+VTX27+VTX28
      VTX=(VTX1+VTX2+VTXSLE+VTXSTE+VTX8)/PI2
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNY11,X,VTY1)
      VTYY1=Q5*VTY1
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNZ11,X,VTZ1)
      VTZ1=Z1*VTZ1
      VTYYZ=VTYZ*(Q5*Y1+Q6*Z1)
      IF(ABS(Z1).LE.1.E-4) GO TO 1399
      VTYZ1=Q6*VTZ1
      VTY=(VTYY1+VTYZ1+VTYYZ)/PI
      VTZY1=-Q6*VTY1
      VTZZ1=Q5*VTZ1
      VTZYZ=VTYZ*(Q5*Z1-Q6*Y1)
      VTZ=(VTZY1+VTZZ1+VTZYZ)/PI
      RETURN
 1399 VTZ=0.
      VTY=(VTYY1+VTYYZ)/PI
      RETURN
 1100 DZWXY=DZWDX(X,Y)
      D2ZWXY=D2ZWDX(X,Y)
      D2ZMXY=D2ZDXY(X,Y)
      AZWXY=DZWXY
      A2ZWXY=D2ZWXY
      A2ZMXY=D2ZMXY
      R2Y=R2/Y1
      SLE1Y1=SLE1-Y1
      Y1STE1=Y1-STE1
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX14,X,VTX4)
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX18,X,VTX8)
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX19,X,VTX9)
      VTX9=-AFACQ5*VTX9/Y1
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX16,X,VTX6)
      VTX6=AFACQ5*VTX6/Y1
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX10,X,VTX10)
      VTX10=AFACQ3*VTX10/Y1
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNX61,X,VTX61)
      VTX61=AFACQ3*VTX61/Y1
      VTX12=2.*D2ZWXY*(SLE1Y1*(ALOG(SLE1Y1/Y1)-1.)+Y1STE1*(ALOG(Y1STE1/
     1Y1)-1.)-(A2MB24/Y1)*((SLE1Y1/SLE1)*ALOG(SLE1Y1/Y1)+(Y1STE1/SLE1)*
     2ALOG(Y1STE1/Y1)-ALOG(SLE1/STE1)))
      VTX=(VTX4+VTX8+VTX9+VTX6+VTX10+VTX61+VTX12+VTX11+VTXSLE+VTXSTE+
     1VTX8)/PI2
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNY13,X,VTY3)
      VTY3=-Q5*VTY3
      CALL SIMPI(STE1,SLE1,NIT,NIT1,TOL,FUNY15,X,VTY5)
      VTY5=Q5*VTY5
      VTY6=-Q5*DZWXY*(ALOG(SLE1Y1/Y1STE1)+A2MB24*((1./Y2)*(ALOG(SLE1/
     1STE1)-ALOG(SLE1Y1/Y1STE1))+(SLE1-STE1)/(Y1*SLE1*STE1)))
      VTYYZ=Q5*Y1*VTYZ
      VTY=(VTY3+VTY5+VTY6+VTYYZ)/PI
      VTZ=DZWXY
      RETURN
 6000 IF(ABS(Z).LE.1.E-4) Z=0.
      CSIG=CMPLX(Y,Z)
      A=AB(X)
      A2MB2=DL1*A*A
      CA=CSQRT(CSIG*CSIG-A2MB2)
      C1=(CSIG+CA)/2.
      Y1=REAL(C1)
      Z1=AIMAG(C1)
      ZY2=Y1*Y1+Z1*Z1
      CQ12=1./(C1*C1-.25*A2MB2)
      CQ34=C1*CQ12
      Q1=REAL(CQ12)
      Q3=REAL(CQ34)
      Q4=AIMAG(CQ34)
      R1=DL4*A
      SEB1=SIEBPI(X)
      VTX=S2EBPI(X)*ALOG(ZY2)-2.*DL*DL1*SEB1*SEB1*Q1
```

```
      VTY=2.*SEB1*Q3                                                    PHTKS468
      VTZ=-2.*SEB1*Q4                                                   PHTKS469
      RETURN                                                            PHTKS470
      END                                                               PHTKS471

      SUBROUTINE PHILFT(X,Y,Z,VLX,VLY,VLZ)                              PHLFT001
      IMPLICIT COMPLEX*8(C)                                             PHLFT002
      COMMON /BLK4/ DL,DL12,DL1,DL2,DL3,DL4,M,MOPT,MELLPT               PHLFT003
      COMMON /BLK6/ PI,PI2,TOL,NIT                                      PHLFT004
      IF(MELLPT.EQ.1) GO TO 300                                         PHLFT005
      XU=SPANLE(X)                                                      PHLFT006
      XL=RB(X)                                                          PHLFT007
      DRDX=DRBDX(X)                                                     PHLFT008
      DSDX=DSPLE(X)                                                     PHLFT009
      R22=XL*XL                                                         PHLFT010
      ZY2=Y**2+Z**2                                                     PHLFT011
   11 SR2=XU+R22/XU                                                     PHLFT012
      RS=XL/XU                                                          PHLFT013
      GN=SR2*(2.*RS*DRDX+(1.-RS*RS)*DSDX)                               PHLFT014
      IF(ABS(ZY2-R22).LE.1.E-5) GO TO 3                                 PHLFT015
      CSIG=CMPLX(Y,Z)                                                   PHLFT016
      C1=R22/CSIG                                                       PHLFT017
      C2=C1/CSIG                                                        PHLFT018
      C3=CSIG+C1                                                        PHLFT019
      C4=C3*C3                                                          PHLFT020
      C5=XL/CSIG                                                        PHLFT021
    4 CNUM=C3*(1.-C2)                                                   PHLFT022
      CDEN=CSQRT(C4-SR2*SR2)                                            PHLFT023
      CF=CNUM/CDEN                                                      PHLFT024
      DF1=REAL(CF)                                                      PHLFT025
      DF2=AIMAG(CF)                                                     PHLFT026
      CG=(2.*C3*C5*DRDX-GN)/CDEN                                        PHLFT027
      DG2=AIMAG(CG)                                                     PHLFT028
      VLX=DG2                                                           PHLFT029
      VLY=DF2                                                           PHLFT030
      VLZ=DF1-1.                                                        PHLFT031
      RETURN                                                            PHLFT032
    3 IF(ABS(Y).LE.1.E-4) GO TO 5                                       PHLFT033
      THETA=ATAN(Z/Y)                                                   PHLFT034
    6 CTHETA=CMPLX(0.,THETA)                                            PHLFT035
      C5=CEXP(-CTHETA)                                                  PHLFT036
      DCOS=REAL(C5)                                                     PHLFT037
      IF(ABS(DCOS).LE.1.E-4) DCOS=0.                                    PHLFT038
      C1=XL*C5                                                          PHLFT039
      C2=CEXP(-2.*CTHETA)                                               PHLFT040
      C3=2.*XL*DCOS                                                     PHLFT041
      C4=C3*C3                                                          PHLFT042
      GO TO 4                                                           PHLFT043
    5 THETA=PI/2.                                                       PHLFT044
      IF(Z.LT..0) GO TO 7                                               PHLFT045
      GO TO 6                                                           PHLFT046
    7 THETA=3.*THETA                                                    PHLFT047
      GO TO 6                                                           PHLFT048
  300 A=AB(X)                                                           PHLFT049
      A2MB2=DL1*A*A                                                     PHLFT050
      S1=SPANLE(X)                                                      PHLFT051
      SCM212=SQRT(S1*S1-A2MB2)                                          PHLFT052
      S1=(S1+SCM212)/2.                                                 PHLFT053
      R1=DL4*A                                                          PHLFT054
      CSIG=CMPLX(Y,Z)                                                   PHLFT055
      CA=CSQRT(CSIG*CSIG-A2MB2)                                         PHLFT056
      C1=(CSIG+CA)/2.                                                   PHLFT057
      YY=REAL(C1)                                                       PHLFT058
      ZZ=AIMAG(C1)                                                      PHLFT059
      ZY2=YY**2+ZZ**2                                                   PHLFT060
      DS1IDX=DSPLE(X)                                                   PHLFT061
      DA1DX=DABDX(X)                                                    PHLFT062
      R2=R1*R1                                                          PHLFT063
      R1S1=R1/S1                                                        PHLFT064
      S1R2S1=S1+R2/S1                                                   PHLFT065
      R1S1SQ=R1S1*R1S1                                                  PHLFT066
      C=CMPLX(YY,ZZ)                                                    PHLFT067
      IF(ABS(R2-ZY2).LE.1.E-5) GO TO 13                                 PHLFT068
      C1=R2/C                                                           PHLFT069
      C2=C1/C                                                           PHLFT070
      C3=C+C1                                                           PHLFT071
      C4=C3*C3                                                          PHLFT072
      C5=C1/R1                                                          PHLFT073
   14 C6=C/(C*C-.25*A2MB2)                                              PHLFT074
      CNUM=C3*(1.-C2)*C6*C                                              PHLFT075
      CDEN=CSQRT(C4-S1R2S1*S1R2S1)                                      PHLFT076
      CF=CNUM/CDEN                                                      PHLFT077
      DF1=REAL(CF)                                                      PHLFT078
      DF2=AIMAG(CF)                                                     PHLFT079
      CSIGX=-.5*DL1*C6*A*DA1DX                                          PHLFT080
      DS1DX=(S1*DS11DX-.5*DL1*A*DA1DX)/SCM212                           PHLFT081
      DR1DX=DL4*DA1DX                                                   PHLFT082
      CGNUM=C3*(1.-C2)*CSIGX-S1R2S1*(1.-R1S1SQ)*DS1DX+2.*(C3*C5-S1R2S1* PHLFT083
     1R1S1)*DR1DX                                                       PHLFT084
      CG=CGNUM/CDEN                                                     PHLFT085
      DG2=AIMAG(CG)                                                     PHLFT086
      VLX=DG2                                                           PHLFT087
      VLY=DF2                                                           PHLFT088
      VLZ=DF1-1.                                                        PHLFT089
      RETURN                                                            PHLFT090
   13 IF(ABS(Y).LE.1.E-4) GO TO 15                                      PHLFT091
      THETA1=ATAN(DL*Z/Y)                                               PHLFT092
   16 CTHET1=CMPLX(0.,THETA1)                                           PHLFT093
      C5=CEXP(-CTHET1)                                                  PHLFT094
      DCOS=REAL(C5)                                                     PHLFT095
      IF(ABS(DCOS).LE.1.E-4) DCOS=0.                                    PHLFT096
      C1=R1*C5                                                          PHLFT097
      C2=CEXP(-2.*CTHET1)                                               PHLFT098
      C3=2.*R1*DCOS                                                     PHLFT099
      C4=C3*C3                                                          PHLFT100
      GO TO 14                                                          PHLFT101
   15 THETA1=PI/2.                                                      PHLFT102
      IF(Z.LT..0) GO TO 17                                              PHLFT103
      GO TO 16                                                          PHLFT104
   17 THETA1=3.*THETA1                                                  PHLFT105
      GO TO 16                                                          PHLFT106
      END                                                               PHLFT107

      SUBROUTINE OUTP(DX,DY,DDERY,IHLF,NDIM,DPRMT)                      OUTP0001
      IMPLICIT COMPLEX*8(C)                                             OUTP0002
      REAL CPU,CPL,COS                                                  OUTP0003
      DIMENSION TKSSTR(10500)                                           OUTP0004
      DIMENSION DY(1),DDERY(1),DPRMT(5)                                 OUTP0005
      DIMENSION CPU(7),CPL(7),RF(7),THETA(5),THETA1(5),THETA2(5),YN1(5),OUTP0006
     1ZN1(5),SCS(5)                                                     OUTP0007
      COMMON /BLK2/ DX11,DX12,DUU1,DUU2,DCD,ALPHA,THETA,THETA1,THETA2,  OUTP0008
     1RF,YN1,ZN1,SCS,DWRTE,UMRM,MMM,M11,NTHETA,IC,ICOPY                 OUTP0009
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  OUTP0010
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    OUTP0011
      COMMON /BLK4/ AL,AL12,AL1,AL2,AL3,AL4,M,MOPT,MELLPT               OUTP0012
      COMMON /BLK5/ XRLE1,XRTE1,HH,H3,NWN                               OUTP0013
      COMMON /BLK6/ PI,PI2,TOL,NIT                                      OUTP0014
      COMMON /BLK9/ MM                                                  OUTP0015
      COMMON /BLK10/ DHF,DEF                                            OUTP0016
      COMMON /BLK11/ RF1,II                                             OUTP0017
      IF(MOPT.EQ.4) GO TO 2000                                          OUTP0018
      IF(M.EQ.2.OR.MOPT.EQ.3) GO TO 3                                   OUTP0019
      IF(M.EQ.3.OR.MOPT.EQ.2) GO TO 11                                  OUTP0020
```

```
      DU=DY(1)                                                          OUTP0021
      UB=DU                                                             OUTP0022
   21 DS1=S1EBPI(DX)                                                    OUTP0023
      DR2=DS1*(1./SEBPI(DX))                                            OUTP0024
      DCP=2.*UB-UR2                                                     OUTP0025
      DUC=DCP*DS1                                                       OUTP0026
      X=DX                                                              OUTP0027
      IF(MMM.EQ.0) GO TO 15                                             OUTP0028
      DDX=ABS(DXOLD-X)                                                  OUTP0029
      DCD=DCD+(DUC+DDCOLD)*DDX/2.                                       OUTP0030
      DXOLD=X                                                           OUTP0031
      DDCOLD=DUC                                                        OUTP0032
      GO TO 14                                                          OUTP0033
   15 DXOLD=X                                                           OUTP0034
      DDCOLD=DUC                                                        OUTP0035
      DCD=0.                                                            OUTP0036
      MMM=1                                                             OUTP0037
   14 IF(IMLF.GT.10) GO TO 1                                            OUTP0038
      IF(M11.EQ.1) RETURN                                               OUTP0039
      IF(ABS(ALPHA).GT.1.E-3.AND.X.GT.XLSM2) GO TO 502                  OUTP0040
      IF(WOPT.EQ.4) GO TO 70                                            OUTP0041
      IF(DX.LE.0.005.OR.DX.GE.0.995) GO TO 20                           OUTP0042
   70 IF(DMRM) 50,20,60                                                 OUTP0043
   50 IF(DX.GT.DWRTE) RETURN                                            OUTP0044
      GO TO 20                                                          OUTP0045
   60 IF(DX.LT.DWRTE) RETURN                                            OUTP0046
   20 X=DX                                                              OUTP0047
      IF(ABS(X-XRLE1).LT.H3.OR.ABS(X-XLSM1).LT.H3) GO TO 501            OUTP0048
      IF(ABS(X-XLSM2).LT.H3.OR.ABS(X-XRTE1).LT.H3) GO TO 501            OUTP0049
      REBSQ=4.*SEBPI(DX)                                                OUTP0050
      REB=SQRT(REBSQ)                                                   OUTP0051
      U2B=2.*S2EBPI(DX)*ALOG(REB)                                       OUTP0052
      IF(MELLPT.EQ.1) GO TO 300                                         OUTP0053
      R=RB(X)                                                           OUTP0054
      RF(1)=R                                                           OUTP0055
      DO 500 II=1,NTHETA                                                OUTP0056
      DO 4 I=1,7                                                        OUTP0057
      Y=YN1(II)*RF(I)                                                   OUTP0058
      Z=ZN1(II)*RF(I)                                                   OUTP0059
      RF1=RF(1)                                                         OUTP0060
      IF((R-RF1).GT..001) GO TO 902                                     OUTP0061
      IF(X.LT.XRLE1) GO TO 900                                          OUTP0062
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 900                             OUTP0063
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 900                             OUTP0064
      IF(MTE.EQ.1) GO TO 1000                                           OUTP0065
      IF(X.LT.XLSM2) GO TO 1002                                         OUTP0066
      S=SPANTE(X)                                                       OUTP0067
      GO TO 1001                                                        OUTP0068
 1000 IF(X.LT.XRTE1) GO TO 1002                                         OUTP0069
      STE=SPANTE(X)                                                     OUTP0070
      YSTE=Y-STE                                                        OUTP0071
      IF(ABS(Z).LE.1.E-3.AND.ABS(YSTE).LE.1.E-3) GO TO 902              OUTP0072
 1002 S=SPANLE(X)                                                       OUTP0073
 1001 YS=Y-S                                                            OUTP0074
      IF(ABS(Z).LE.1.E-3.AND.ABS(YS).LE.1.E-3) GO TO 902                OUTP0075
  900 IF(ICOPY) 5000,5000,6000                                          OUTP0076
 5000 CALL PHITKS(X,Y,Z,VTX,VTY,VTZ)                                    OUTP0077
      GO TO 6001                                                        OUTP0078
   19 IF(ABS(ALPHA).LE.1.E-4) GO TO 40                                  OUTP0079
      IF(X.LT.XRLE1) GO TO 901                                          OUTP0080
      CALL PHILFT(X,Y,Z,VLX,VLY,VLZ)                                    OUTP0081
      VLX=ALPHA*VLX                                                     OUTP0082
      VLY=ALPHA*VLY                                                     OUTP0083
      VLZ=ALPHA*VLZ                                                     OUTP0084
      GO TO 4                                                           OUTP0085
  901 VLX=2.*ALPHA*VTZ                                                  OUTP0086
      VLFAC=ALPHA*REBSQ/(RF1*RF1)                                       OUTP0087
      VLY=-VLFAC*SIN(2.*THETA(II))                                      OUTP0088
      VLZ=VLFAC*COS(2.*THETA(II))                                       OUTP0089
    4 CPU(I)=-2.*(UB-U2B+VTX+VLX+ALPHA*(VTZ+VLZ))-(VTY+VLY)**2-(VTZ+VLZ)OUTP0090
     1**2                                                               OUTP0091
      VTZ=-VTZ                                                          OUTP0092
      VLX=-VLX                                                          OUTP0093
      VLY=-VLY                                                          OUTP0094
      CPL(I)=-2.*(UB-U2B+VTX+VLX+ALPHA*(VTZ+VLZ))-(VTY+VLY)**2-(VTZ+VLZ)OUTP0095
     1**2                                                               OUTP0096
      GO TO 5                                                           OUTP0097
   40 VLX=0.                                                            OUTP0098
      VLY=0.                                                            OUTP0099
      VLZ=0.                                                            OUTP0100
      GO TO 4                                                           OUTP0101
 6000 VTX=TKSSTR(IC)                                                    OUTP0102
      VTY=TKSSTR(IC+1)                                                  OUTP0103
      VTZ=TKSSTR(IC+2)                                                  OUTP0104
      IC=IC+3                                                           OUTP0105
      GO TO 19                                                          OUTP0106
 6001 TKSSTR(IC)=VTX                                                    OUTP0107
      TKSSTR(IC+1)=VTY                                                  OUTP0108
      TKSSTR(IC+2)=VTZ                                                  OUTP0109
      IC=IC+3                                                           OUTP0110
      GO TO 19                                                          OUTP0111
  902 CPU(I)=999999.                                                    OUTP0112
      CPL(I)=999999.                                                    OUTP0113
    5 CONTINUE                                                          OUTP0114
      IF(ABS(THETA1(II)).LE.1.E-4.AND.ABS(ALPHA).GT.1.E-4) GO TO 101    OUTP0115
      WRITE (6,100) X,R,THETA1(II),CPU                                  OUTP0116
  100 FORMAT (1H ,F7.4,2X,F7.4,2X,1PE11.4,7(2X,1PE11.4))                OUTP0117
      IF(ABS(ALPHA).LE.1.E-4) GO TO 500                                 OUTP0118
      WRITE (6,100) X,R,THETA2(II),CPL                                  OUTP0119
      GO TO 500                                                         OUTP0120
  101 IF(X.LT.XRLE1) GO TO 499                                          OUTP0121
      WRITE (6,102) X,R,CPU                                             OUTP0122
  102 FORMAT (1H ,F7.4,2X,F7.4,2X,11H 0.0(UPPER),7(2X,1PE11.4))         OUTP0123
      WRITE (6,103) X,R,CPL                                             OUTP0124
  103 FORMAT (1H ,F7.4,2X,F7.4,2X,11H 0.0(LOWER),7(2X,1PE11.4))         OUTP0125
      GO TO 500                                                         OUTP0126
  499 WRITE (6,100) X,R,THETA1(II),CPU                                  OUTP0127
  500 CONTINUE                                                          OUTP0128
  501 DWRTE=DWRTE+DMRM                                                  OUTP0129
      WRITE (6,104)                                                     OUTP0130
  104 FORMAT (1H )                                                      OUTP0131
      RETURN                                                            OUTP0132
  502 M11=1                                                             OUTP0133
      RETURN                                                            OUTP0134
    3 DU=DY(1)+UHF                                                      OUTP0135
      UB=DU                                                             OUTP0136
      DX11=DX12                                                         OUTP0137
      DX12=DX                                                           OUTP0138
      DUU1=DUU2                                                         OUTP0139
      DUU2=DU                                                           OUTP0140
      GO TO 21                                                          OUTP0141
   11 DU=DY(1)+UEF                                                      OUTP0142
      UB=DU                                                             OUTP0143
      GO TO 21                                                          OUTP0144
    1 WRITE (6,110)                                                     OUTP0145
  110 FORMAT (1H0,74HINTEGRATION TERMINATED BECAUSE ACCUMULATED ERRORS HOUTP0146
     1AVE CAUSED INTEGRATION /1X55HSUBROUTINE TO BISECT ORIGINAL STEP SIOUTP0147
     2ZE (.001) 10 TIMES )                                              OUTP0148
      GO TO 20                                                          OUTP0149
  300 A=AB(X)                                                           OUTP0150
      DO 1500 II=1,NTHETA                                               OUTP0151
      RF(1)=A/SCS(II)                                                   OUTP0152
      R=RF(1)                                                           OUTP0153
      DO 115 I=1,7                                                      OUTP0154
      Y=YN1(II)*RF(I)                                                   OUTP0155
      Z=ZN1(II)*RF(I)                                                   OUTP0156
      RF1=RF(1)                                                         OUTP0157
      IF((R-RF1).GT..001) GO TO 1902                                    OUTP0158
      IF(X.LT.XRLE1) GO TO 1900                                         OUTP0159
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 1900                            OUTP0160
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 1900                            OUTP0161
      IF(MTE.EQ.1) GO TO 1100                                           OUTP0162
      IF(X.LT.XLSM2) GO TO 1102                                         OUTP0163
      S=SPANTE(X)                                                       OUTP0164
```

```
      GO TO 1111                                                        OUTP0165
 1100 IF(X.LT.XRTE1) GO TO 1102                                         OUTP0166
      STE=SPANTE(X)                                                     OUTP0167
      YSTE=Y-STE                                                        OUTP0168
      IF(ABS(Z).LE.1.E-3.AND.ABS(YSTE).LE.1.E-3) GO TO 1902             OUTP0169
 1102 S=SPANLE(X)                                                       OUTP0170
 1111 YS=Y-S                                                            OUTP0171
      IF(ABS(Z).LE.1.E-3.AND.ABS(YS).LE.1.E-3) GO TO 1902               OUTP0172
 1900 IF(ICOPY) 5100,5100,6100                                          OUTP0173
 5100 CALL PHITAS(X,Y,Z,VTX,VTY,VTZ)                                    OUTP0174
      GO TO 6101                                                        OUTP0175
  119 IF(ABS(ALPHA).LE.1.E-4) GO TO 140                                 OUTP0176
      IF(X.LT.XRLE1) GO TO 1901                                         OUTP0177
      CALL PHILFT(X,Y,Z,VLX,VLY,VLZ)                                    OUTP0178
      VLX=ALPHA*VLX                                                     OUTP0179
      VLY=ALPHA*VLY                                                     OUTP0180
      VLZ=ALPHA*VLZ                                                     OUTP0181
      GO TO 114                                                         OUTP0182
 1901 CSIG=CMPLX(Y,Z)                                                   OUTP0183
      A2MR2=AL)*A*A                                                     OUTP0184
      CA=CSQRT(CSIG*CSIG-A2MR2)                                         OUTP0185
      C1=(CSIG+CA)/2.                                                   OUTP0186
      C034=C1/(C1*C1-.25*A2MR2)                                         OUTP0187
      R1=AL4*A                                                          OUTP0188
      SEB1=SIEBPI(DX)                                                   OUTP0189
      CRC=R1/C1                                                         OUTP0190
      CRC2=CRC*CRC                                                      OUTP0191
      CG=-SEB1*(AL*/ .1*(1.+CRC2)*C034+(AL+1.)*(AL+1.)/(AL*C1))         OUTP0192
      CF=(1.+CRC2)*( )34*C1                                             OUTP0193
      VLX=ALPHA*AIM  3(CG)                                              OUTP0194
      VLY=ALPHA*AIM  3(CF)                                              OUTP0195
      VLZ=ALPHA*(RE  .(CF)-1.)                                          OUTP0196
  114 CPU(I)=-2.*(  -U2B+VTX+VLX+ALPHA*(VTZ+VLZ))-(VTY+VLY)**2-(VTZ+VLZ)OUTP0197
     1**2                                                               OUTP0198
      VTZ=-VTZ                                                          OUTP0199
      VLX=-VLX                                                          OUTP0200
      VLY=-VLY                                                          OUTP0201
      CPL(I)=-2.*   -U2B+VTX+VLX+ALPHA*(VTZ+VLZ))-(VTY+VLY)**2-(VTZ+VLZ)OUTP0202
     1**2                                                               OUTP0203
      GO TO 115                                                         OUTP0204
  140 VLX=0.                                                            OUTP0205
      VLY=0.                                                            OUTP0206
      VLZ=0.                                                            OUTP0207
      GO TO 114                                                         OUTP0208
 6100 VTX=TKSST(   .)                                                   OUTP0209
      VTY=TKSST(   :+1)                                                 OUTP0210
      VTZ=TKSST    :+2)                                                 OUTP0211
      IC=IC+3                                                           OUTP0212
      GO TO 119                                                         OUTP0213
 6101 TKSSTR(I(    /TX                                                  OUTP0214
      TKSSTR(I(    )=VTY                                                OUTP0215
      TKSSTR(I     )=VTZ                                                OUTP0216
      IC=IC+3                                                           OUTP0217
      GO TO 11                                                          OUTP0218
 1902 CPU(I)='   999.                                                   OUTP0219
      CPL(I)='   999.                                                   OUTP0220
  115 CONTINU                                                           OUTP0221
      IF(ABS(THETA1(II)).LE.1.E-4.AND.ABS(ALPHA).GT.1.E-4) GO TO 1101   OUTP0222
      WRITE (6,100) X,R,THETA1(II),CPU                                  OUTP0223
      IF(ABS(ALPHA).LE.1.E-4) GO TO 1500                                OUTP0224
      WRITE (6,100) X,R,THETA2(II),CPL                                  OUTP0225
      GO TO 1500                                                        OUTP0226
 1101 IF(X.LT.XRLE1) GO TO 1499                                         OUTP0227
      WRITE (6,102) X,R,CPU                                             OUTP0228
      WRITE (6,103) X,R,CPL                                             OUTP0229
      GO TO 1500                                                        OUTP0230
 1499 WRITE (6,100) X,R,THETA1(II),CPU                                  OUTP0231
 1500 CONTINUE                                                          OUTP0232
      DWRTE=DWRTE+DWRM                                                  OUTP0233
      WRITE (6,104)                                                     OUTP0234
      RETURN                                                            OUTP0235
 2000 UB=UEQBODY(DX)                                                    OUTP0236
      GO TO 21                                                          OUTP0237
      END                                                               OUTP0238
```

```
      FUNCTION RB(X)                                                    RB000001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  RB000002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    RB000003
      COMMON /BLK5/ XRLE1,XRTE1,H,H3,NNN                                RB000004
      COMMON /BLK6/ PI,PI2,TOL,NIT                                      RB000005
      EXTERNAL ZW                                                       RB000006
      N=1                                                               RB000007
      R1=SQRT(4.*SEBPI(X))                                              RB000008
      REQ=R1                                                            RB000009
      IF(NNN.EQ.0) GO TO 2                                              RB000010
    7 IF(MTE.EQ.1) GO TO 10                                             RB000011
      IF(X.LE.XRLE1.OR.X.GE.XRTE1) GO TO 8                              RB000012
      IF(X.LT.XLSM2) XU=SPANLE(X)                                       RB000013
      IF(X.GT.XLSM2) XU=SPANTE(X)                                       RB000014
    5 XL=ROLD                                                           RB000015
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,ZW,X,ANS)                           RB000016
      FR=PI*(ROLD*ROLD-REQ*REQ)+4.*ANS                                  RB000017
      DFDR=PI2*ROLD-4.*ZW(X,ROLD)                                       RB000018
      RNEW=ROLD-FR/DFDR                                                 RB000019
      IF(ABS(RNEW-ROLD).LE.1.E-5) GO TO 4                               RB000020
      ROLD=RNEW                                                         RB000021
      N=N+1                                                             RB000022
      IF(N.GT.11) GO TO 4                                               RB000023
      GO TO 5                                                           RB000024
    4 RB=RNEW                                                           RB000025
      ROLD=RB                                                           RB000026
      IF(RB.LE..0) GO TO 9                                              RB000027
      RETURN                                                            RB000028
    2 ROLD=REQ                                                          RB000029
      NNN=1                                                             RB000030
      GO TO 7                                                           RB000031
    8 RB=REQ                                                            RB000032
      ROLD=RB                                                           RB000033
      RETURN                                                            RB000034
    9 WRITE (6,103) X                                                   RB000035
  103 FORMAT (1H0,51HPROGRAM TERMINATED BECAUSE INDENTED BODY RADIUS HASRB000036
     1 / 1X,31HBECOME LESS THAN ZERO AT X/L = ,E12.5)                   RB000037
      STOP                                                              RB000038
   10 XU=SPANLE(X)                                                      RB000039
      IF(X.LE.XRLE1.OR.X.GE.XLSM2) GO TO 8                              RB000040
      IF(X.LE.XRTE1) GO TO 5                                            RB000041
      XL=SPANTE(X)                                                      RB000042
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,ZW,X,ANS)                           RB000043
      R2B=REQ*REQ-4.*ANS/PI                                             RB000044
      IF(R2B.LE.0.) GO TO 9                                             RB000045
      RB=SQRT(R2B)                                                      RB000046
      ROLD=RB                                                           RB000047
      RETURN                                                            RB000048
      END                                                               RB000049
```

```
      FUNCTION UEQBDY(X)                                                UEQBD001
      DIMENSION XEB(201),UEB(201)                                       UEQBD002
      COMMON /BLK19/ XEB,UEB,NXEB,ITEST                                 UEQBD003
      IF(ITEST.EQ.0) GO TO 10                                           UEQBD004
      IF(XEB(NXEB)-X) 13,13,12                                          UEQBD005
   12 DO 1 J=KK,NXEB                                                    UEQBD006
      IF(XEB(J)-X) 1,2,2                                                UEQBD007
    1 CONTINUE                                                          UEQBD008
    2 UEQBDY=UEB(J-1)+((UEB(J)-UEB(J-1))/(XEB(J)-XEB(J-1)))*(X-XEB(J-1))UEQBD009
      KK=J-1                                                            UEQBD010
      RETURN                                                            UEQBD011
   10 IF((XEB(1)-X).GT.0.) GO TO 11                                     UEQBD012
      ITEST=1                                                           UEQBD013
      KK=2                                                              UEQBD014
      GO TO 12                                                          UEQBD015
   11 UEQBDY=UEB(1)-((UEB(2)-UEB(1))/(XEB(2)-XEB(1)))*(XEB(1)-X)        UEQBD016
      RETURN                                                            UEQBD017
   13 UEQBDY=UEB(NXEB)+((UEB(NXEB)-UEB(NXEB-1))/(XEB(NXEB)-XEB(NXEB-1)))UEQBD018
     1*(X-XEB(NXEB))                                                    UEQBD019
      RETURN                                                            UEQBD020
      END                                                               UEQBD021
```

```
      SUBROUTINE FCT(DX,DY,DZ)                                          FCT00001
      DIMENSION DY(1),DZ(1)                                             FCT00002
      COMMON /BLK1/ DM,DK,D1,D1M,D2M,DZS,DU1,D2A0,DV,DV3,DTAU2          FCT00003
      COMMON /BLK4/ AL,AL12,AL1,AL2,AL3,AL4,M,MOPT,MELLPT               FCT00004
      COMMON /BLK9/ MM                                                  FCT00005
      COMMON /BLK10/ DHF,DEF                                            FCT00006
      DA=SEBP1(DX)                                                      FCT00007
      DA1=S1EBP1(DX)                                                    FCT00008
      DA2=S2EBP1(DX)                                                    FCT00009
      DA3=S3EBP1(DX)                                                    FCT00010
      IF(MOPT.EQ.2) GO TO 11                                            FCT00011
      IF(MOPT.EQ.3) GO TO 12                                            FCT00012
      IF(ABS(DX-DZS).LE.5.E-3) GO TO 10                                 FCT00013
      IF (M.EQ.1) GO TO 1                                               FCT00014
      IF(M.EQ.2) GO TO 2                                                FCT00015
      IF(M.EQ.3) GO TO 3                                                FCT00016
    1 IF(DX-DZS) 20,20,30                                               FCT00017
   20 DZ(1)=DA1*DA2/DA+EXP((DY(1)+D2M-DINT(DX)-DA2*ALOG(D1*DA/DX))/DA2) FCT00018
      RETURN                                                            FCT00019
   30 DUU=D1M+DK*DY(1)                                                  FCT00020
      IF(DUU) 20,20,40                                                  FCT00021
   40 DV1=DA/(DX*DX)                                                    FCT00022
      DV2=DINT(DX)                                                      FCT00023
      D2M=DA3*ALOG(DUU*DV1)+DA1*DA2/DA-2.*D2A0/DX+2.*D1INT(DX)          FCT00024
      DZ(1)=DA1*DA2/DA+EXP((DY(1)+D2M-DV2-DA2*ALOG(D1*DA/DX))/DA2)      FCT00025
      DIFF=D2M-DZ(1)                                                    FCT00026
      IF(DIFF) 60,60,50                                                 FCT00027
   50 RETURN                                                            FCT00028
   60 M=2                                                               FCT00029
      MM=2                                                              FCT00030
      DV=DY(1)-DA2*ALOG(DV1)-2.*DV2                                     FCT00031
      DV3=DX                                                            FCT00032
      RETURN                                                            FCT00033
    2 DHF=DA2*ALOG(DA/(DX*DX))+2.*DINT(DX)                              FCT00034
      DUU=D1M+DK*(DY(1)+DHF)                                            FCT00035
      IF(DUU) 80,80,70                                                  FCT00036
   70 DZ(1)=DA3*ALOG(DUU)                                               FCT00037
      RETURN                                                            FCT00038
   80 M=3                                                               FCT00039
      MM=3                                                              FCT00040
      RETURN                                                            FCT00041
    3 DEF=DA2*ALOG(DA/((DX*(1.-DX))))+DINT(DX)+DINT1(DX)                FCT00042
      DUU=-D1M-DK*(DY(1)+DEF)                                           FCT00043
      DZ(1)=DA3*ALOG(DUU)                                               FCT00044
      RETURN                                                            FCT00045
   10 DZ(1)=DU1                                                         FCT00046
      RETURN                                                            FCT00047
   11 DEF=DA2*ALOG(DA/(DX*(1.-DX)))+DINT(DX)+DINT1(DX)                  FCT00048
      DUU=-D1M-DK*(DY(1)+DEF)                                           FCT00049
      IF(DUU) 22,22,21                                                  FCT00050
   21 DZ(1)=DA3*ALOG(DUU)                                               FCT00051
      RETURN                                                            FCT00052
   22 MM=2                                                              FCT00053
      WRITE(6,700) DX                                                   FCT00054
  700 FORMAT (1H0,61HCALCULATION TERMINATED BECAUSE FLOW ABOUT EQUIVALENFCT00055
     1T BODY HAS / 1X,20HSUPERSONIC AT X/L = ,E12.5)                    FCT00056
      WRITE (6,701)                                                     FCT00057
  701 FORMAT (1H ,45HINPUT MACH NUMBER GREATER THAN LOWER CRITICAL )    FCT00058
      RETURN                                                            FCT00059
   12 DHF=DA2*ALOG(DA/(DX*DX))+2.*DINT(DX)                              FCT00060
      DUU=D1M+DK*(DY(1)+DHF)                                            FCT00061
      IF(DUU) 24,24,23                                                  FCT00062
   23 DZ(1)=DA3*ALOG(DUU)                                               FCT00063
      RETURN                                                            FCT00064
   24 MM=3                                                              FCT00065
      WRITE (6,703) DX                                                  FCT00066
  703 FORMAT (1H0,61HCALCULATION TERMINATED BECAUSE FLOW ABOUT EQUIVALENFCT00067
     1T BODY HAS / 1X,25HBECOME SUBSONIC AT X/L = ,E12.5)               FCT00068
      IF(DX-DZS) 31,41,41                                               FCT00069
   31 WRITE (6,704)                                                     FCT00070
  704 FORMAT (1H ,42HINPUT MACH NUMBER LESS THAN UPPER CRITICAL )       FCT00071
   41 RETURN                                                            FCT00072
      END                                                               FCT00073
```

```
      FUNCTION DINT(DZ)                                                 DINT0001
      EXTERNAL DFUN                                                     DINT0002
      COMMON /BLK8/ DTOL                                                DINT0003
      COMMON /BLK17/ DW,DA2                                             DINT0004
      IF(ABS(DZ).LT.1.E-5) GO TO 25                                     DINT0005
      DW=DZ                                                             DINT0006
      DA2=S2EBP1(DZ)                                                    DINT0007
      NIT=10                                                            DINT0008
      CALL SIMP(0.,DZ,NIT,NIT1,DTOL,DFUN,DANS)                          DINT0009
      DINT=DANS                                                         DINT0010
      RETURN                                                            DINT0011
   25 DINT=0.                                                           DINT0012
      RETURN                                                            DINT0013
      END                                                               DINT0014
```

```
      FUNCTION DINT1(DZ)                                                DINT1001
      EXTERNAL DFUN                                                     DINT1002
      COMMON /BLK8/ DTOL                                                DINT1003
      COMMON /BLK17/ DW,DA2                                             DINT1004
      IF(ABS(1.-DZ).LT.1.E-5) GO TO 25                                  DINT1005
      DW=DZ                                                             DINT1006
      DA2=S2EBP1(DZ)                                                    DINT1007
      NIT=10                                                            DINT1008
      CALL SIMP(1.,DZ,NIT,NIT1,DTOL,DFUN,DANS)                          DINT1009
      DINT1=DANS                                                        DINT1010
      RETURN                                                            DINT1011
   25 DINT1=0.                                                          DINT1012
      RETURN                                                            DINT1013
      END                                                               DINT1014
```

```
      FUNCTION D1INT(DZ)                                                D1INT001
      EXTERNAL DFUN1                                                    D1INT002
      COMMON /BLK8/ DTOL                                                D1INT003
      COMMON /BLK18/ DZ3,DA3                                            D1INT004
      IF(ABS(DZ).LE.1.E-5) GO TO 25                                     D1INT005
      DZ3=DZ                                                            D1INT006
      DA3=S3EBP1(DZ)                                                    D1INT007
      NIT=10                                                            D1INT008
      CALL SIMP(0.,DZ,NIT,NIT1,DTOL,DFUN1,DANS)                         D1INT009
      D1INT=DANS                                                        D1INT010
      RETURN                                                            D1INT011
   25 D1INT=0.                                                          D1INT012
      RETURN                                                            D1INT013
      END                                                               D1INT014
```

```
      FUNCTION DFUN(DZ)                                               DFUN0001
      COMMON /BLK17/ DW,DA2                                           DFUN0002
      IF(ABS(DZ-DW).LT.1.E-5) GO TO 20                                DFUN0003
      DB2=S2EBPI(DZ)                                                  DFUN0004
      DFUN=(DA2-DB2)/(DW-DZ)                                          DFUN0005
      RETURN                                                          DFUN0006
   20 DFUN=S3EBPI(DW)                                                 DFUN0007
      RETURN                                                          DFUN0008
      END                                                             DFUN0009
      FUNCTION DFUN1(DZ)                                              DFUN1001
      COMMON /BLK18/ DZ3,DA3                                          DFUN1002
      IF(ABS(DZ-DZ3).LT.1.E-5) GO TO 20                               DFUN1003
      DB3=S3EBPI(DZ)                                                  DFUN1004
      DFUN1=(DA3-DB3)/(DZ3-DZ)                                        DFUN1005
      RETURN                                                          DFUN1006
   20 DFUN1=S4EBPI(DZ3)                                               DFUN1007
      RETURN                                                          DFUN1008
      END                                                             DFUN1009
```

```
      FUNCTION DLIFT(X)                                               DLIFT001
      COMMON /BLK4/ AL,AL12,AL1,AL2,AL3,AL4,M,MOPT,MELLPT             DLIFT002
      IF(MELLPT.EQ.1) GO TO 1                                         DLIFT003
      S=SPANLE(X)                                                     DLIFT004
      S2=S*S                                                          DLIFT005
      RBA=RB(X)                                                       DLIFT006
      RBA4=RBA**4                                                     DLIFT007
      RBEQ2=4.*SEBPI(X)                                               DLIFT008
      DLIFT=S2+RBA4/S2-RBEQ2                                          DLIFT009
      RETURN                                                          DLIFT010
    1 DS=SPANLE(X)                                                    DLIFT011
      DS2=DS*DS                                                       DLIFT012
      A=AB(X)                                                         DLIFT013
      DC2=AL1*A*A                                                     DLIFT014
      DSC12=DS+SQRT(DS2-DC2)                                          DLIFT015
      DSC122=DSC12*DSC12                                              DLIFT016
      DAPB=2.*AL4*A                                                   DLIFT017
      DREB2=4.*SEBPI(X)                                               DLIFT018
      DLIFT=.25*DSC122*(1.+2.*DC2/DSC122+(DAPB/DSC12)**4)-DREB2       DLIFT019
      RETURN                                                          DLIFT020
      END                                                             DLIFT021
```

```
      FUNCTION DMOMNT(X)                                              DMOMT001
      EXTERNAL DLIFT,SEBPI                                            DMOMT002
      COMMON /BLK4/ AL,AL12,AL1,AL2,AL3,AL4,M,MOPT,MELLPT             DMOMT003
      COMMON /BLK5/ XRLE1,XRTE1,HH,H3,NNN                             DMOMT004
      COMMON /BLK6/ PI,PI2,TOL,NIT                                    DMOMT005
      CALL SIMP(0.,XRLE1,NIT,NIT1,TOL,SEBPI,ANS)                      DMOMT006
      ANS1=4.*ANS                                                     DMOMT007
      IF(MELLPT.EQ.1) ANS1=AL*ANS1                                    DMOMT008
      CALL SIMP(XRLE1,X,NIT,NIT1,TOL,DLIFT,ANS)                       DMOMT009
      DMOMNT=ANS1+ANS                                                 DMOMT010
      RETURN                                                          DMOMT011
      END                                                             DMOMT012
```

```
      FUNCTION SEBPI(DZ)                                              SEBPI001
      COMMON /BLK7/ DCTAU2,DN,K                                       SEBPI002
      IF(K.GT.0) GO TO 2                                              SEBPI003
      DS1=DZ-DZ**DN                                                   SEBPI004
    1 SEBPI=DCTAU2*DS1*DS1/16.                                        SEBPI005
      RETURN                                                          SEBPI006
    2 DS1=1.00-DZ-(1.00-DZ)**DN                                       SEBPI007
      GO TO 1                                                         SEBPI008
      END                                                             SEBPI009
```

```
      FUNCTION S1EBPI(DZ)                                             S1EBP001
      COMMON /BLK7/ DCTAU2,DN,K                                       S1EBP002
      IF (K.GT.0) GO TO 2                                             S1EBP003
      DS1=DZ**(DN-1.00)                                               S1EBP004
    1 S1EBPI=DCTAU2*DZ*(1.-(DN+1.)*DS1+DN*DS1*DS1)/8.                 S1EBP005
      RETURN                                                          S1EBP006
    2 DS1=(1.00-DZ)**(DN-1.00)                                        S1EBP007
      S1EBPI=-DCTAU2*(1.-DZ)*(1.-(DN+1.)*DS1+DN*DS1*DS1)/8.           S1EBP008
      RETURN                                                          S1EBP009
      END                                                             S1EBP010
```

```
      FUNCTION S2EBPI(DZ)                                             S2EBP001
      COMMON /BLK7/ DCTAU2,DN,K                                       S2EBP002
      IF(K.GT.0) GO TO 2                                              S2EBP003
      DS1=DZ**(DN-1.00)                                               S2EBP004
    1 S2EBPI=DCTAU2*(1.-DN*DS1*(DN+1.-(2.*DN-1.)*DS1))/8.             S2EBP005
      RETURN                                                          S2EBP006
    2 DS1=(1.00-DZ)**(DN-1.00)                                        S2EBP007
      GO TO 1                                                         S2EBP008
      END                                                             S2EBP009
```

```
      FUNCTION S3EBPI(DZ)                                             S3EBP001
      COMMON /BLK7/ DCTAU2,DN,K                                       S3EBP002
      IF(K.GT.0) GO TO 2                                              S3EBP003
      DS1=DZ**(DN-2.00)                                               S3EBP004
      DS2=DS1*DS1*DZ                                                  S3EBP005
    1 S3EBPI=DCTAU2*DN*(DN-1.)*(-(DN+1.)*DS1+2.*(2.*DN-1.)*DS2)/8.    S3EBP006
      RETURN                                                          S3EBP007
    2 DS1=-(1.-DZ)**(DN-2.)                                           S3EBP008
      DS2=-DS1*DS1*(1.-DZ)                                            S3EBP009
      GO TO 1                                                         S3EBP010
      END                                                             S3EBP011
```

```
      FUNCTION S4EBPI(DZ)                                               S4EBP001
      COMMON /BLK7/ DLTAU2,DN,K                                         S4EBP002
      IF (K.GT.0) GO TO 2                                               S4EBP003
      DS1=DZ**(DN-3.)                                                   S4EBP004
      DS2=DZ**(2.*DN-4.)                                                S4EBP005
    1 S4EBPI=DLTAU2*DN*(DN-1.)*(-(DN+1.)*(DN-2.)*DS1+2.*(2.*DN-         S4EBP006
     11.)*(2.*(DN-3.)*DS2)/8.                                           S4EBP007
      RETURN                                                            S4FBP008
    2 DS1=(1.-DZ)**(DN-3.)                                              S4EBP009
      DS2=(1.-DZ)**(2.*DN-4.)                                           S4EBP010
      GO TO 1                                                           S4EBP011
      END                                                               S4EBP012
```

```
      FUNCTION AB(X)                                                    AB000001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XHLE,XPTE,TBLE,TBTE,TBLMTE,XCMT,  AB000002
     1XLSM1,XLSM2,SSMAX,DAIN,MTE,MTR                                    AB000003
      COMMON /BLK4/ DL,DL12,DL1,DL2,DL3,DL4,M,MUPT,MFLLPT               AB000004
      COMMON /BLK5/ XRLE1,XRTE1,H,H3,NNN                                AB000005
      COMMON /BLK6/ PI,PI2,TOL,NIT                                      AB000006
      EXTERNAL ZW                                                       AB000007
      N=1                                                               AB000008
      R1=SQRT(4.*S4EBPI(X))                                             AB000009
      REQ=R1                                                            AB000010
      IF(NNN.EQ.0) GO TO 2                                              AB000011
    7 IF(MTE.EQ.1) GO TO 100                                            AB000012
      IF(X.LE.XRLE1.OR.X.GE.XRTE1) GO TO 8                              AB000013
      IF(X.LE.XLSM2) XU=SPANLE(X)                                       AB000014
      IF(X.GT.XLSM2) XU=SPANTE(X)                                       AB000015
    5 XL=AOLD                                                           AB000016
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,ZW,X,ANS)                           AB000017
      FR=PI*(AOLD*AOLD/DL-REQ*REQ)+4.*ANS                               AB000018
      DFDR=PI2*AOLD/DL-4.*ZW(X,AOLD)                                    AB000019
      ANEW=AOLD-FR/DFDR                                                 AB000020
      IF(ABS(ANEW-AOLD).LE.1.E-5) GO TO 4                               AB000021
      AOLD=ANEW                                                         AB000022
      N=N+1                                                             AB000023
      IF(N.GT.11) GO TO 4                                               AB000024
      GO TO 5                                                           AB000025
    4 AB=ANEW                                                           AB000026
      IF(N.GT.11) WRITE (6,201)                                         AB000027
  201 FORMAT (1H ,'AHN GT 11')                                          AB000028
      AOLD=AB                                                           AB000029
      IF(AB.LE.0.) GO TO 10                                             AB000030
      RETURN                                                            AB000031
    2 AOLD=DL12*REQ                                                     AB000032
      NNN=1                                                             AB000033
      GO TO 7                                                           AB000034
    8 AB=DL12*REQ                                                       AB000035
      AOLD=AB                                                           AB000036
      RETURN                                                            AB000037
   10 WRITE (6,11) X                                                    AB000038
   11 FORMAT (1H0,86HPROGRAM TERMINATED BECAUSE INDENTED BODY MAJOR AXISAB000039
     1 HAS BECOME LESS THAN ZERO AT X/L = ,E12.5)                       AB000040
      STOP                                                              AB000041
  100 XU=SPANLE(X)                                                      AB000042
      IF(X.LE.XRLE1.OR.X.GE.XLSM2) GO TO 8                              AB000043
      IF(X.LE.XRTE1) GO TO 5                                            AB000044
      XL=SPANTE(X)                                                      AB000045
      CALL SIMP1(XL,XU,NIT,NIT1,TOL,ZW,X,ANS)                           AB000046
      A2B=DL*(REQ*REQ-4.*ANS/PI)                                        AB000047
      IF(A2B.LE.0.) GO TO 10                                            AB000048
      AB=SQRT(A2B)                                                      AB000049
      AOLD=AB                                                           AB000050
      RETURN                                                            AB000051
      END                                                               AB000052
```

```
      FUNCTION FUNX1(X0,X1)                                             FUNX1001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                       FUNX1002
      Y0=R2/X1                                                          FUNX1003
      Y1=Z2+(Y-X1)*(Y-X1)                                               FUNX1004
      Y2=Z2+(Y+X1)*(Y+X1)                                               FUNX1005
      Y3=Z2+(Y-Y0)*(Y-Y0)                                               FUNX1006
      Y4=Z2+(Y+Y0)*(Y+Y0)                                               FUNX1007
      Z1=ALOG(Y1*Y2*Y3*Y4/ZY8)                                          FUNX1008
      FUNX1=Z1*U2ZWDX(X0,X1)                                            FUNX1009
      RETURN                                                            FUNX1010
      END                                                               FUNX1011
```

```
      FUNCTION FUNX2(X0,X1)                                             FUNX2001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                       FUNX2002
      Y0=R2/X1                                                          FUNX2003
      Y1=Y-Y0                                                           FUNX2004
      Y2=Y+Y0                                                           FUNX2005
      Y3=Z2+Y1*Y1                                                       FUNX2006
      Y4=Z2+Y2*Y2                                                       FUNX2007
      Z1=(Y2/Y4-Y1/Y3)/X1                                               FUNX2008
      FUNX2=Z1*UZWDX(X0,X1)                                             FUNX2009
      RETURN                                                            FUNX2010
      END                                                               FUNX2011
```

```
      FUNCTION FUNX3(X0,X1)                                             FUNX3001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                       FUNX3002
      Y0=R2/X1                                                          FUNX3003
      Y1=Y+X1                                                           FUNX3004
      Y2=Y+Y0                                                           FUNX3005
      Z1=ALOG(Y1*Y2*Y1*Y2/Y22)                                          FUNX3006
      FUNX3=Z1*U2ZWDX(X0,X1)                                            FUNX3007
      RETURN                                                            FUNX3008
      END                                                               FUNX3009
```

```
      FUNCTION FUNX4(X0,X1)                                             FUNX4001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                       FUNX4002
      COMMON /BLK14/ UZWXY,U2ZWXY,D2R,D27R,D2ZWXY,D27WXR                FUNX4003
      Y1=Y-X1                                                           FUNX4004
      IF(ABS(Y1).LE.1.E-4) GO TO 1                                      FUNX4005
      Y1=Y1/Y                                                           FUNX4006
      Y2=Y1*Y1                                                          FUNX4007
      Z1=ALOG(Y2)                                                       FUNX4008
      FUNX4=Z1*(U2ZWDX(X0,X1)-D27WXY)                                   FUNX4009
      RETURN                                                            FUNX4010
    1 FUNX4=0.                                                          FUNX4011
      RETURN                                                            FUNX4012
      END                                                               FUNX4013
```

```
      FUNCTION FUNX5(X0,X1)                                          FUNX5001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNX5002
      COMMON /BLK14/ DZDXY,D2ZDXY,DZR,D2ZR,D2ZMXY,D2ZMXR            FUNX5003
      Y1=(Y-R2/X1)                                                   FUNX5004
      IF(ABS(Y1).LE.1.E-4) GO TO 1                                   FUNX5005
      Y1=Y1/Y                                                        FUNX5006
      Y2=Y1*Y1                                                       FUNX5007
      Z1=ALOG(Y2)                                                    FUNX5008
      FUNX5=Z1*(DZWDX(X0,X1)-DZR)                                    FUNX5009
      RETURN                                                         FUNX5010
    1 FUNX5=0.                                                       FUNX5011
      RETURN                                                         FUNX5012
      END                                                            FUNX5013

      FUNCTION FUNX6(X0,X1)                                          FUNX6001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNX6002
      Z1=1./(X1*R2Y)                                                 FUNX6003
      FUNX6=Z1*DZ*DX(X0,X1)                                          FUNX6004
      RETURN                                                         FUNX6005
      END                                                            FUNX6006

      FUNCTION FUNX7(X0,X1)                                          FUNX7001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNX7002
      COMMON /BLK14/ DZ*XY,D2ZWXY,DZR,D2ZR,D2ZMXY,D2ZMXR             FUNX7003
      Y1=X1-R2Y                                                      FUNX7004
      IF(ABS(Y1).LE.1.E-4) GO TO 1                                   FUNX7005
      Z1=1./Y1                                                       FUNX7006
      FUNX7=Z1*(DZWDX(X0,X1)-DZR)                                    FUNX7007
      RETURN                                                         FUNX7008
    1 FUNX7=D2ZMXR                                                   FUNX7009
      RETURN                                                         FUNX7010
      END                                                            FUNX7011

      FUNCTION FUNX8(X0,X1)                                          FUNX8001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNX8002
      Y0=R2/X1                                                       FUNX8003
      Y1=(Y+X1)**2                                                   FUNX8004
      Y2=(Y-Y0)**2                                                   FUNX8005
      Y3=(Y+Y0)**2                                                   FUNX8006
      Y6=Y**6                                                        FUNX8007
      Z1=ALOG(Y1*Y2*Y3/Y6)                                           FUNX8008
      FUNX8=Z1*D2ZWDX(X0,X1)                                         FUNX8009
      RETURN                                                         FUNX8010
      END                                                            FUNX8011

      FUNCTION FUNX9(X0,X1)                                          FUNX9001
      COMMON /BLK13/ Y,Z,R2,Z8,Y22,ZY8,R2Y,A2MB24                    FUNX9002
      Y1=X1+R2Y                                                      FUNX9003
      Y2=X1-R2Y                                                      FUNX9004
      FUNX9=(1./Y1-1./Y2)*DZWDX(X0,X1)                               FUNX9005
      RETURN                                                         FUNX9006
      END                                                            FUNX9007

      FUNCTION FUNY1(X0,X1)                                          FUNY1001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNY1002
      Y0=R2/X1                                                       FUNY1003
      Y1=Y-X1                                                        FUNY1004
      Y2=Y+X1                                                        FUNY1005
      Y3=Y-Y0                                                        FUNY1006
      Y4=Y+Y0                                                        FUNY1007
      Y5=Z2+Y1*Y1                                                    FUNY1008
      Y6=Z2+Y2*Y2                                                    FUNY1009
      Y7=Z2+Y3*Y3                                                    FUNY1010
      Y8=Z2+Y4*Y4                                                    FUNY1011
      Z1=Y1/Y5+Y2/Y6+Y3/Y7+Y4/Y8                                     FUNY1012
      FUNY1=Z1*DZ*DX(X0,X1)                                          FUNY1013
      RETURN                                                         FUNY1014
      END                                                            FUNY1015

      FUNCTION FUNY2(X0,X1)                                          FUNY2001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNY2002
      Y0=R2/X1                                                       FUNY2003
      Y1=Y+X1                                                        FUNY2004
      Y2=Y+Y0                                                        FUNY2005
      Z1=1./Y1+1./Y2                                                 FUNY2006
      FUNY2=Z1*DZ*DX(X0,X1)                                          FUNY2007
      RETURN                                                         FUNY2008
      END                                                            FUNY2009

      FUNCTION FUNY3(X0,X1)                                          FUNY3001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNY3002
      COMMON /BLK14/ DZWXY,D2ZWXY,DZR,D2ZR,D2ZMXY,D2ZMXR             FUNY3003
      Y1=X1-Y                                                        FUNY3004
      IF (ABS(Y1).LE.1.E-4) GO TO 1                                  FUNY3005
      FUNY3=(DZ*DX(X0,X1)-DZWXY)/Y1                                  FUNY3006
      RETURN                                                         FUNY3007
    1 FUNY3=D2ZMXY                                                   FUNY3008
      RETURN                                                         FUNY3009
      END                                                            FUNY3010

      FUNCTION FUNY4(X0,X1)                                          FUNY4001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNY4002
      COMMON /BLK14/ DZWXY,D2ZWXY,DZR,D2ZR,D2ZMXY,D2ZMXR             FUNY4003
      Y1=X1-R2Y                                                      FUNY4004
      IF(ABS(Y1).LE.1.E-4) GO TO 1                                   FUNY4005
      Z1=X1/Y1                                                       FUNY4006
      FUNY4=Z1*(DZWDX(X0,X1)-DZR)                                    FUNY4007
      RETURN                                                         FUNY4008
    1 FUNY4=X1*D2ZMXR                                                FUNY4009
      RETURN                                                         FUNY4010
      END                                                            FUNY4011

      FUNCTION FUNY5(X0,X1)                                          FUNY5001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24                    FUNY5002
      Y0=R2/X1                                                       FUNY5003
      Z1=1./(Y+X1)+1./(Y-Y0)+1./(Y+Y0)                               FUNY5004
      FUNY5=Z1*DZ*DX(X0,X1)                                          FUNY5005
      RETURN                                                         FUNY5006
      END                                                            FUNY5007
```

```
      FUNCTION FUNZ1(X0,X1)                                     FUNZ1001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FUNZ1002
      Y0=R2/X1                                                  FUNZ1003
      Y1=Z2+(Y-X1)*(Y-X1)                                       FUNZ1004
      Y2=Z2+(Y+X1)*(Y+X1)                                       FUNZ1005
      Y3=Z2+(Y-Y0)*(Y-Y0)                                       FUNZ1006
      Y4=Z2+(Y+Y0)*(Y+Y0)                                       FUNZ1007
      Z1=1./Y1+1./Y2+1./Y3+1./Y4                                FUNZ1008
      FUNZ1=Z1*DZWDX(X0,X1)                                     FUNZ1009
      RETURN                                                    FUNZ1010
      END                                                       FUNZ1011

      FUNCTION FUNX10(X0,X1)                                    FNX10001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX10002
      X3=X1+A2MB24/X1                                           FNX10003
      Z1=1./(X1*(X1-R2Y))                                       FNX10004
      FUNX10=Z1*DZWDX(X0,X3)                                    FNX10005
      RETURN                                                    FNX10006
      END                                                       FNX10007

      FUNCTION FUNX11(X0,X1)                                    FNX11001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX11002
      X2=A2MB24/X1                                              FNX11003
      X3=X1+X2                                                  FNX11004
      X4=X2/X1                                                  FNX11005
      Y0=R2/X1                                                  FNX11006
      Y1=Z2+(Y-X1)*(Y-X1)                                       FNX11007
      Y2=Z2+(Y+X1)*(Y+X1)                                       FNX11008
      Y3=Z2+(Y-Y0)*(Y-Y0)                                       FNX11009
      Y4=Z2+(Y+Y0)*(Y+Y0)                                       FNX11010
      Z1=ALOG(Y1*Y2*Y3*Y4/ZY8)                                  FNX11011
      Z1=(1.-X4)*Z1                                             FNX11012
      FUNX11=Z1*DZZWDX(X0,X3)                                   FNX11013
      RETURN                                                    FNX11014
      END                                                       FNX11015

      FUNCTION FUNX13(X0,X1)                                    FNX13001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX13002
      X2=A2MB24/X1                                              FNX13003
      X3=X1+X2                                                  FNX13004
      X4=X2/X1                                                  FNX13005
      Y0=R2/X1                                                  FNX13006
      Y1=Y+X1                                                   FNX13007
      Y2=Y+Y0                                                   FNX13008
      Z1=ALOG(Y1*Y2*Y1*Y2/Y22)                                  FNX13009
      Z1=(1.-X4)*Z1                                             FNX13010
      FUNX13=Z1*DZZWDX(X0,X3)                                   FNX13011
      RETURN                                                    FNX13012
      END                                                       FNX13013

      FUNCTION FUNX14(X0,X1)                                    FNX14001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX14002
      COMMON /BLK16/ DZWXY,DZZWXY,DZA,DZZA,DZZMXY,DZZMXA        FNX14003
      X2=A2MB24/X1                                              FNX14004
      X3=X1+X2                                                  FNX14005
      X4=X2/X1                                                  FNX14006
      Y1=Y-X1                                                   FNX14007
      IF(ABS(Y1).LE.1.E-4) GO TO 1                              FNX14008
      Y1=Y1/Y                                                   FNX14009
      Y2=Y1*Y1                                                  FNX14010
      Z1=(1.-X4)*ALOG(Y2)                                       FNX14011
      FUNX14=Z1*(DZZWDX(X0,X3)-DZZWXY)                          FNX14012
      RETURN                                                    FNX14013
    1 FUNX14=0.                                                 FNX14014
      RETURN                                                    FNX14015
      END                                                       FNX14016

      FUNCTION FUNX15(X0,X1)                                    FNX15001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX15002
      COMMON /BLK16/ DZWXY,DZZWXY,DZA,DZZA,DZZMXY,DZZMXA        FNX15003
      X2=A2MB24/X1                                              FNX15004
      X3=X1+X2                                                  FNX15005
      X4=X2/X1                                                  FNX15006
      Y1=(Y-R2/X1)                                              FNX15007
      IF(ABS(Y1).LE.1.E-4) GO TO 1                              FNX15008
      Y1=Y1/Y                                                   FNX15009
      Y2=Y1*Y1                                                  FNX15010
      Z1=(1.-X4)*ALOG(Y2)                                       FNX15011
      FUNX15=Z1*(DZZWDX(X0,X3)-DZZA)                            FNX15012
      RETURN                                                    FNX15013
    1 FUNX15=0.                                                 FNX15014
      RETURN                                                    FNX15015
      END                                                       FNX15016

      FUNCTION FUNX16(X0,X1)                                    FNX16001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX16002
      X2=A2MB24/X1                                              FNX16003
      X3=X1+X2                                                  FNX16004
      Z1=1./(X1*R2Y)                                            FNX16005
      FUNX16=Z1*DZWDX(X0,X3)                                    FNX16006
      RETURN                                                    FNX16007
      END                                                       FNX16008

      FUNCTION FUNX61(X0,X1)                                    FNX61001
      FUNX61=FUNX16(X0,X1)/X1                                   FNX61002
      RETURN                                                    FNX61003
      END                                                       FNX61004

      FUNCTION FUNX17(X0,X1)                                    FNX17001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX17002
      COMMON /BLK16/ DZWXY,DZZWXY,DZA,DZZA,DZZMXY,DZZMXA        FNX17003
      X2=A2MB24/X1                                              FNX17004
      X3=X1+X2                                                  FNX17005
      X4=X2/X1                                                  FNX17006
      Y1=X1-R2Y                                                 FNX17007
      IF(ABS(Y1).LE.1.E-4) GO TO 1                              FNX17008
      Z1=1./Y1                                                  FNX17009
      FUNX17=Z1*(DZWDX(X0,X3)-DZA)                              FNX17010
      RETURN                                                    FNX17011
    1 FUNX17=DZZMXA*(1.-X4)                                     FNX17012
      RETURN                                                    FNX17013
      END                                                       FNX17014

      FUNCTION FUNX71(X0,X1)                                    FNX71001
      FUNX71=FUNX17(X0,X1)/X1                                   FNX71002
      RETURN                                                    FNX71003
      END                                                       FNX71004

      FUNCTION FUNX18(X0,X1)                                    FNX18001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24               FNX18002
      Y16=Y**6                                                  FNX18003
      X2=A2MB24/X1                                              FNX18004
      X3=X1+X2                                                  FNX18005
      X4=X2/X1                                                  FNX18006
      Y0=R2/X1                                                  FNX18007
      Y1=(Y+X1)**2                                              FNX18008
      Y2=(Y-Y0)**2                                              FNX18009
      Y3=(Y+Y0)**2                                              FNX18010
      Z1=(1.-X4)*ALOG(Y1*Y2*Y3/Y16)                             FNX18011
      FUNX18=Z1*DZWDX(X0,X3)                                    FNX18012
      RETURN                                                    FNX18013
      END                                                       FNX18014
```

```
      FUNCTION FUNX19(X0,X1)                                  FNX19001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNX19002
      X3=X1+A2MB24/X1                                         FNX19003
      Z1=1./(X1-R2Y)                                          FNX19004
      FUNX19=Z1*DZWDX(X0,X3)                                  FNX19005
      RETURN                                                  FNX19006
      END                                                     FNX19007

      FUNCTION FUNX21(X0,X1)                                  FNX21001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNX21002
      X2=A2MB24/X1                                            FNX21003
      X3=X1+X2                                                FNX21004
      Y0=R2/X1                                                FNX21005
      Y1=Y-Y0                                                 FNX21006
      Y2=Z2+Y1*Y1                                             FNX21007
      Z1=Y1/(Y2*X1)                                           FNX21008
      FUNX21=Z1*DZWDX(X0,X3)                                  FNX21009
      RETURN                                                  FNX21010
      END                                                     FNX21011

      FUNCTION FUNX22(X0,X1)                                  FNX22001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNX22002
      X2=A2MB24/X1                                            FNX22003
      X3=X1+X2                                                FNX22004
      Y0=R2/X1                                                FNX22005
      Y1=Z2+(Y-Y0)*(Y-Y0)                                     FNX22006
      Z1=1./(Y1*X1)                                           FNX22007
      FUNX22=Z1*DZWDX(X0,X3)                                  FNX22008
      RETURN                                                  FNX22009
      END                                                     FNX22010

      FUNCTION FUNX23(X0,X1)                                  FNX23001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNX23002
      X2=A2MB24/X1                                            FNX23003
      X3=X1+X2                                                FNX23004
      Y0=R2/X1                                                FNX23005
      Y1=Y+Y0                                                 FNX23006
      Y2=Z2+Y1*Y1                                             FNX23007
      Z1=Y1/(X1*Y2)                                           FNX23008
      FUNX23=Z1*DZWDX(X0,X3)                                  FNX23009
      RETURN                                                  FNX23010
      END                                                     FNX23011

      FUNCTION FUNX24(X0,X1)                                  FNX24001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNX24002
      X2=A2MB24/X1                                            FNX24003
      X3=X1+X2                                                FNX24004
      Y0=R2/X1                                                FNX24005
      Y1=Z2+(Y+Y0)*(Y+Y0)                                     FNX24006
      Z1=1./(Y1*X1)                                           FNX24007
      FUNX24=Z1*DZWDX(X0,X3)                                  FNX24008
      RETURN                                                  FNX24009
      END                                                     FNX24010

      FUNCTION FUNX25(X0,X1)                                  FNX25001
      FUNX25=FUNX21(X0,X1)*X1                                 FNX25002
      RETURN                                                  FNX25003
      END                                                     FNX25004

      FUNCTION FUNX26(X0,X1)                                  FNX26001
      FUNX26=FUNX22(X0,X1)/X1                                 FNX26002
      RETURN                                                  FNX26003
      END                                                     FNX26004

      FUNCTION FUNX27(X0,X1)                                  FNX27001
      FUNX27=FUNX23(X0,X1)/X1                                 FNX27002
      RETURN                                                  FNX27003
      END                                                     FNX27004

      FUNCTION FUNX28(X0,X1)                                  FNX28001
      FUNX28=FUNX24(X0,X1)/X1                                 FNX28002
      RETURN                                                  FNX28003
      END                                                     FNX28004

      FUNCTION FUNY11(X0,X1)                                  FNY11001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNY11002
      X2=A2MB24/X1                                            FNY11003
      X3=X1+X2                                                FNY11004
      X4=X2/X1                                                FNY11005
      Y0=R2/X1                                                FNY11006
      Y1=Y-X1                                                 FNY11007
      Y2=Y+X1                                                 FNY11008
      Y3=Y-Y0                                                 FNY11009
      Y4=Y+Y0                                                 FNY11010
      Y5=Z2+Y1*Y1                                             FNY11011
      Y6=Z2+Y2*Y2                                             FNY11012
      Y7=Z2+Y3*Y3                                             FNY11013
      Y8=Z2+Y4*Y4                                             FNY11014
      Z1=Y1/Y5+Y2/Y6+Y3/Y7+Y4/Y8                              FNY11015
      Z1=(1.-X4)*Z1                                           FNY11016
      FUNY11=Z1*DZWDX(X0,X3)                                  FNY11017
      RETURN                                                  FNY11018
      END                                                     FNY11019

      FUNCTION FUNY12(X0,X1)                                  FNY12001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNY12002
      X2=A2MB24/X1                                            FNY12003
      X3=X1+X2                                                FNY12004
      X4=X2/X1                                                FNY12005
      Y0=R2/X1                                                FNY12006
      Y1=Y+X1                                                 FNY12007
      Y2=Y+Y0                                                 FNY12008
      Z1=1./Y1+1./Y2                                          FNY12009
      Z1=(1.-X4)*Z1                                           FNY12010
      FUNY12=Z1*DZWDX(X0,X3)                                  FNY12011
      RETURN                                                  FNY12012
      END                                                     FNY12013

      FUNCTION FUNY13(X0,X1)                                  FNY13001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZY8,R2Y,A2MB24             FNY13002
      COMMON /BLK16/ DZWXY,D2ZWXY,DZA,D2ZA,D2ZWXY,D2ZWXA      FNY13003
      X2=A2MB24/X1                                            FNY13004
      X3=X1+X2                                                FNY13005
      X4=X2/X1                                                FNY13006
      Y1=X1-Y                                                 FNY13007
      IF (ABS(Y1).LE.1.E-4) GO TO 1                           FNY13008
      Z1=(1.-X4)/Y1                                           FNY13009
      FUNY13=Z1*(DZWDX(X0,X3)-DZWXY)                          FNY13010
      RETURN                                                  FNY13011
    1 FUNY13=D2ZWXY*(1.-X4)*(1.-X4)                           FNY13012
      RETURN                                                  FNY13013
      END                                                     FNY13014
```

```
      FUNCTION FUNY14(X0,X1)                                              FNY14001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZYB,R2Y,A2MB24                         FNY14002
      COMMON /BLK16/ DZWYY,D2ZWXY,DZA,D2ZA,D2ZMAY,D2ZMXA                  FNY14003
      X2=A2MB24/X1                                                        FNY14004
      X3=X1+X2                                                            FNY14005
      X4=X2/X1                                                            FNY14006
      Y1=X1-R2Y                                                           FNY14007
      IF(ABS(Y1).LE.1.E-4) GO TO 1                                        FNY14008
      Z1=X1*(1.-X4)/Y1                                                    FNY14009
      FUNY14=Z1*(DZWDX(X0,X3)-DZA)                                        FNY14010
      RETURN                                                              FNY14011
    1 FUNY14=X1*D2ZMXA*(1.-X4)*(1.-X4)                                    FNY14012
      RETURN                                                              FNY14013
      END                                                                 FNY14014
```

```
      FUNCTION FUNY15(X0,X1)                                              FNY15001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZYB,R2Y,A2MB24                         FNY15002
      COMMON /BLK16/ DZWXY,D2ZWXY,DZA,D2ZA,D2ZMAY,D2ZMXA                  FNY15003
      X2=A2MB24/X1                                                        FNY15004
      X3=X1+X2                                                            FNY15005
      X4=X2/X1                                                            FNY15006
      Y1=X1-Y                                                             FNY15007
      IF(ABS(Y1).LE.1.E-4) GO TO 1                                        FNY15008
      Z1=(1.-X4)/Y1                                                       FNY15009
      FUNY15=Z1*(DZWDX(X0,X3)-DZWXY)                                      FNY15010
      RETURN                                                              FNY15011
    1 FUNY15=D2ZMAY*(1.-X4)*(1.-X4)                                       FNY15012
      RETURN                                                              FNY15013
      END                                                                 FNY15014
```

```
      FUNCTION FUNZ11(X0,X1)                                              FNZ11001
      COMMON /BLK13/ Y,Z,R2,Z2,Y22,ZYB,R2Y,A2MB24                         FNZ11002
      X2=A2MB24/X1                                                        FNZ11003
      X3=X1+X2                                                            FNZ11004
      X4=X2/X1                                                            FNZ11005
      Y0=R2/X1                                                            FNZ11006
      Y1=Z2+(Y-X1)*(Y-X1)                                                 FNZ11007
      Y2=Z2+(Y+X1)*(Y+X1)                                                 FNZ11008
      Y3=Z2+(Y-Y0)*(Y-Y0)                                                 FNZ11009
      Y4=Z2+(Y+Y0)*(Y+Y0)                                                 FNZ11010
      Z1=1./Y1+1./Y2+1./Y3+1./Y4                                          FNZ11011
      Z1=Z1*(1.-X4)                                                       FNZ11012
      FUNZ11=Z1*DZWDX(X0,X3)                                              FNZ11013
      RETURN                                                              FNZ11014
      END                                                                 FNZ11015
```

```
      FUNCTION SPANLE(X)                                                  SPNLE001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,    SPNLE002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                      SPNLE003
      IF(MTR.EQ.1) GO TO 1                                                SPNLE004
      SPANLE=(X-XRLE)/TBLE                                                SPNLE005
      RETURN                                                              SPNLE006
    1 IF(X.GE.XLSM1) GO TO 2                                              SPNLE007
      SPANLE=(X-XRLE)/TBLE                                                SPNLE008
      RETURN                                                              SPNLE009
    2 SPANLE=SSMAX                                                        SPNLE010
      RETURN                                                              SPNLE011
      END                                                                 SPNLE012
```

```
      FUNCTION SPANTE(X)                                                  SPNTE001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,    SPNTE002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                      SPNTE003
      SPANTE=(X-XRTE)/TBTE                                                SPNTE004
      RETURN                                                              SPNTE005
      END                                                                 SPNTE006
```

```
      FUNCTION DSPLE(X)                                                   DSPLE001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,    DSPLE002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                      DSPLE003
      IF(MTR.EQ.1) GO TO 1                                                DSPLE004
      DSPLE=1./TBLE                                                       DSPLE005
      RETURN                                                              DSPLE006
    1 IF(X.GE.XLSM1) GO TO 2                                              DSPLE007
      DSPLE=1./TBLE                                                       DSPLE008
      RETURN                                                              DSPLE009
    2 DSPLE=0.                                                            DSPLE010
      RETURN                                                              DSPLE011
      END                                                                 DSPLE012
```

```
      FUNCTION DSPTE(X)                                                   DSPTE001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,    DSPTE002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                      DSPTE003
      DSPTE=1./TBTE                                                       DSPTE004
      RETURN                                                              DSPTE005
      END                                                                 DSPTE006
```

```
      FUNCTION ZW(X,Y)                                                    ZW000001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTF,TBLMTE,XCMT,    ZW000002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                      ZW000003
      COMMON /BLK5/ XRLE1,XRTE1,HN,H3,NNN                                 ZW000004
      IF(MTE.EQ.1) GO TO 1000                                             ZW000005
      IF(X.LE.XLSM2) GO TO 1102                                           ZW000006
      S=SPANTE(X)                                                         ZW000007
      GO TO 1103                                                          ZW000008
    3 CL=XTEMLE-TBLMTE*Y                                                  ZW000009
      XBL=X-XRLE-TBLE*Y                                                   ZW000010
      XC=XBL/CL                                                           ZW000011
      IF(XCMT.LT..5) GO TO 1                                              ZW000012
      XCN=XC**DN                                                          ZW000013
      ZW=DAIR*CL*(XC-XCN)                                                 ZW000014
      RETURN                                                              ZW000015
    1 XC=1.-XC                                                            ZW000016
      XCN=XC**DN                                                          ZW000017
      ZW=DAIR*CL*(XC-XCN)                                                 ZW000018
      RETURN                                                              ZW000019
    2 ZW=0.                                                               ZW000020
      RETURN                                                              ZW000021
 1102 S=SPANLE(X)                                                         ZW000022
 1103 IF(ABS(Y-S).LE.1.E-4) GO TO 2                                       ZW000023
      GO TO 3                                                             ZW000024
 1000 IF(X.LE.XRTE1) GO TO 1102                                           ZW000025
      S=SPANTE(X)                                                         ZW000026
      IF(ABS(Y-S).LE.1.E-4) GO TO 2                                       ZW000027
      GO TO 1102                                                          ZW000028
      END                                                                 ZW000029
```

```
      FUNCTION DZWDX(X,Y)                                                DZWDX001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XHLE,XRTE,TBLE,TRTE,TBLMTE,XCNT,  DZWDX002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    DZWDX003
      CL=XTEMLE-TBLMTE*Y                                                 DZWDX004
      XBL=X-XHLE-TBLE*Y                                                  DZWDX005
      XC=XBL/CL                                                          DZWDX006
      IF(XCNT.LT..5) GO TO 1                                             DZWDX007
      IF(ABS(XC).LE.1.E-4) GO TO 2                                       DZWDX008
      XCN1=XC**UN1                                                       DZWDX009
      DZWDX=DAIR*(1.-DN*XCN1)                                            DZWDX010
      RETURN                                                             DZWDX011
    2 DZWDX=DAIR                                                         DZWDX012
      RETURN                                                             DZWDX013
    1 XC=1.-XC                                                           DZWDX014
      IF(ABS(XC).LE.1.E-4) GO TO 3                                       DZWDX015
      XCN1=XC**UN1                                                       DZWDX016
      DZWDX=DAIR*(-1.+DN*XCN1)                                           DZWDX017
      RETURN                                                             DZWDX018
    3 DZWDX=-DAIR                                                        DZWDX019
      RETURN                                                             DZWDX020
      END                                                                DZWDX021

      FUNCTION D2ZWDX(X,Y)                                               D2ZDX001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XHLE,XRTE,TBLE,TBTE,TBLMTE,XCNT,  D2ZDX002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    D2ZDX003
      CL=XTEMLE-TBLMTE*Y                                                 D2ZDX004
      XBL=X-XHLE-TBLE*Y                                                  D2ZDX005
      XC=XBL/CL                                                          D2ZDX006
      IF(XCNT.LT..5) GO TO 1                                             D2ZDX007
      IF(ABS(XC).LE.1.E-4) GO TO 2                                       D2ZDX008
      XCN2=XC**UN2                                                       D2ZDX009
      D2ZWDX=-DAIR*DN*DN1*XCN2/CL                                        D2ZDX010
      RETURN                                                             D2ZDX011
    2 IF(ABS(DN-2.).LE.1.E-4) GO TO 4                                    D2ZDX012
      D2ZWDX=0.                                                          D2ZDX013
      RETURN                                                             D2ZDX014
    4 D2ZWDX=-DAIR*DN*DN1/CL                                             D2ZDX015
      RETURN                                                             D2ZDX016
    1 XC=1.-XC                                                           D2ZDX017
      IF(ABS(XC).LE.1.E-4) GO TO 3                                       D2ZDX018
      XCN2=XC**UN2                                                       D2ZDX019
      D2ZWDX=-DAIR*DN*DN1*XCN2/CL                                        D2ZDX020
      RETURN                                                             D2ZDX021
    3 D2ZWDX=0.                                                          D2ZDX022
      RETURN                                                             D2ZDX023
      END                                                                D2ZDX024

      FUNCTION DZWDY(X,Y)                                                DZWDY001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XHLE,XRTF,TBLE,TBTE,TBLMTE,XCNT,  DZWDY002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    DZWDY003
      CL=XTEMLE-TBLMTE*Y                                                 DZWDY004
      XBL=X-XHLE-TBLE*Y                                                  DZWDY005
      XC=XBL/CL                                                          DZWDY006
      IF(XCNT.LT..5) GO TO 1                                             DZWDY007
      IF(ABS(XC).LE.1.E-4) GO TO 2                                       DZWDY008
      XCN1=XC**UN1                                                       DZWDY009
      XCN=XCN1*XC                                                        DZWDY010
      DZWDY=-DAIR*((1.-DN*XCN1+DN1*XCN)*TBLE-DN1*XCN*TBTE)               DZWDY011
      RETURN                                                             DZWDY012
    2 DZWDY=-DAIR*TBLE                                                   DZWDY013
      RETURN                                                             DZWDY014
    1 XC=1.-XC                                                           DZWDY015
      IF(ABS(XC).LE.1.E-4) GO TO 3                                       DZWDY016
      XCN1=XC**UN1                                                       DZWDY017
      XCN=XCN1*XC                                                        DZWDY018
      DZWDY=-DAIR*(DN1*XCN*TBLE-(1.-XCN-DN*(1.-XC)*XCN1)*TBTE)           DZWDY019
      RETURN                                                             DZWDY020
    3 DZWDY=-DAIR*TBTE                                                   DZWDY021
      RETURN                                                             DZWDY022
      END                                                                DZWDY023

      FUNCTION D2ZDXY(X,Y)                                               D2ZXY001
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XHLE,XRTE,TBLE,TRTE,TBLMTE,XCNT,  D2ZXY002
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    D2ZXY003
      CL=XTEMLE-TBLMTE*Y                                                 D2ZXY004
      XBL=X-XHLE-TBLE*Y                                                  D2ZXY005
      XC=XBL/CL                                                          D2ZXY006
      FAC=DAIR*DN*DN1*((1.-XC)*TBLE+XC*TRTE)/CL                          D2ZXY007
      IF(XCNT.LT..5) GO TO 1                                             D2ZXY008
      IF(ABS(XC).LE.1.E-4) GO TO 2                                       D2ZXY009
      XCN2=XC**UN2                                                       D2ZXY010
      D2ZDXY=FAC*XCN2                                                    D2ZXY011
      RETURN                                                             D2ZXY012
    2 IF(ABS(DN-2.).LE.1.E-4) GO TO 4                                    D2ZXY013
      D2ZDXY=0.                                                          D2ZXY014
      RETURN                                                             D2ZXY015
    4 D2ZDXY=FAC                                                         D2ZXY016
      RETURN                                                             D2ZXY017
    1 XC=1.-XC                                                           D2ZXY018
      IF(ABS(XC).LE.1.E-4) GO TO 3                                       D2ZXY019
      XCN2=XC**UN2                                                       D2ZXY020
      D2ZDXY=FAC*XCN2                                                    D2ZXY021
      RETURN                                                             D2ZXY022
    3 D2ZDXY=0.                                                          D2ZXY023
      RETURN                                                             D2ZXY024
      END                                                                D2ZXY025

      SUBROUTINE CEL1(RES,AK,IER)                                        CEL10001
      IER=0                                                              CEL10002
      GEO=1.-AK*AK                                                       CEL10003
      IF(GEO)1,2,3                                                       CEL10004
    1 IER=1                                                              CEL10005
      RETURN                                                             CEL10006
    2 RES=1.E75                                                          CEL10007
      RETURN                                                             CEL10008
    3 GEO=SQRT(GEO)                                                      CEL10009
      ARI=1.                                                             CEL10010
    4 AARI=ARI                                                           CEL10011
      TEST=AARI*1.E-4                                                    CEL10012
      ARI=GEO+ARI                                                        CEL10013
      IF(AARI-GEO-TEST)6,6,5                                             CEL10014
    5 GEO=SQRT(AARI*GEO)                                                 CEL10015
      ARI=0.5*ARI                                                        CEL10016
      GO TO 4                                                            CEL10017
    6 RES=3.14159265/ARI                                                 CEL10018
      RETURN                                                             CEL10019
      END                                                                CEL10020

      SUBROUTINE SIMP(DXL,DXU,NIT,NIT1,DTOL,DFUN,DANS)                   SIMP0001
      NIT1=10                                                            SIMP0002
      DH=(DXU-DXL)/2.                                                    SIMP0003
      IF(ABS(DH).LT.1.E-5) GO TO 4                                       SIMP0004
      DSUM1=DFUN(DXL)+DFUN(DXU)                                          SIMP0005
      DSUM2=DFUN(DXL+DH)                                                 SIMP0006
      DANS=DH*(DSUM1+4.*DSUM2)/3.                                        SIMP0007
      N=2                                                                SIMP0008
      DO 3 I=1,NIT                                                       SIMP0009
      DANS1=DANS                                                         SIMP0010
      N=N*2                                                              SIMP0011
      DH=DH/2.                                                           SIMP0012
      DSUM3=0.                                                           SIMP0013
      NLIM=N-1                                                           SIMP0014
      DO 1 K=1,NLIM,2                                                    SIMP0015
      DF=K                                                               SIMP0016
    1 DSUM3=DSUM3+DFUN(DXL+DH*DF)                                        SIMP0017
      DANS=DH*(DSUM1+2.*DSUM2+4.*DSUM3)/3.                               SIMP0018
```

```
      IF(ABS(UANS).LT.1.E-2) GO TO 2                                    SIMP0019
      UERR=ABS(UANS1/UANS-1.)                                           SIMP0020
      IF(UERR.LE.UTOL) RETURN                                           SIMP0021
      GO TO 3                                                           SIMP0022
    2 DERR=ABS(UANS-DANS1)                                              SIMP0023
      IF(DERR.LT.1.E-4) RETURN                                          SIMP0024
    3 DSUM2=DSUM2+DSUM3                                                 SIMP0025
      IF(ABS(UANS).LE.1.E-5) RETURN                                     SIMP0026
      NIT1=0                                                            SIMP0027
      RETURN                                                            SIMP0028
    4 DANS=0.                                                           SIMP0029
      RETURN                                                            SIMP0030
      END                                                               SIMP0031

      SUBROUTINE SIMP1(XL,XU,NIT,NIT1,TOL,FUN,X1,ANS)                   SIMP1001
      NIT1=10                                                           SIMP1002
      H=(XU-XL)/2.                                                      SIMP1003
      IF(ABS(H).LE.1.E-6) GO TO 4                                       SIMP1004
      SUM1=FUN(X1,XL)                                                   SIMP1005
      SUM1=SUM1+FUN(X1,XU)                                              SIMP1006
      SUM2=FUN(X1,XL+H)                                                 SIMP1007
      ANS=H*(SUM1+4.*SUM2)/3.                                           SIMP1008
      N=2                                                               SIMP1009
      DO 3 I=1,NIT                                                      SIMP1010
      ANS1=ANS                                                          SIMP1011
      N=N*2                                                             SIMP1012
      H=H/2.                                                            SIMP1013
      SUM3=0.                                                           SIMP1014
      NLIM=N-1                                                          SIMP1015
      DO 1 K=1,NLIM,2                                                   SIMP1016
      F=K                                                               SIMP1017
    1 SUM3 = SUM3+FUN(X1,XL+H*F)                                        SIMP1018
      ANS=H*(SUM1+2.*SUM2+4.*SUM3)/3.                                   SIMP1019
      IF(ABS(ANS).LT.1.E-2) GO TO 2                                     SIMP1020
      ERR=ABS(ANS1/ANS-1.)                                              SIMP1021
      IF(ERR.LE.TOL) RETURN                                             SIMP1022
      GO TO 3                                                           SIMP1023
    2 ERR=ABS(ANS-ANS1)                                                 SIMP1024
      IF(ERR.LE.1.E-5) RETURN                                           SIMP1025
    3 SUM2=SUM2+SUM3                                                    SIMP1026
      NIT1=0                                                            SIMP1027
      WRITE (6,999)                                                     SIMP1028
  999 FORMAT (1H ,'NIT1 = 0')                                           SIMP1029
      RETURN                                                            SIMP1030
    4 ANS=0.                                                            SIMP1031
      RETURN                                                            SIMP1032
      END                                                               SIMP1033

      FUNCTION DRBDX(X)                                                 DRBDX001
      DOUBLE PRECISION RB2,RB1,RF1,RF2,DNUM                             DRBDX002
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  DRBDX003
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    DRBDX004
      COMMON /BLK5/ XRLE1,XRTE1,H,H3,NNN                                DRBDX005
      IF(X.LT.XRLE1) GO TO 4                                            DRBDX006
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 4                               DRBDX007
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 4                               DRBDX008
      RB2=RB(X-2.*H)                                                    DRBDX009
      RB1=RB(X-H)                                                       DRBDX010
      RF1=RB(X+H)                                                       DRBDX011
      RF2=RB(X+2.*H)                                                    DRBDX012
      DNUM=-RB2-2.*RB1+2.*RF1+RF2                                       DRBDX013
      DRBDX=DNUM/(8.*H)                                                 DRBDX014
      RETURN                                                            DRBDX015
    4 R=SQRT(4.*SCBPI(X))                                               DRBDX016
      DRBDX=2.*S1EBPI(X)/R                                              DRBDX017
      RETURN                                                            DRBDX018
      END                                                               DRBDX019

      FUNCTION D2RBDX(X)                                                D2RDX001
      DOUBLE PRECISION RB4,RB3,RB2,RB1,RB0,RF1,RF2,RF3,RF4,DNUM         D2RDX002
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  D2RDX003
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    D2RDX004
      COMMON /BLK5/ XRLE1,XRTE1,H,H3,NNN                                D2RDX005
      IF(X.LT.XRLE1) GO TO 4                                            D2RDX006
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 4                               D2RDX007
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 4                               D2RDX008
      RB4=RB(X-4.*H)                                                    D2RDX009
      RB3=RB(X-3.*H)                                                    D2RDX010
      RB2=RB(X-2.*H)                                                    D2RDX011
      RB1=RB(X-H)                                                       D2RDX012
      RB0=RB(X)                                                         D2RDX013
      RF1=RB(X+H)                                                       D2RDX014
      RF2=RB(X+2.*H)                                                    D2RDX015
      RF3=RB(X+3.*H)                                                    D2RDX016
      RF4=RB(X+4.*H)                                                    D2RDX017
      DNUM=RB4+4.*RB3+4.*RB2-4.*RB1-10.*RB0-4.*RF1+4.*RF2+4.*RF3+RF4    D2RDX018
      D2RBDX=DNUM/(64.*H*H)                                             D2RDX019
      RETURN                                                            D2RDX020
    4 R=SQRT(4.*SCBPI(X))                                               D2RDX021
      DRDX=2.*S1EBPI(X)/R                                               D2RDX022
      D2RBDX=(2.*S2EBPI(X)-DRDX*DRDX)/R                                 D2RDX023
      RETURN                                                            D2RDX024
      END                                                               D2RDX025

      FUNCTION DABDX(X)                                                 DABDX001
      DOUBLE PRECISION RB2,RB1,RF1,RF2,DNUM                             DABDX002
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  DABDX003
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    DABDX004
      COMMON /BLK4/ DL,DL12,DL1,DL2,DL3,DL4,M,MUPT,MELLPT               DABDX005
      COMMON /BLK5/ XRLE1,XRTE1,H,H3,NNN                                DABDX006
      IF(X.LT.XRLE1) GO TO 4                                            DABDX007
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 4                               DABDX008
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 4                               DABDX009
      RB2=AB(X-2.*H)                                                    DABDX010
      RB1=AB(X-H)                                                       DABDX011
      RF1=AB(X+H)                                                       DABDX012
      RF2=AB(X+2.*H)                                                    DABDX013
      DNUM=-RB2-2.*RB1+2.*RF1+RF2                                       DABDX014
      DABDX=DNUM/(8.*H)                                                 DABDX015
      RETURN                                                            DABDX016
    4 R=SQRT(4.*SCBPI(X))                                               DABDX017
      DABDX=DL12*2.*S1EBPI(X)/R                                         DABDX018
      RETURN                                                            DABDX019
      END                                                               DABDX020

      FUNCTION D2ABDX(X)                                                D2ADX001
      DOUBLE PRECISION RB4,RB3,RB2,RB1,RB0,RF1,RF2,RF3,RF4,DNUM         D2ADX002
      COMMON /BLK3/ DN,DN1,DN2,XTEMLE,XRLE,XRTE,TBLE,TBTE,TBLMTE,XCMT,  D2ADX003
     1XLSM1,XLSM2,SSMAX,DAIR,MTE,MTR                                    D2ADX004
      COMMON /BLK4/ DL,DL12,DL1,DL2,DL3,DL4,M,MUPT,MELLPT               D2ADX005
      COMMON /BLK5/ XRLE1,XRTE1,H,H3,NNN                                D2ADX006
      IF(X.LT.XRLE1) GO TO 4                                            D2ADX007
      IF(MTE.EQ.1.AND.X.GT.XLSM2) GO TO 4                               D2ADX008
      IF(MTE.NE.1.AND.X.GT.XRTE1) GO TO 4                               D2ADX009
      RB4=AB(X-4.*H)                                                    D2ADX010
      RB3=AB(X-3.*H)                                                    D2ADX011
      RB2=AB(X-2.*H)                                                    D2ADX012
      RB1=AB(X-H)                                                       D2ADX013
      RB0=AB(X)                                                         D2ADX014
      RF1=AB(X+H)                                                       D2ADX015
      RF2=AB(X+2.*H)                                                    D2ADX016
      RF3=AB(X+3.*H)                                                    D2ADX017
      RF4=AB(X+4.*H)                                                    D2ADX018
      DNUM=RB4+4.*RB3+4.*RB2-4.*RB1-10.*RB0-4.*RF1+4.*RF2+4.*RF3+RF4    D2ADX019
      D2ABDX=DNUM/(64.*H*H)                                             D2ADX020
      RETURN                                                            D2ADX021
    4 R=SQRT(4.*SCBPI(X))                                               D2ADX022
      DRDX=2.*S1EBPI(X)/R                                               D2ADX023
      D2ABDX=DL12*(2.*S2EBPI(X)-DRDX*DRDX)/R                            D2ADX024
      RETURN                                                            D2ADX025
      END                                                               D2ADX026
```

REFERENCES

1. Steger, J. L. and Lomax, H.: Generalized Relaxation Methods Applied to Problems in Transonic Flow. Proc. 2nd Int. Conf. on Numerical Methods in Fluid Dynamics, Sept. 1970; Lecture Notes in Physics, 1971, pp. 193-197.

2. Garabedian, P. R. and Korn, D. G.: Numerical Design of Transonic Airfoils. Numerical Solution of Partial Differential Equations - II, Bert Hubbard, ed., Academic Press, Inc., 1971, pp. 253-271.

3. Murman, E. M. and Cole, J. D.: Calculation of Plane Steady Transonic Flows. AIAA J., Vol. 9, No. 1, Jan. 1971, pp. 114-121.

4. Murman, E. M. and Krupp, J. A.: The Numerical Calculation of Steady Transonic Flows Past Thin Lifting Airfoils and Slender Bodies. AIAA Paper No. 71-566, Presented at AIAA 4th Fluid and Plasma Dynamics Conf., Palo Alto, Calif., June 21-23, 1971

5. Bailey, F. R.: Numerical Calculation of Transonic Flow About Slender Bodies of Revolution. NASA TN D-6582, Dec. 1971.

6. Spreiter, J. R. and Stahara, S. S.: Calculative Techniques for Transonic Flows About Certain Classes of Airfoils and Slender Bodies. NASA CR-1722, 1971.

7. Stahara, S. S. and Spreiter, J. R.: Calculative Techniques for Transonic Flows About Certain Classes of Wing-Body Combinations. NEAR TR 36, Nov. 1971. NASA CR-2103, 1972.

8. Heaslet, M. A. and Spreiter, J. R.: Three-Dimensional Transonic Flow Theory Applied to Slender Wings and Bodies. NACA Rep. 1318, 1957.

9. Adams, M. C. and Sears, W. R.: Slender-Body Theory - Review and Extensions. J. Aero. Sci., Vol. 20, No. 2, 1953, pp. 85-98.

10. Stocker, P. M.: Supersonic Flow Past Bodies of Revolution with Thin Wings of Small Aspect Ratio. Aero. Quart., Vol. III, May 1951, pp. 61-79.

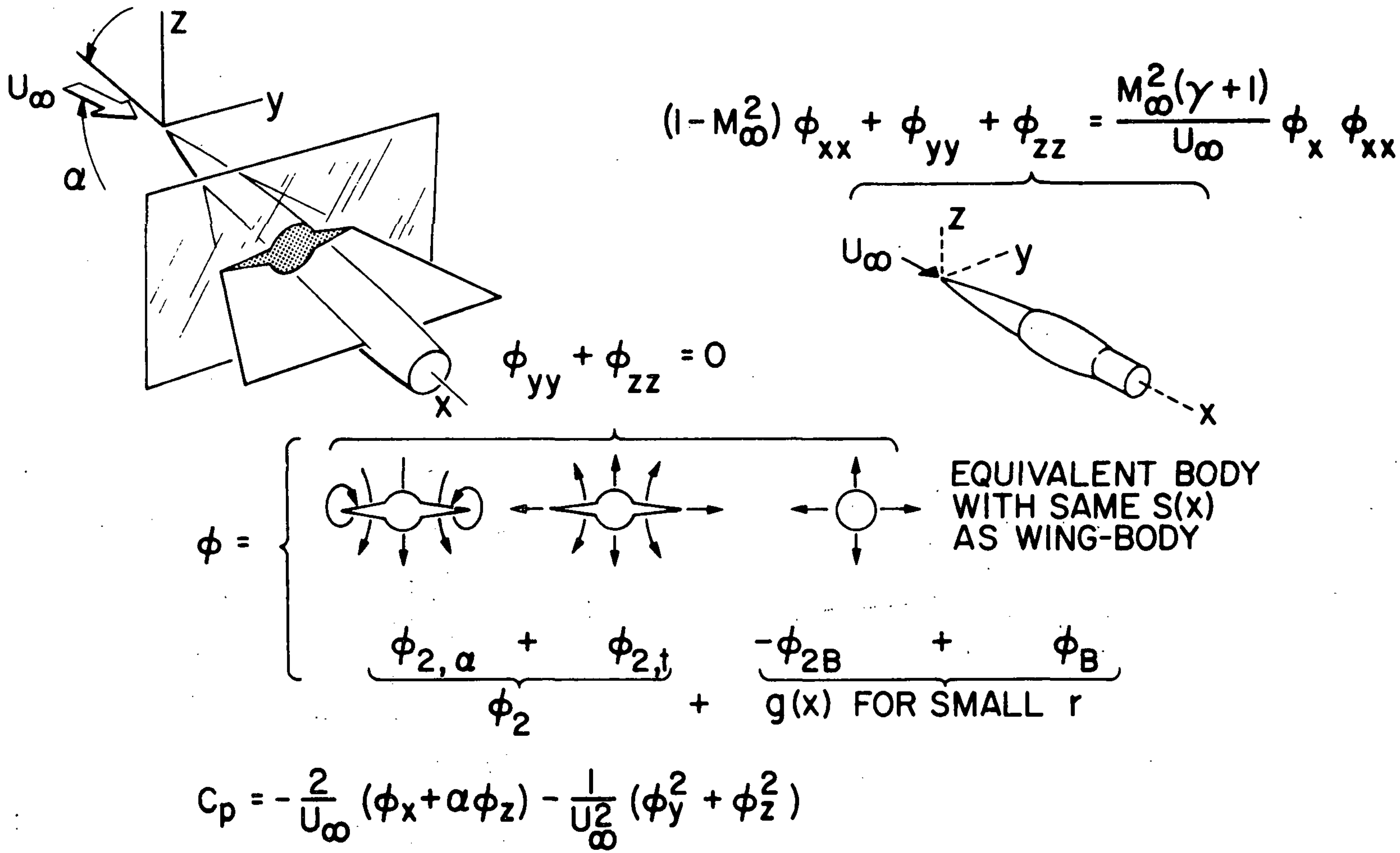


Figure 1.- Transonic equivalence rule for slender wing-body combinations.

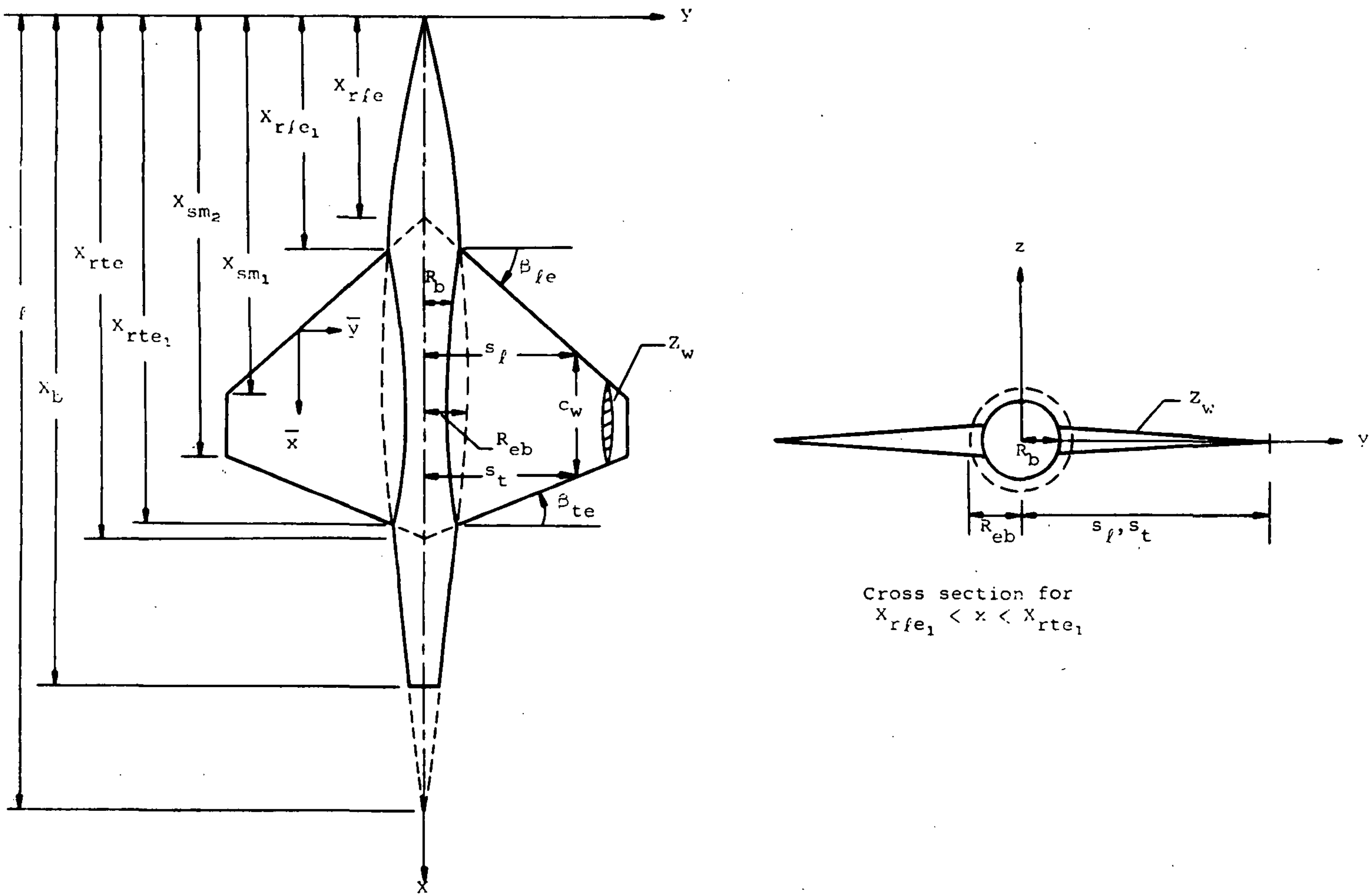


Figure 2(a).- General configuration of finite thickness wing-indented circular body combinations with straight/sweptforward trailing edge planforms ($\beta_{te} \leq 0$).

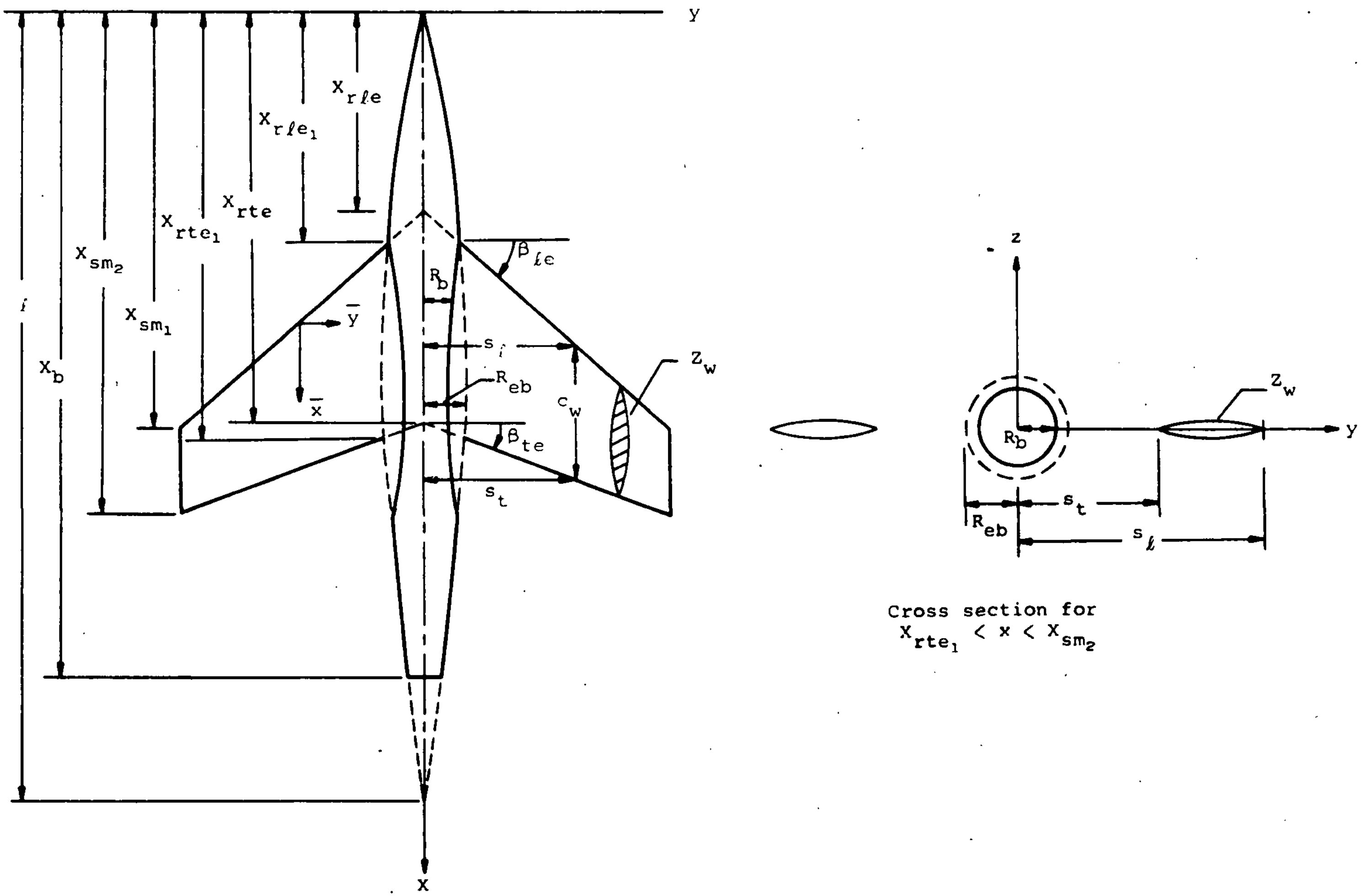


Figure 2(b). - General configuration of finite thickness wing-indented circular body combinations with sweptback trailing edge planforms ($\beta_{te} > 0$).

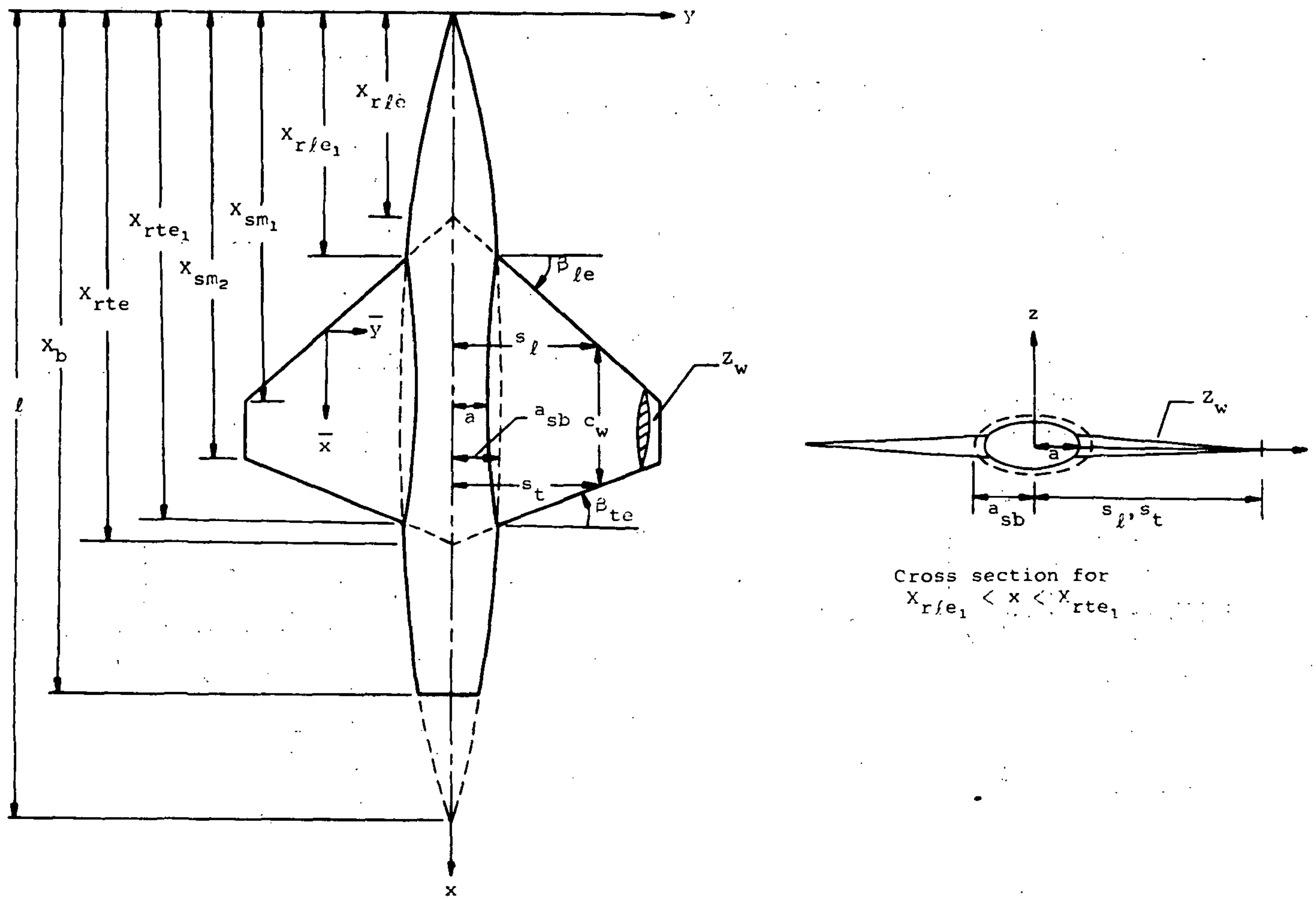


Figure 3(a).- General configuration of finite thickness wing-indented elliptic body combinations with straight/sweptforward trailing edge planforms ($\beta_{te} \lesssim 0$).

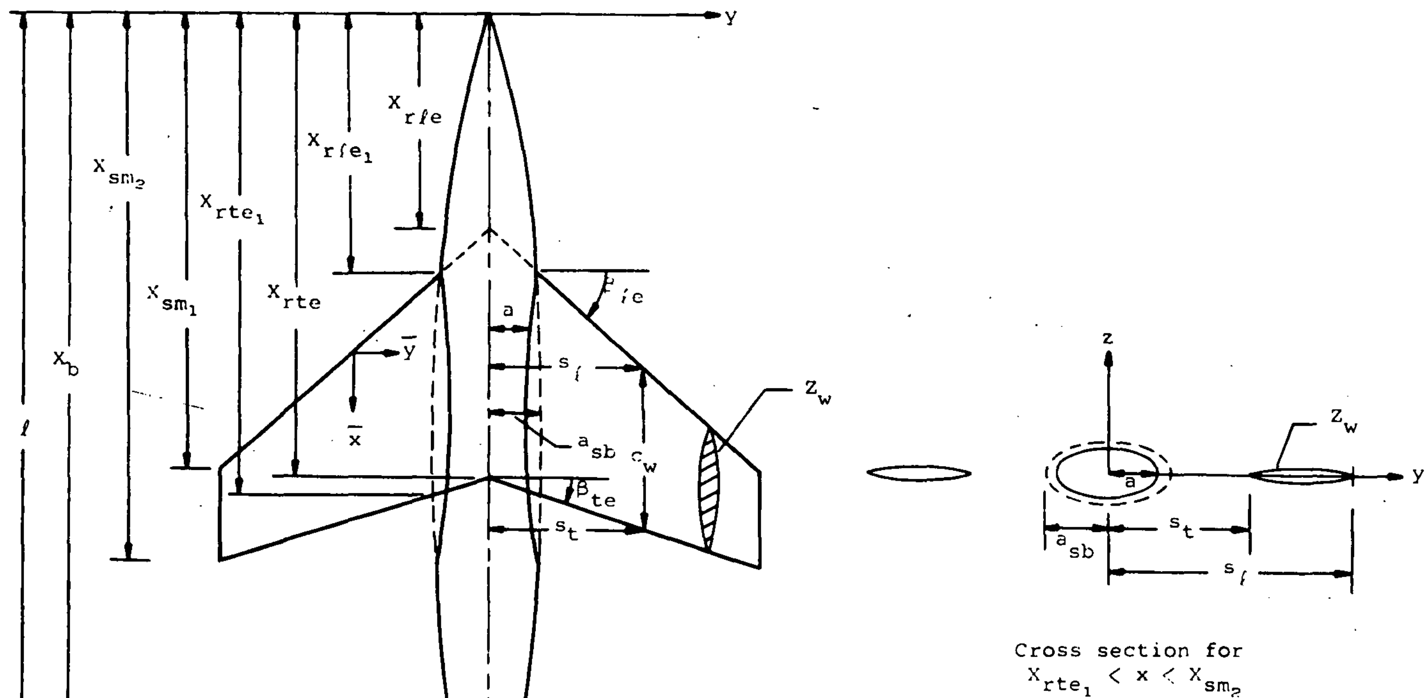


Figure 3(b).- General configuration of finite thickness wing-indented elliptic body combinations with sweptback trailing edge planforms ($\beta_{te} > 0$).

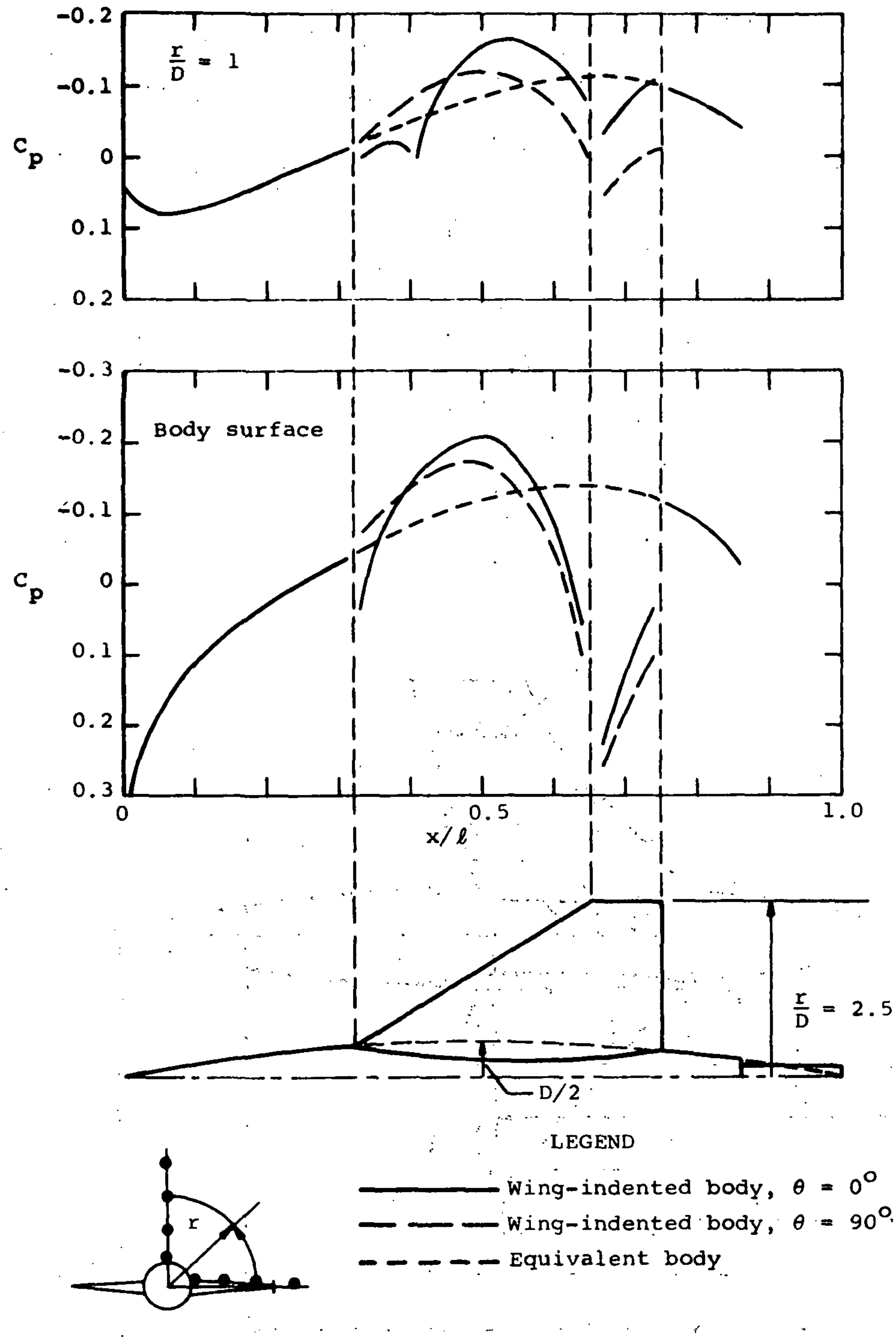


Figure 4.- Theoretical surface and flow field pressure distributions at $M_{\infty} = 1$ for a nonlifting parabolic-arc profile wing--indented parabolic-arc body combination; equivalent body thickness ratio $D/\ell = 0.1$, wing aspect ratio $AR = 1.7$, thickness/chord ratio $t/c_w = 0.04$, planform taper ratio $TR = 0.2$, and with $X_{r\ell e/\ell} = 0.25$, $C_{R_T} = 0.50$, $X_{b/\ell} = 0.86$.

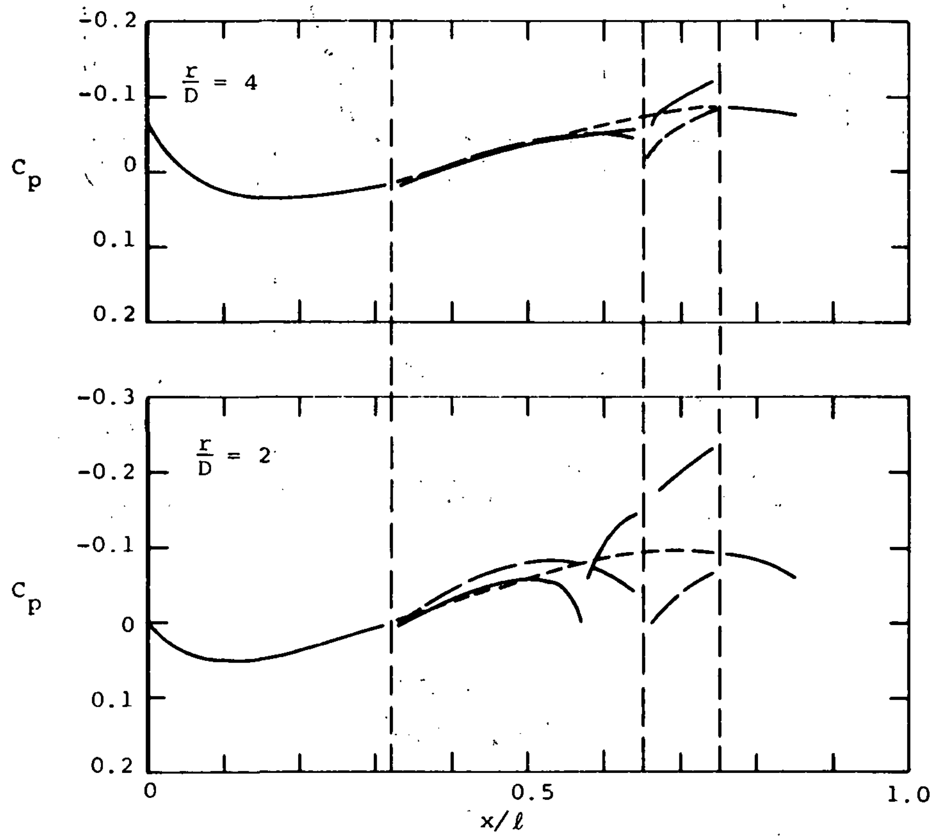


Figure 4.- Concluded

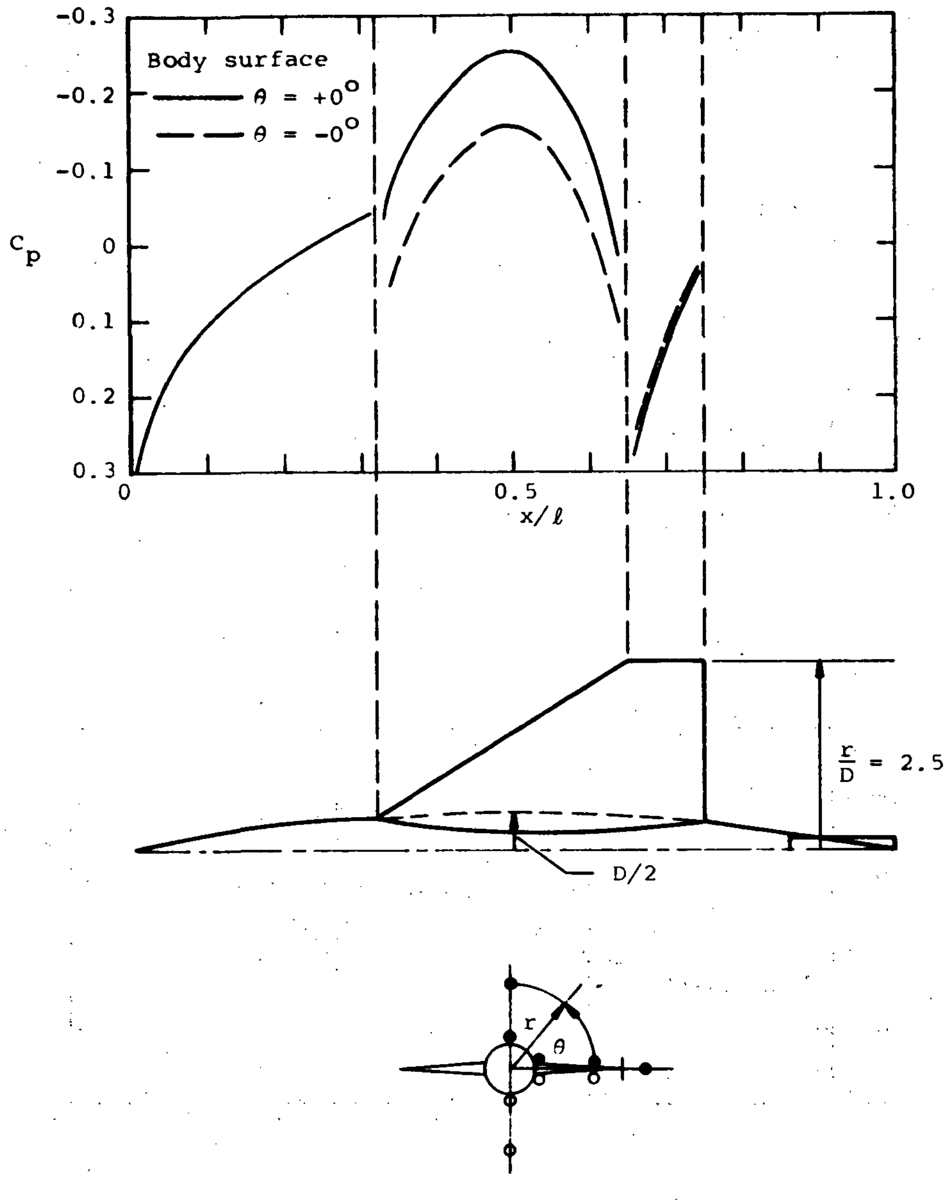


Figure 5.- Theoretical surface and flow field pressure distributions and loadings at $M_\infty = 1$ and $\alpha = 2^\circ$ for a parabolic-arc profile wing--indented parabolic-arc body combination; equivalent body thickness ratio $D/\ell = 0.1$, wing aspect ratio $AR = 1.7$, thickness/chord ratio $t/c_w = 0.04$, planform taper ratio $TR = 0.2$, and with $X_{r\ell e/\ell} = 0.25$, $c_{R_{T/\ell}} = 0.5$, $X_{b/\ell} = 0.86$.

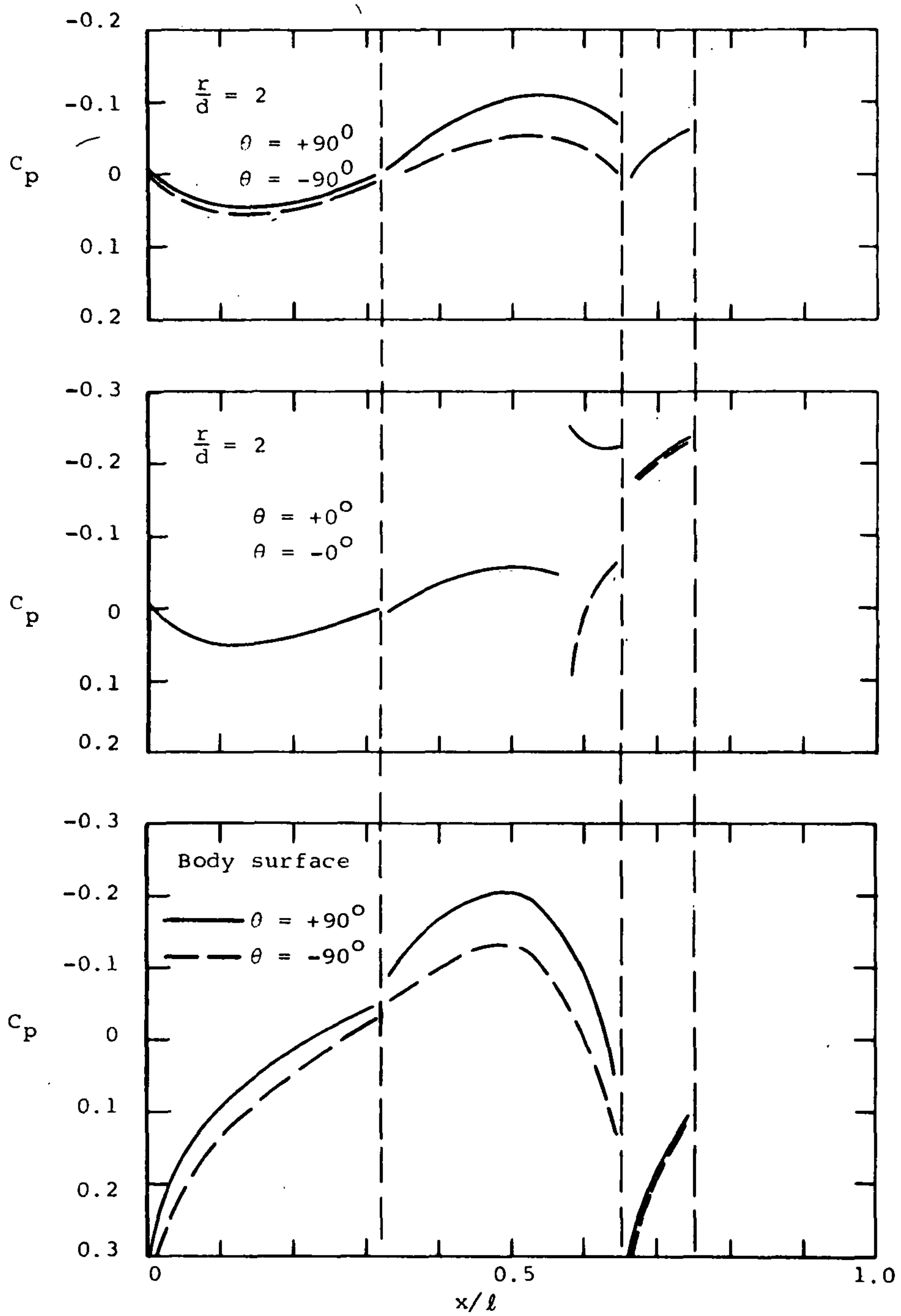


Figure 5.- Continued.

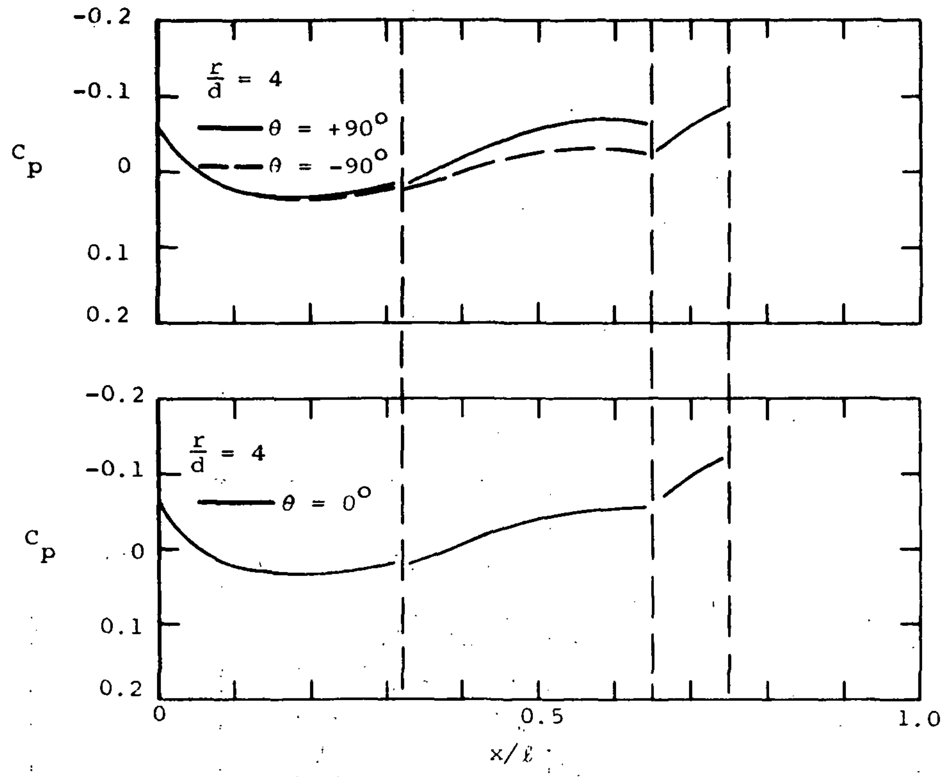


Figure 5.- Concluded.

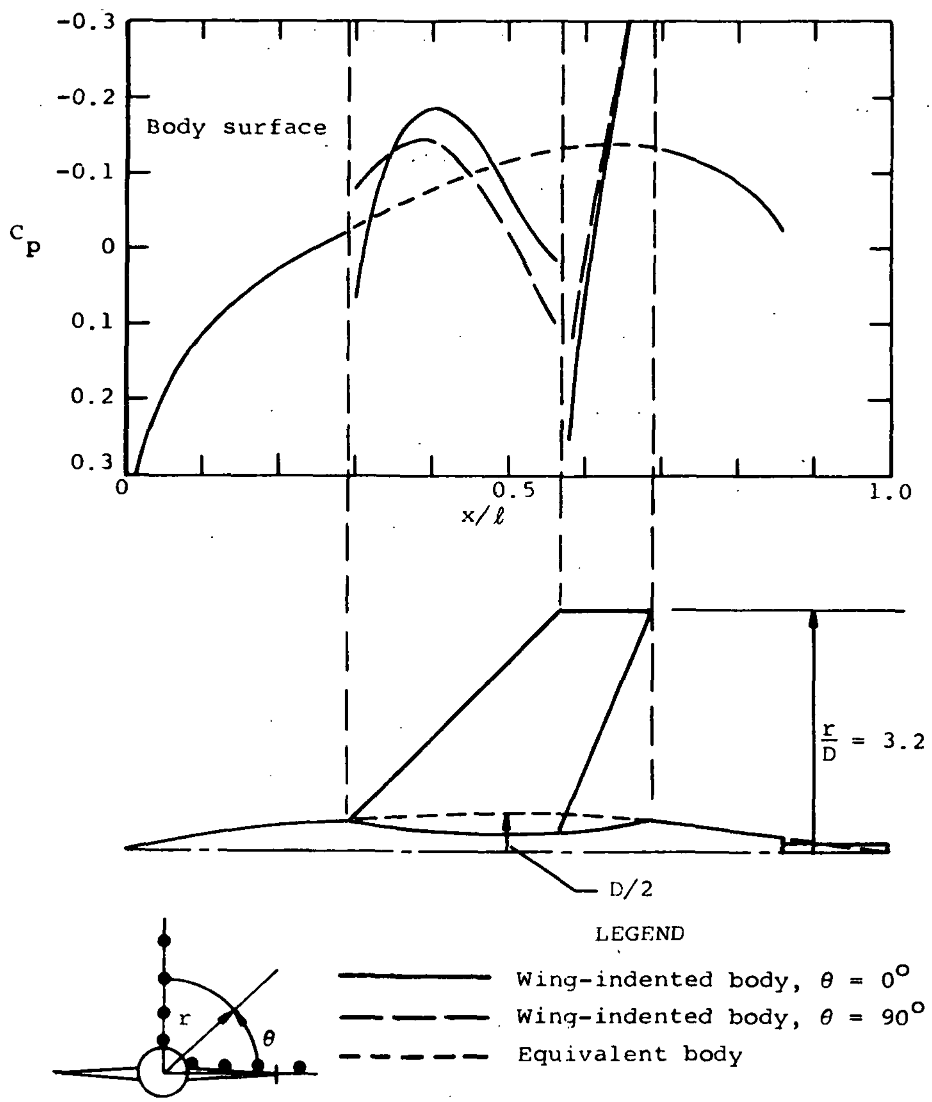


Figure 6.- Theoretical surface and flow field pressure distributions at $M_{\infty} = 1$ for a nonlifting parabolic-arc profile wing--indented parabolic-arc body combination; equivalent body thickness ratio $D/\ell = 0.1$, wing aspect ratio $AR = 2.8$, thickness ratio $t/c_w = 0.04$, planform taper ratio $TR = 0.4$, and with $X_{r\ell e/\ell} = 0.25$, $C_{T_R} = 0.3$, $X_{b/\ell} = 0.86$.

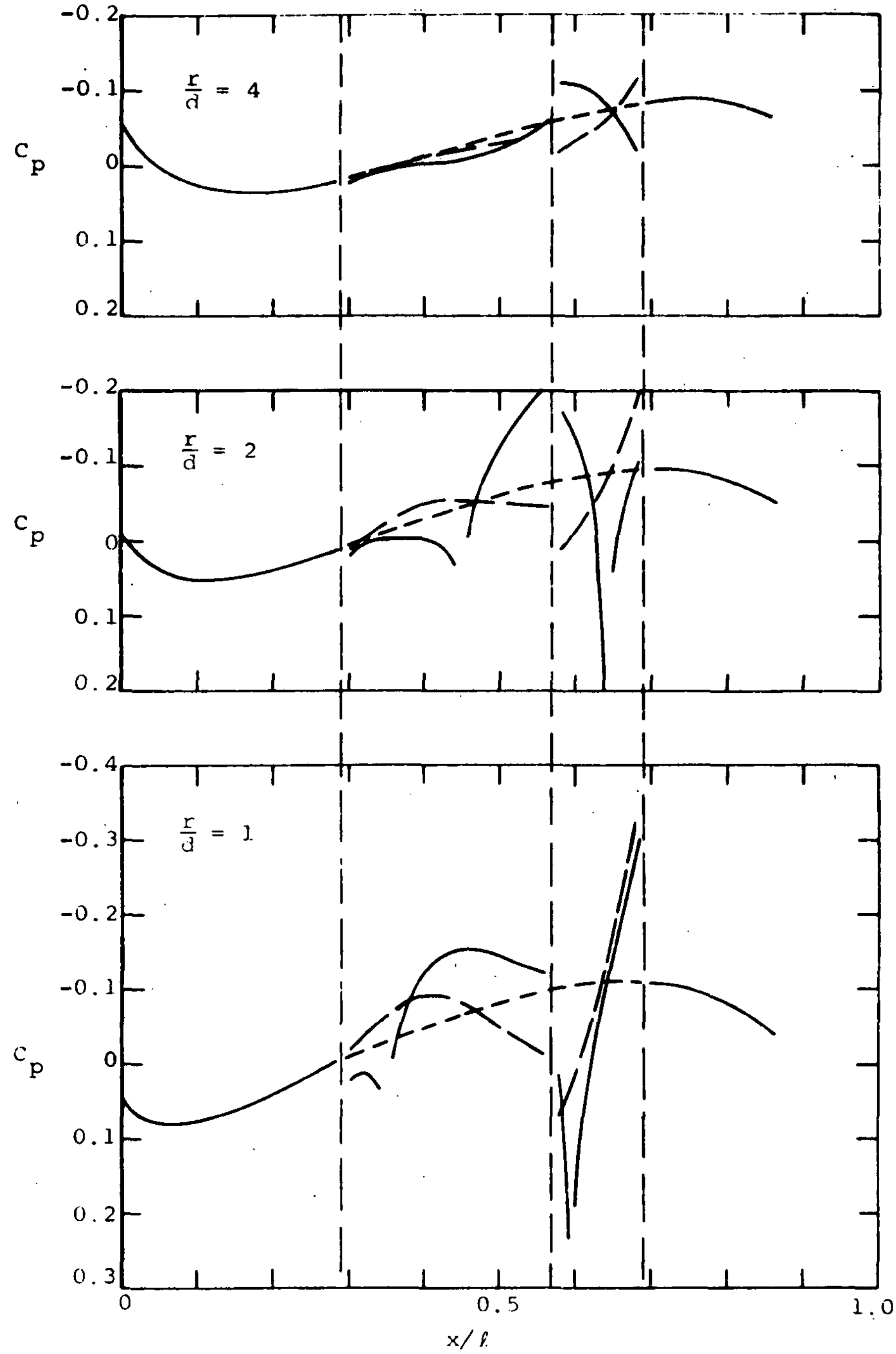


Figure 6.- Concluded.

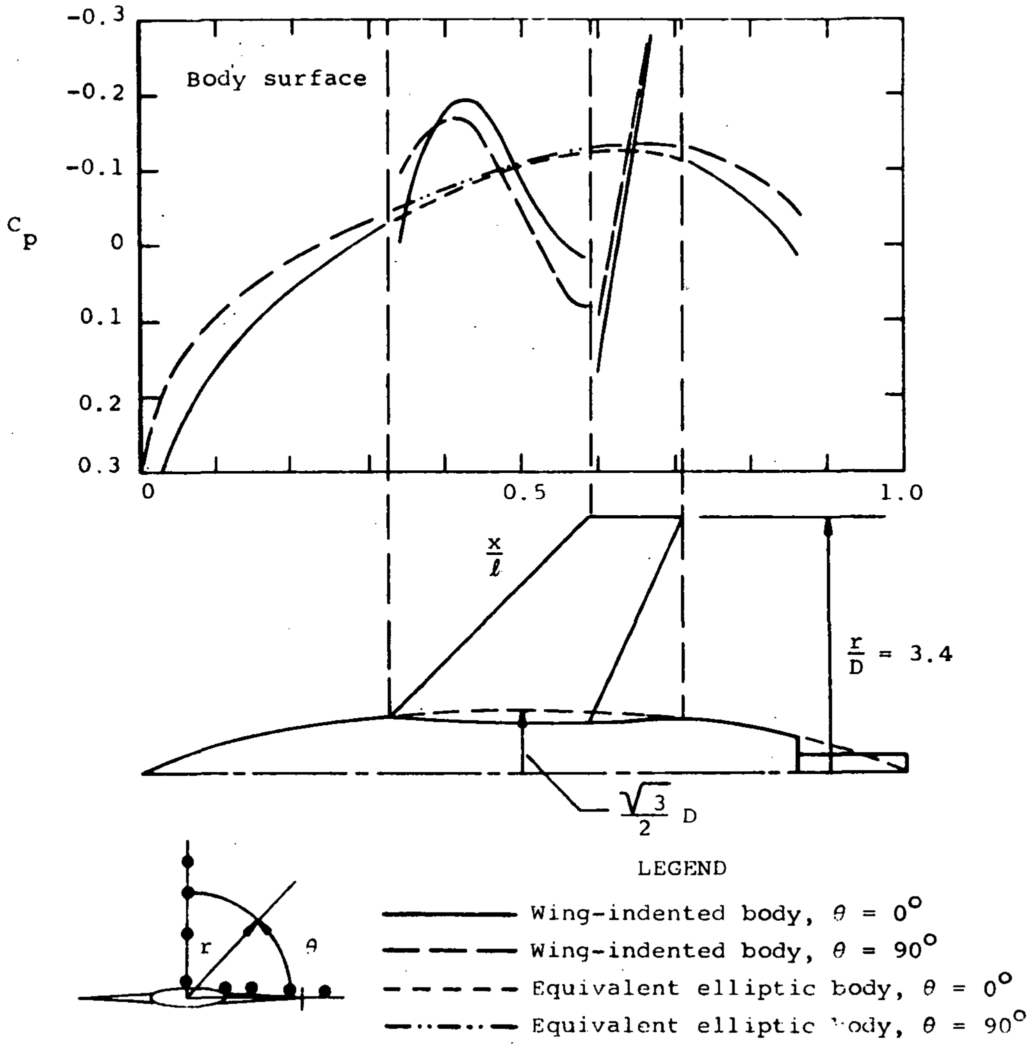


Figure 7.- Theoretical surface and flow field pressure distributions at $M_\infty = 1$ for a nonlifting parabolic-arc profile wing--indented parabolic-arc body combination; having a body of elliptical cross section with $\lambda = 3$; equivalent body thickness ratio $D/\ell = 0.1$, wing aspect ratio AR = 2.8, thickness ratio $t/c_w = 0.04$, planform taper ratio TR = 0.4, and with $X_{r\ell e/\ell} = 0.25$, $C_{R_{T/\ell}} = 0.3$, $X_{b/\ell} = 0.86$.

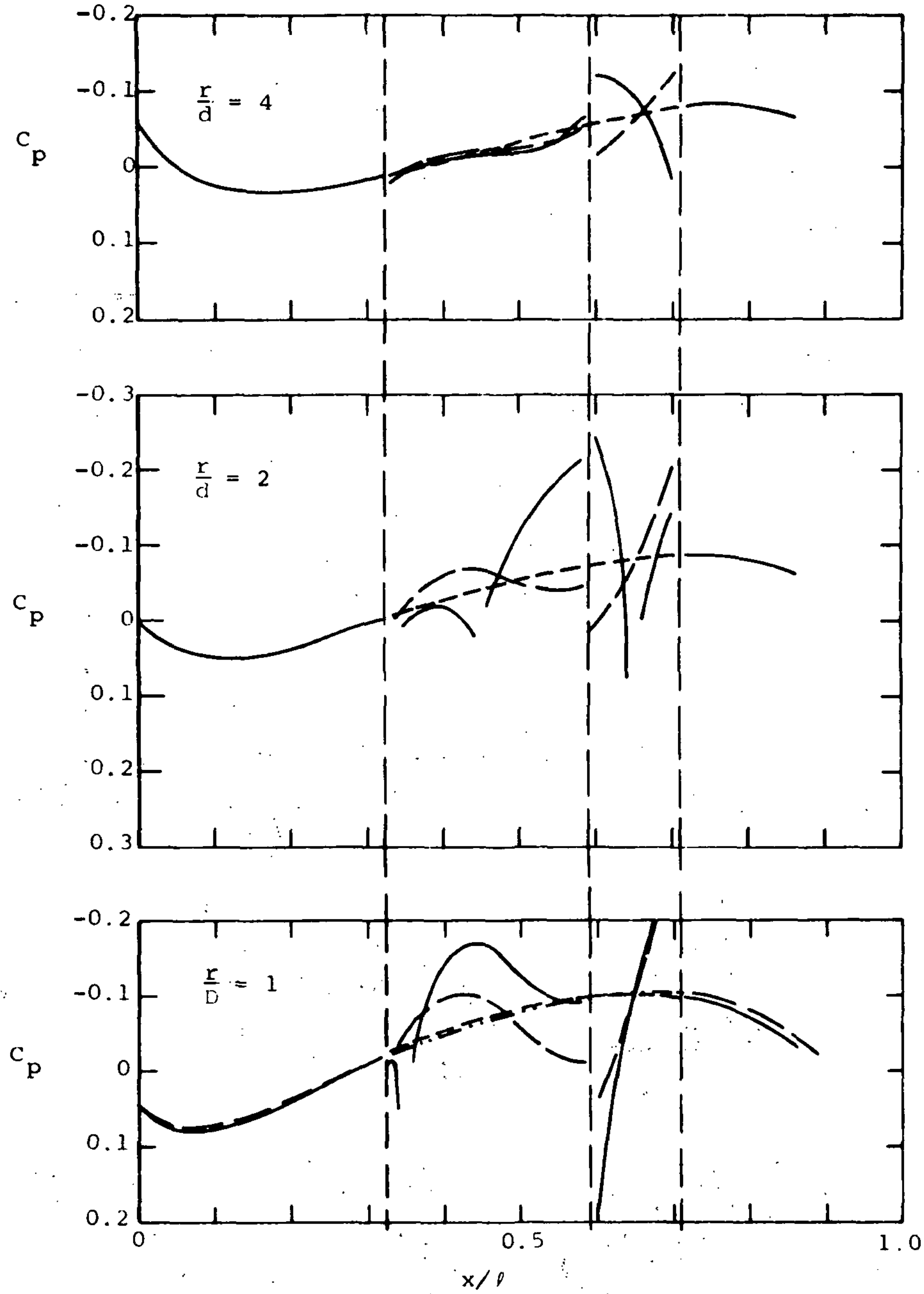


Figure 7.- Concluded.

&TRANIN AMACH=1.,MOPT=1,TAUB=.1,TAUW=.04,XMTB=.5,XMTW=.5,ANGLE=58.,
SSMAX=.25,XRLE=.25,TR=.2,CRT=.5,XLBASE=.86,XLOUTP=.05,&END

(a) Input.

CALCULATION OF SURFACE AND FLOW FIELD PRESSURE DISTRIBUTIONS FOR FLOW AT FREE STREAM MACH NUMBERS AT OR NEAR ONE, BELOW THE LOWER CRITICAL,OR ABOVE THE UPPER CRITICAL ABOUT A FINITE THICKNESS WING-INDENTED CIRCULAR BODY COMBINATION WITH THE EQUIVALENT BODY OF REVOLUTION EITHER USER-SPECIFIED OR HAVING ORDINATES R PROPORTIONAL TO X/L-(X/L)**N OR 1-X/L-(1-X/L)**N, THE WING HAVING A CONSTANT THICKNESS/CHORD RATIO, TAPER RATIO BETWEEN 0 AND 1, AND WITH ORDINATES Z PROPORTIONAL TO XBAR/C -(XBAR/C)**M OR 1-XBAR/C-(1-XBAR/C)**M BY USING THE TRANSONIC EQUIVALENCE RULE AND THE LOCAL LINEARIZATION METHOD

WING-BODY COMBINATION GEOMETRY AND FLOW FIELD CHARACTERISTICS

EQUIVALENT BODY THICKNESS RATIO =	0.10000E 00
EQUIVALENT BODY MAXIMUM THICKNESS AT X/L =	0.50000E 00
EXPONENT N FOR EQUIVALENT BODY ORDINATES =	0.20000E 01
SEB''(X) = 0 AT X/L =	0.21132E 00
WING MAX. THICKNESS AT XBAR/C =	0.50000E 00
WING THICKNESS/CHORD RATIO =	0.40000E-01
EXPONENT M FOR WING ORDINATES =	0.20000E 01
LEADING EDGE OF WING ROOT CHORD AT X/L =	0.25000E 00
TRAILING EDGE OF WING ROOT CHORD AT X/L =	0.75000E 00
PLANFORM TAPER RATIO =	0.20000E 00
LEADING EDGE PIERCES BODY AT X/L =	0.31960E 00
TRAILING EDGE PIERCES BODY AT X/L =	0.75008E 00
BODY BASE AT X/L =	0.86000E 00
LEADING EDGE SWEEP ANGLE (DEG) =	0.58000E 02
TRAILING EDGE SWEEP ANGLE (DEG) =	0.00000
LOCATION OF WINGTIP LEADING EDGE AT X/L =	0.65008E 00
LOCATION OF WINGTIP TRAILING EDGE AT X/L =	0.75008E 00
NORMALIZED MAX. SEMISPAN SSMAX/L =	0.25000E 00
ANGLE OF ATTACK ALPHA (DEG) =	0.00000
RATIO OF SPECIFIC HEATS =	0.14000E 01
FREE STREAM MACH NUMBER =	0.10000E 01

START OF INTEGRATION FROM SEB''(X) = 0 TO NOSE

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0333	0.0000	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2113	0.0333	9.0000E 01	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2003	0.0320	0.0000	2.8813E-02	3.8171E-02	3.7124E-02	3.6069E-02	3.5247E-02	3.4587E-02	3.4039E-02
0.2003	0.0320	9.0000E 01	2.8813E-02	3.8171E-02	3.7124E-02	3.6069E-02	3.5247E-02	3.4587E-02	3.4039E-02
0.1503	0.0255	0.0000	6.6692E-02	5.9471E-02	4.7472E-02	4.0071E-02	3.4755E-02	3.0613E-02	2.7221E-02
0.1503	0.0255	9.0000E 01	6.6692E-02	5.9471E-02	4.7472E-02	4.0071E-02	3.4755E-02	3.0613E-02	2.7221E-02
0.1003	0.0181	0.0000	1.1374E-01	7.5672E-02	5.0874E-02	3.6118E-02	2.5607E-02	1.7441E-02	1.0765E-02
0.1003	0.0181	9.0000E 01	1.1374E-01	7.5672E-02	5.0874E-02	3.6118E-02	2.5607E-02	1.7441E-02	1.0765E-02
0.0503	0.0096	0.0000	1.8138E-01	7.9464E-02	4.0133E-02	1.7036E-02	6.3612E-04	-1.2090E-02	-2.2490E-02
0.0503	0.0096	9.0000E 01	1.8138E-01	7.9464E-02	4.0133E-02	1.7038E-02	6.3612E-04	-1.2090E-02	-2.2490E-02
0.0043	0.0009	0.0000	3.7747E-01	4.6107E-02	-7.9117E-03	-3.9512E-02	-6.1932E-02	-7.9323E-02	-9.3533E-02
0.0043	0.0009	9.0000E 01	3.7747E-01	4.6107E-02	-7.9117E-03	-3.9512E-02	-6.1932E-02	-7.9323E-02	-9.3533E-02

START OF INTEGRATION FROM SEB''(X) = 0 TO TAIL

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0333	0.0000	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2113	0.0333	9.0000E 01	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2503	0.0375	0.0000	-3.4545E-03	1.4990E-02	2.3030E-02	2.7311E-02	3.0278E-02	3.2559E-02	3.4413E-02
0.2503	0.0375	9.0000E 01	-3.4545E-03	1.4990E-02	2.3030E-02	2.7311E-02	3.0278E-02	3.2559E-02	3.4413E-02
0.3003	0.0420	0.0000	-3.1658E-02	-8.3207E-03	6.9848E-03	1.5600E-02	2.1656E-02	2.6337E-02	3.0154E-02
0.3003	0.0420	9.0000E 01	-3.1658E-02	-8.3207E-03	6.9848E-03	1.5600E-02	2.1656E-02	2.6337E-02	3.0154E-02
0.3503	0.0453	0.0000	-4.2756E-02	-1.1470E-02	-4.4772E-03	5.0147E-03	1.2459E-02	1.8439E-02	2.3466E-02
0.3503	0.0453	9.0000E 01	-9.1482E-02	-3.9736E-02	-1.1314E-02	1.9768E-03	1.0749E-02	1.7344E-02	2.2646E-02

Figure 8.- Sample input/output for a wing-body combination having a circular body and with $TR \neq 0$, $\beta_{te} \leq 0$, $\alpha = 0$.

0.4003	0.0463	0.0000	-1.3545E-01	-9.0759E-03	-2.9387E-02	-1.6439E-02	-5.8401E-03	2.7547E-03	9.9210E-03
0.4003	0.0463	9.0000E 01	-1.3421E-01	-7.9245E-02	-4.0477E-02	-2.1309E-02	-8.5757E-03	1.0043E-03	8.7054E-03
0.4503	0.0452	0.0000	-1.8585E-01	-1.0661E-01	-4.7339E-02	-3.4247E-02	-2.2217E-02	-1.2275E-02	-3.9321E-03
0.4503	0.0452	9.0000E 01	-1.6115E-01	-1.0676E-01	-6.3579E-02	-4.1061E-02	-2.6016E-02	-1.4701E-02	-5.6153E-03
0.5003	0.0421	0.0000	-2.0276E-01	-1.5242E-01	-5.6059E-02	-4.7548E-02	-3.5858E-02	-2.5902E-02	-1.7476E-02
0.5003	0.0421	9.0000E 01	-1.6633E-01	-1.1840E-01	-7.8063E-02	-5.5503E-02	-4.0206E-02	-2.8662E-02	-1.9387E-02

START OF SUPERSONIC CALCULATION

SUPERSONIC CALCULATION STARTS AT X/L = 0.54232E 00

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.5503	0.0374	0.0000	-1.7406E-01	-1.6090E-01	-4.4047E-02	-5.5216E-02	-4.5729E-02	-3.7216E-02	-2.9953E-02
0.5503	0.0374	9.0000E 01	-1.3590E-01	-1.0833E-01	-8.0084E-02	-6.1958E-02	-4.9152E-02	-3.9346E-02	-3.1418E-02
0.6003	0.0319	0.0000	-7.6846E-02	-1.3388E-01	-1.0586E-01	-5.6845E-02	-5.0777E-02	-4.4920E-02	-3.9956E-02
0.6003	0.0319	9.0000E 01	-4.4981E-02	-6.8740E-02	-6.4622E-02	-5.6721E-02	-4.9885E-02	-4.4233E-02	-3.9450E-02
0.7003	0.0300	0.0000	1.3714E-01	-6.9617E-02	-2.0376E-01	-1.3049E-01	-9.9491E-02	-8.6103E-02	-7.8081E-02
0.7003	0.0300	9.0000E 01	1.8803E-01	2.0859E-02	-3.3187E-02	-5.4919E-02	-6.0142E-02	-6.1365E-02	-6.1009E-02
0.8003	0.0320	0.0000	-8.8124E-02	-7.8922E-02	-8.0097E-02	-8.1227E-02	-8.2102E-02	-8.2804E-02	-8.3386E-02
0.8003	0.0320	9.0000E 01	-8.8124E-02	-7.8922E-02	-8.0097E-02	-8.1227E-02	-8.2102E-02	-8.2904E-02	-8.3386E-02
0.8503	0.0255	0.0000	-3.9535E-02	-4.7040E-02	-5.9191E-02	-6.6680E-02	-7.2057E-02	-7.6247E-02	-7.9679E-02
0.8503	0.0255	9.0000E 01	-3.9535E-02	-4.7040E-02	-5.9191E-02	-6.6680E-02	-7.2057E-02	-7.6247E-02	-7.9679E-02

DRAG COEFFICIENT = 0.10440E 00

(b) Output.

Figure 8.- Concluded.

```
&TRANIN AMACH=1.,MOPT=1,TAUB=.1,TAUW=.04,XMTB=.5,XMTW=.5,ANGLE=58.,
SSMAX=.25,XRLE=.25,TR=.2,CRT=.5,XLBASE=.86,XLOUTP=.05,ALPHA=2.,&END
```

(a) Input.

```
CALCULATION OF SURFACE AND FLOW FIELD PRESSURE DISTRIBUTIONS
FOR FLOW AT FREE STREAM MACH NUMBERS AT OR NEAR ONE, BELOW
THE LOWER CRITICAL,OR ABOVE THE UPPER CRITICAL ABOUT A FINITE
THICKNESS WING-INDENTED CIRCULAR BODY COMBINATION WITH THE
EQUIVALENT BODY OF REVOLUTION EITHER USER-SPECIFIED OR HAVING
ORDINATES R PROPORTIONAL TO X/L-(X/L)**N OR 1-X/L-(1-X/L)**N,
THE WING HAVING A CONSTANT THICKNESS/CHORD RATIO, TAPER RATIO
BETWEEN 0 AND 1, AND WITH ORDINATES Z PROPORTIONAL TO XBAR/C
-(XBAR/C)**M OR 1-XBAR/C-(1-XBAR/C)**M BY USING THE TRANSONIC
EQUIVALENCE RULE AND THE LOCAL LINEARIZATION METHOD
```

```
WING-BODY COMBINATION GEOMETRY AND FLOW FIELD CHARACTERISTICS

EQUIVALENT BODY THICKNESS RATIO =                  0.10000E 00
EQUIVALENT BODY MAXIMUM THICKNESS AT X/L =         0.50000E 00
EXPONENT N FOR EQUIVALENT BODY ORDINATES =         0.20000E 01
SEB''(X) = 0 AT X/L =                              0.21132E 00
WING MAX. THICKNESS AT XBAR/C =                    0.50000E 00
WING THICKNESS/CHORD RATIO =                       0.40000E-01
EXPONENT M FOR WING ORDINATES =                    0.20000E 01
LEADING EDGE OF WING ROOT CHORD AT X/L =           0.25000E 00
TRAILING EDGE OF WING ROOT CHORD AT X/L =          0.75000E 00
PLANFORM TAPER RATIO =                             0.20000E 00
LEADING EDGE PIERCES BODY AT X/L =                 0.31960E 00
TRAILING EDGE PIERCES BODY AT X/L =                0.75008E 00
BODY BASE AT X/L =                                 0.86000E 00
LEADING EDGE SWEEP ANGLE (DEG) =                   0.58000E 02
TRAILING EDGE SWEEP ANGLE (DEG) =                  0.00000
LOCATION OF WINGTIP LEADING EDGE AT X/L =          0.65008E 00
LOCATION OF WINGTIP TRAILING EDGE AT X/L =         0.75009E 00
NORMALIZED MAX. SEMISPAN SSMAX/L =                 0.25000E 00
ANGLE OF ATTACK ALPHA (DEG) =                      0.20000E 01
RATIO OF SPECIFIC HEATS =                          0.14000E 01
FREE STREAM MACH NUMBER =                          0.10000E 01
```

START OF INTEGRATION FROM SEB''(X) = 0 TO NOSE

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0333	0.0000	1.7652E-02	3.2874E-02	3.4202E-02	3.4446E-02	3.4531E-02	3.4571E-02	3.4592E-02
0.2113	0.0333	9.0000E 01	6.4031E-03	2.5652E-02	3.0344E-02	3.1830E-02	3.2554E-02	3.2982E-02	3.3265E-02
0.2113	0.0333	-9.0000E 01	3.8649E-02	4.1178E-02	3.8331E-02	3.7182E-02	3.6576E-02	3.6202E-02	3.5949E-02
0.2003	0.0320	0.0000	2.5157E-02	3.7908E-02	3.7060E-02	3.6041E-02	3.5231E-02	3.4577E-02	3.4032E-02
0.2003	0.0320	9.0000E 01	1.3294E-02	3.0640E-02	3.3198E-02	3.3426E-02	3.3256E-02	3.2991E-02	3.2706E-02
0.2003	0.0320	-9.0000E 01	4.6768E-02	4.6177E-02	4.1173E-02	3.8768E-02	3.7269E-02	3.6204E-02	3.5385E-02
0.1503	0.0255	0.0000	6.3037E-02	5.9307E-02	4.7432E-02	4.0053E-02	3.4745E-02	3.0607E-02	2.7216E-02
0.1503	0.0255	9.0000E 01	4.8381E-02	5.2304E-02	4.3790E-02	3.7600E-02	3.2857E-02	2.9124E-02	2.5979E-02
0.1503	0.0255	-9.0000E 01	8.7441E-02	6.6945E-02	5.1233E-02	4.2577E-02	3.6634E-02	3.2115E-02	2.8472E-02
0.1003	0.0181	0.0000	1.1008E-01	7.5591E-02	5.0854E-02	3.6109E-02	2.5602E-02	1.7438E-02	1.0762E-02
0.1003	0.0181	9.0000E 01	9.2651E-02	6.9771E-02	4.7879E-02	3.4114E-02	2.4102E-02	1.6236E-02	9.7597E-03
0.1003	0.0181	-9.0000E 01	1.3728E-01	8.1728E-02	5.3907E-02	3.8139E-02	2.7122E-02	1.8653E-02	1.1774E-02
0.0503	0.0096	0.0000	1.7773E-01	7.9442E-02	4.0126E-02	1.7035E-02	6.3473E-04	-1.2091E-02	-2.2451E-02
0.0503	0.0096	9.0000E 01	1.5748E-01	7.5897E-02	3.8340E-02	1.5840E-02	-2.6243E-04	-1.2809E-02	-2.3090E-02
0.0503	0.0096	-9.0000E 01	2.0771E-01	8.3076E-02	4.1938E-02	1.8240E-02	1.5375E-03	-1.1369E-02	-2.1890E-02
0.0043	0.0009	0.0000	3.7391E-01	4.6107E-02	-7.9117E-03	-3.9512E-02	-6.1932E-02	-7.9323E-02	-9.3533E-02
0.0043	0.0009	9.0000E 01	3.5100E-01	4.5750E-02	-8.0902E-03	-3.9631E-02	-6.2022E-02	-7.9395E-02	-9.3592E-02
0.0043	0.0009	-9.0000E 01	4.0637E-01	4.6465E-02	-7.7331E-03	-3.9393E-02	-6.1843E-02	-7.9252E-02	-9.3473E-02

START OF INTEGRATION FROM SEB''(X) = 0 TO TAIL

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0333	0.0000	1.7652E-02	3.2874E-02	3.4202E-02	3.4446E-02	3.4531E-02	3.4571E-02	3.4592E-02
0.2113	0.0333	9.0000E 01	6.4031E-03	2.5652E-02	3.0344E-02	3.1830E-02	3.2554E-02	3.2982E-02	3.3265E-02
0.2113	0.0333	-9.0000E 01	3.8649E-02	4.1178E-02	3.8331E-02	3.7182E-02	3.6576E-02	3.6202E-02	3.5949E-02
0.2503	0.0375	0.0000	-7.1099E-03	1.4623E-02	2.2942E-02	2.7273E-02	3.0257E-02	3.2545E-02	3.4404E-02
0.2503	0.0375	9.0000E 01	-1.6161E-02	7.9276E-03	1.9235E-02	2.4746E-02	2.8343E-02	3.1005E-02	3.3116E-02
0.2503	0.0375	-9.0000E 01	1.1708E-02	2.2791E-02	2.6953E-02	2.9952E-02	3.2257E-02	3.4140E-02	3.5730E-02

Figure 9.- Sample input/output for a wing-body combination having a circular body and with $TR \neq 0$, $\beta_{le} \leq 0$, $\alpha \neq 0$.

0.3003	0.0420	0.0000	-3.5314E-02	-8.7891E-03	6.8748E-03	1.5552E-02	2.1629E-02	2.6320E-02	3.0142E-02
0.3003	0.0420	9.0000E 01	-4.1592E-02	-1.4545E-02	3.6267E-03	1.3319E-02	1.9932E-02	2.4951E-02	2.8997E-02
0.3003	0.0420	-9.0000E 01	-1.9288E-02	-1.3121E-03	1.0553E-02	1.7975E-02	2.3434E-02	2.7757E-02	3.1336E-02
0.3503	0.0453	0.0(UPPER)	-1.0355E-01	-1.2315E-02	-4.6357E-03	4.9451E-03	1.2420E-02	1.8414E-02	2.3389E-02
0.3503	0.0453	0.0(LOWER)	2.0472E-02	-1.2315E-02	-4.6357E-03	4.9451E-03	1.2420E-02	1.8414E-02	2.3389E-02
0.3503	0.0453	9.0000E 01	-1.1550E-01	-5.8221E-02	-2.2168E-02	-5.5204E-03	5.0506E-03	1.2755E-02	1.8608E-02
0.3503	0.0453	-9.0000E 01	-6.5025E-02	-2.0265E-02	-1.7190E-04	9.6057E-03	1.6523E-02	2.1981E-02	2.6518E-02
0.4003	0.0463	0.0(UPPER)	-1.8853E-01	-1.7800E-02	-2.9744E-02	-1.6577E-02	-5.9150E-03	2.7077E-03	9.8886E-03
0.4003	0.0463	0.0(LOWER)	-7.9930E-02	-1.7800E-02	-2.9744E-02	-1.6577E-02	-5.9150E-03	2.7077E-03	9.8886E-03
0.4003	0.0463	9.0000E 01	-1.6594E-01	-1.0647E-01	-5.8079E-02	-3.3811E-02	-1.8184E-02	-6.7745E-03	2.1792E-03
0.4003	0.0463	-9.0000E 01	-1.0004E-01	-5.0703E-02	-2.2359E-02	-8.5742E-03	1.1681E-03	8.8714E-03	1.5293E-02
0.4503	0.0452	0.0(UPPER)	-2.3608E-01	-1.8094E-01	-4.8127E-02	-3.4507E-02	-2.2351E-02	-1.2358E-02	-3.9884E-03
0.4503	0.0452	0.0(LOWER)	-1.3318E-01	-3.3905E-02	-4.8127E-02	-3.4507E-02	-2.2351E-02	-1.2358E-02	-3.9884E-03
0.4503	0.0452	9.0000E 01	-1.9573E-01	-1.3825E-01	-8.5776E-02	-5.7354E-02	-3.8717E-02	-2.5058E-02	-1.4341E-02
0.4503	0.0452	-9.0000E 01	-1.2414E-01	-7.3690E-02	-4.0675E-02	-2.4400E-02	-1.3094E-02	-4.1977E-03	3.2137E-03
0.5003	0.0421	0.0(UPPER)	-2.5131E-01	-2.1180E-01	-5.7976E-02	-4.7998E-02	-3.6079E-02	-2.6035E-02	-1.7565E-02
0.5003	0.0421	0.0(LOWER)	-1.5177E-01	-9.2198E-02	-5.7976E-02	-4.7998E-02	-3.6079E-02	-2.6035E-02	-1.7565E-02
0.5003	0.0421	9.0000E 01	-2.0258E-01	-1.5245E-01	-1.0375E-01	-7.4991E-02	-5.5641E-02	-4.1356E-02	-3.0135E-02
0.5003	0.0421	-9.0000E 01	-1.2763E-01	-8.2573E-02	-5.1443E-02	-3.5490E-02	-2.4446E-02	-1.5750E-02	-8.4837E-03

START OF SUPERSONIC CALCULATION

SUPERSONIC CALCULATION STARTS AT X/L = 0.54232E 00

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.5503	0.0374	0.0(UPPER)	-2.2116E-01	-2.1471E-01	-5.3007E-02	-5.5999E-02	-4.6073E-02	-3.7416E-02	-3.0085E-02
0.5503	0.0374	0.0(LOWER)	-1.2453E-01	-1.0559E-01	-5.3007E-02	-5.5999E-02	-4.6073E-02	-3.7416E-02	-3.0085E-02
0.5503	0.0374	9.0000E 01	-1.7361E-01	-1.4418E-01	-1.0851E-01	-8.4207E-02	-6.7067E-02	-5.4217E-02	-4.4079E-02
0.5503	0.0374	-9.0000E 01	-9.5743E-02	-7.0551E-02	-5.0508E-02	-3.9020E-02	-3.0796E-02	-2.4174E-02	-1.8540E-02
0.6003	0.0319	0.0(UPPER)	-1.2235E-01	-1.8426E-01	-2.2219E-01	-5.8233E-02	-5.1298E-02	-4.5209E-02	-4.0183E-02
0.6003	0.0319	0.0(LOWER)	-2.8904E-02	-8.1685E-02	6.0715E-04	-5.8233E-02	-5.1298E-02	-4.5209E-02	-4.0183E-02
0.6003	0.0319	9.0000E 01	-8.4287E-02	-1.0607E-01	-9.5302E-02	-8.1400E-02	-7.0081E-02	-6.1158E-02	-5.3986E-02
0.6003	0.0319	-9.0000E 01	-3.2376E-03	-2.9388E-02	-3.2612E-02	-3.1195E-02	-2.9127E-02	-2.6915E-02	-2.4707E-02
0.7003	0.0300	0.0(UPPER)	1.4021E-01	-6.7898E-02	-2.0739E-01	-1.3326E-01	-1.0027E-01	-8.6509E-02	-7.8337E-02
0.7003	0.0300	0.0(LOWER)	1.3651E-01	-6.9357E-02	-2.0202E-01	-1.3326E-01	-1.0027E-01	-8.6509E-02	-7.8337E-02
0.7003	0.0300	9.0000E 01	1.6740E-01	2.1638E-02	-3.7193E-02	-5.4065E-02	-5.9448E-02	-6.0759E-02	-6.0537E-02
0.7003	0.0300	-9.0000E 01	1.9110E-01	2.2186E-02	-3.7693E-02	-5.4774E-02	-6.0151E-02	-6.1445E-02	-6.1119E-02

DRAG COEFFICIENT = 0.13419E 00
LIFT COEFFICIENT = 0.17070E 01
PITCHING MOMENT COEFFICIENT = -0.89056E 00

(b) Output.

Figure 9.- Concluded.

```
&TRANIN AMACH=1.,MOPT=1,TAUB=.1,TAUW=.04,XMTB=.5,XMTW=.5,ANGLE=58.,
SSMAX=.25,XRLE=.25,TR=.2,CRT=.5,XLBASE=.86,XLOUTP=.05,ALPHA=2.,AL=3.,&END
```

(a) Input.

CALCULATION OF SURFACE AND FLOW FIELD PRESSURE DISTRIBUTIONS FOR FLOW AT FREE STREAM MACH NUMBERS AT OR NEAR ONE, BELOW THE LOWER CRITICAL, OR ABOVE THE UPPER CRITICAL ABOUT A FINITE THICKNESS WING-INDENTED BODY COMBINATION WITH THE BODY HAVING AN ELLIPTIC CROSS SECTION THAT MAINTAINS A CONSTANT RATIO OF MAJOR/MINOR AXES ALONG THE ENTIRE BODY LENGTH WITH THE EQUIVALENT BODY OF REVOLUTION EITHER USER-SPECIFIED OR HAVING ORDINATES R PROPORTIONAL TO X/L-(X/L)**N OR 1-X/L-(1-X/L)**N, THE WING HAVING A CONSTANT THICKNESS/CHORD RATIO, TAPER RATIO BETWEEN 0 AND 1, AND WITH ORDINATES Z PROPORTIONAL TO XBAR/C -(XBAR/C)**M OR 1-XBAR/C-(1-XBAR/C)**M BY USING THE TRANSONIC EQUIVALENCE RULE AND THE LOCAL LINEARIZATION METHOD

WING-BODY COMBINATION GEOMETRY AND FLOW FIELD CHARACTERISTICS

RATIO OF SEMIMAJOR/SEMIMINOR AXIS =	0.30000E 01
EQUIVALENT BODY THICKNESS RATIO =	0.10000E 00
EQUIVALENT BODY MAXIMUM THICKNESS AT X/L =	0.50000E 00
EXPONENT N FOR EQUIVALENT BODY ORDINATES =	0.20000E 01
SEB''(X) = 0 AT X/L =	0.21132E 00
WING MAX. THICKNESS AT XBAR/C =	0.50000E 00
WING THICKNESS/CHORD RATIO =	0.40000E-01
EXPONENT M FOR WING ORDINATES =	0.20000E 01
LEADING EDGE OF WING ROOT CHORD AT X/L =	0.25000E 00
TRAILING EDGE OF WING ROOT CHORD AT X/L =	0.75000E 00
PLANFORM TAPER RATIO =	0.20000E 00
LEADING EDGE PIERCES BODY AT X/L =	0.38070E 00
TRAILING EDGE PIERCES BODY AT X/L =	0.75008E 00
BODY BASE AT X/L =	0.86000E 00
LEADING EDGE SWEEP ANGLE (DEG) =	0.58000E 02
TRAILING EDGE SWEEP ANGLE (DEG) =	0.00000
LOCATION OF WINGTIP LEADING EDGE AT X/L =	0.65008E 00
LOCATION OF WINGTIP TRAILING EDGE AT X/L =	0.75008E 00
NORMALIZED MAX. SEMISPAN SSMAX/L =	0.25000E 00
ANGLE OF ATTACK ALPHA (DEG) =	0.20000E 01
RATIO OF SPECIFIC HEATS =	0.14000E 01
FREE STREAM MACH NUMBER =	0.10000E 01

START OF INTEGRATION FROM SEB''(X) = 0 TO NOSE

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0577	0.0000	2.9697E-02	3.6854E-02	3.5139E-02	3.4458E-02	3.4762E-02	3.4718E-02	3.4695E-02
0.2113	0.0192	9.0000E 01	-4.9801E-03	1.9287E-02	2.7071E-02	2.9726E-02	3.1020E-02	3.1780E-02	3.2278E-02
0.2113	0.0192	-9.0000E 01	3.2254E-02	4.2027E-02	3.9859E-02	3.8496E-02	3.7659E-02	3.7114E-02	3.6733E-02
0.2003	0.0555	0.0000	3.6827E-02	4.2121E-02	3.8053E-02	3.6478E-02	3.5476E-02	3.4733E-02	3.4140E-02
0.2003	0.0185	9.0000E 01	6.7900E-04	2.3926E-02	2.9855E-02	3.1294E-02	3.1709E-02	3.1781E-02	3.1715E-02
0.2003	0.0185	-9.0000E 01	3.5332E-02	4.6873E-02	4.2658E-02	4.0060E-02	3.8338E-02	3.7105E-02	3.6162E-02
0.1503	0.0442	0.0000	8.4855E-02	6.3604E-02	4.8461E-02	4.0507E-02	3.5000E-02	3.0769E-02	2.7329E-02
0.1503	0.0147	9.0000E 01	2.5626E-02	4.4900E-02	4.0471E-02	3.5539E-02	3.1418E-02	2.7975E-02	2.5041E-02
0.1503	0.0147	-9.0000E 01	7.4711E-02	6.7389E-02	5.2603E-02	4.2751E-02	3.7610E-02	3.2941E-02	2.9185E-02
0.1003	0.0313	0.0000	1.4130E-01	7.8617E-02	5.1593E-02	3.6437E-02	2.5786E-02	1.7556E-02	1.0844E-02
0.1003	0.0104	9.0000E 01	6.6863E-02	6.3481E-02	4.5206E-02	3.2466E-02	2.2920E-02	1.5317E-02	9.0091E-03
0.1003	0.0104	-9.0000E 01	1.2841E-01	8.2569E-02	5.5140E-02	3.9140E-02	2.7939E-02	1.9338E-02	1.2362E-02
0.0503	0.0166	0.0000	2.1961E-01	8.0565E-02	4.0407E-02	1.7159E-02	7.0433E-04	-1.2047E-02	-2.2460E-02
0.0503	0.0055	9.0000E 01	1.2382E-01	7.2489E-02	3.6874E-02	1.4920E-02	-9.3070E-04	-1.3333E-02	-2.3520E-02
0.0503	0.0055	-9.0000E 01	1.8182E-01	8.4504E-02	4.2851E-02	1.8913E-02	2.0667E-03	-1.0935E-02	-2.1521E-02
0.0043	0.0015	0.0000	4.2062E-01	4.6119E-02	-7.9069E-03	-3.4510E-02	-6.1932E-02	-7.9323E-02	-9.3532E-02
0.0043	0.0005	9.0000E 01	3.0929E-01	4.5501E-02	-8.2121E-03	-3.4711E-02	-6.2082E-02	-7.9443E-02	-9.3632E-02
0.0043	0.0005	-9.0000E 01	3.7323E-01	4.6691E-02	-7.6170E-03	-3.4314E-02	-6.1784E-02	-7.9204E-02	-9.3434E-02

START OF INTEGRATION FROM SEB''(X) = 0 TO TAIL

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0577	0.0000	2.9697E-02	3.6854E-02	3.5139E-02	3.4858E-02	3.4762E-02	3.4718E-02	3.4695E-02
0.2113	0.0192	9.0000E 01	-4.9801E-03	1.9287E-02	2.7071E-02	2.9726E-02	3.1020E-02	3.1780E-02	3.2278E-02
0.2113	0.0192	-9.0000E 01	3.2254E-02	4.2027E-02	3.9859E-02	3.8496E-02	3.7659E-02	3.7114E-02	3.6733E-02
0.2503	0.0650	0.0000	-3.3385E-04	1.7161E-02	2.3564E-02	2.7547E-02	3.0411E-02	3.2644E-02	3.4472E-02
0.2503	0.0217	9.0000E 01	-2.3538E-02	3.0961E-03	1.6355E-02	2.2843E-02	2.6929E-02	2.9884E-02	3.2190E-02
0.2503	0.0217	-9.0000E 01	8.6659E-03	2.4411E-02	2.8741E-02	3.1337E-02	3.3373E-02	3.5068E-02	3.6522E-02

Figure 10.- Sample input/output for a wing-body combination having an elliptical body and with $TR \neq 0$, $\beta_{te} \leq 0$, $\alpha \neq 0$.

0.3003	0.0728	0.0000	-3.4176E-02	-9.3541E-03	6.8810E-03	1.5561E-02	2.1635E-02	2.4324E-02	3.0145E-02
0.3003	0.0243	9.0000E 01	-4.4569E-02	-1.6684E-02	1.6510E-03	1.1865E-02	1.8804E-02	2.4036E-02	2.8227E-02
0.3003	0.0243	-9.0000E 01	-1.8815E-02	1.6199E-03	1.2553E-02	1.9422E-02	2.4553E-02	2.8666E-02	3.2100E-02
0.3503	0.0788	0.0000	-6.5433E-02	-3.5815E-02	-1.0685E-02	1.7634E-03	1.0372E-02	1.6986E-02	2.2366E-02
0.3503	0.0263	9.0000E 01	-6.3149E-02	-3.4999E-02	-1.3465E-02	-5.9458E-04	8.4323E-03	1.5357E-02	2.0967E-02
0.3503	0.0263	-9.0000E 01	-4.3844E-02	-2.0633E-02	-4.7157E-03	5.5071E-03	1.3085E-02	1.9109E-02	2.4107E-02
0.4003	0.0830	0.0(UPPER)	-1.3538E-01	-7.7849E-03	-1.9989E-02	-8.7076E-03	2.3704E-04	7.4149E-03	1.3370E-02
0.4003	0.0830	0.0(LOWER)	4.4171E-02	-7.7849E-03	-1.9989E-02	-8.7076E-03	2.3704E-04	7.4149E-03	1.3370E-02
0.4003	0.0277	9.0000E 01	-1.3518E-01	-7.8563E-02	-4.1570E-02	-2.2118E-02	-9.4300E-03	-1.1666E-04	7.2124E-03
0.4003	0.0277	-9.0000E 01	-7.8030E-02	-3.2890E-02	-1.2268E-02	-1.3539E-03	6.5047E-03	1.2771E-02	1.8018E-02
0.4503	0.0834	0.0(UPPER)	-2.2484E-01	-1.9656E-01	-5.3038E-02	-3.7551E-02	-2.4492E-02	-1.3921E-02	-5.1265E-03
0.4503	0.0836	0.0(LOWER)	-9.5429E-02	-4.7413E-02	-5.3038E-02	-3.7551E-02	-2.4492E-02	-1.3921E-02	-5.1265E-03
0.4503	0.0279	9.0000E 01	-1.5857E-01	-1.3547E-01	-8.6272E-02	-5.8245E-02	-3.9544E-02	-2.5715E-02	-1.4804E-02
0.4503	0.0279	-9.0000E 01	-1.2538E-01	-7.2592E-02	-4.2515E-02	-2.6271E-02	-1.4680E-02	-5.4707E-03	2.2317E-03
0.5003	0.0801	0.0(UPPER)	-2.6787E-01	-2.3975E-01	-7.2912E-02	-5.8920E-02	-4.4316E-02	-3.2241E-02	-2.2133E-02
0.5003	0.0801	0.0(LOWER)	-1.5022E-01	-1.1636E-01	-7.2912E-02	-5.8920E-02	-4.4316E-02	-3.2241E-02	-2.2133E-02
0.5003	0.0267	9.0000E 01	-2.3275E-01	-1.6946E-01	-1.1677E-01	-8.4960E-02	-6.3288E-02	-4.7152E-02	-3.4355E-02
0.5003	0.0267	-9.0000E 01	-1.5563E-01	-1.0022E-01	-6.5108E-02	-4.5939E-02	-3.2469E-02	-2.1853E-02	-1.3003E-02

START OF SUPERSONIC CALCULATION

SUPERSONIC CALCULATION STARTS AT X/L = 0.54232E 00

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.5503	0.0729	0.0(UPPER)	-2.4483E-01	-2.3568E-01	-6.7498E-02	-6.7246E-02	-5.4718E-02	-4.3983E-02	-3.4932E-02
0.5503	0.0729	0.0(LOWER)	-1.3406E-01	-1.2342E-01	-6.7498E-02	-6.7246E-02	-5.4718E-02	-4.3983E-02	-3.4932E-02
0.5503	0.0243	9.0000E 01	-2.1879E-01	-1.6978E-01	-1.2470E-01	-9.6074E-02	-7.6019E-02	-6.0956E-02	-4.9027E-02
0.5503	0.0243	-9.0000E 01	-1.3938E-01	-9.5664E-02	-6.7159E-02	-5.1242E-02	-4.0027E-02	-3.1143E-02	-2.3682E-02
0.6003	0.0631	0.0(UPPER)	-1.3192E-01	-1.7966E-01	-2.2356E-01	-5.9157E-02	-5.2205E-02	-4.5961E-02	-4.0752E-02
0.6003	0.0631	0.0(LOWER)	-2.8191E-02	-7.5253E-02	-2.4762E-04	-5.9157E-02	-5.2205E-02	-4.5961E-02	-4.0752E-02
0.6003	0.0210	9.0000E 01	-1.1992E-01	-1.1682E-01	-9.9784E-02	-8.4131E-02	-7.1967E-02	-6.2514E-02	-5.4961E-02
0.6003	0.0210	-9.0000E 01	-3.8163E-02	-4.0630E-02	-3.7462E-02	-3.4207E-02	-3.1237E-02	-2.8456E-02	-2.5839E-02
0.7003	0.0550	0.0(UPPER)	1.3230E-01	-3.9002E-02	-1.9151E-01	-1.2489E-01	-9.4236E-02	-8.2051E-02	-7.5077E-02
0.7003	0.0550	0.0(LOWER)	1.2689E-01	-4.0953E-02	-1.8671E-01	-1.2489E-01	-9.4236E-02	-8.2051E-02	-7.5077E-02
0.7003	0.0183	9.0000E 01	1.5910E-01	2.1566E-02	-3.1702E-02	-4.8917E-02	-5.5222E-02	-5.7500E-02	-5.8083E-02
0.7003	0.0183	-9.0000E 01	1.6091E-01	2.1883E-02	-3.2230E-02	-4.9625E-02	-5.5919E-02	-5.8138E-02	-5.8657E-02

DRAG COEFFICIENT = C.12462E 00
LIFT COEFFICIENT = 0.17073E 01
PITCHING MOMENT COEFFICIENT = -0.8694OE 00

(b) Output.

Figure 10.- Concluded.

```
&TRANIN AMACH=1.,MOPT=1,TAUB=.1,TAUW=.04,XMTB=.5,XMTW=.5,ANGLE=45.,
SSMAX=.3215,XRLE=.25,TR=.4,CRT=.4,XLBASE=.86,XLOUTP=.05,&END
```

(a) Input.

```
CALCULATION OF SURFACE AND FLOW FIELD PRESSURE DISTRIBUTIONS
FOR FLOW AT FREE STREAM MACH NUMBERS AT OR NEAR ONE, BELOW
THE LOWER CRITICAL,OR ABOVE THE UPPER CRITICAL ABOUT A FINITE
THICKNESS WING-INDENTED CIRCULAR BODY COMBINATION WITH THE
EQUIVALENT BODY OF REVOLUTION EITHER USER-SPECIFIED OR HAVING
ORDINATES R PROPORTIONAL TO X/L-(X/L)**N OR 1-X/L-(1-X/L)**N,
THE WING HAVING A CONSTANT THICKNESS/CHORD RATIO, TAPER RATIO
BETWEEN 0 AND 1, AND WITH ORDINATES Z PROPORTIONAL TO XBAR/C
-(XBAR/C)**M OR 1-XBAR/C-(1-XBAR/C)**M BY USING THE TRANSONIC
EQUIVALENCE RULE AND THE LOCAL LINEARIZATION METHOD
```

WING-BODY COMBINATION GEOMETRY AND FLOW FIELD CHARACTERISTICS

```
EQUIVALENT BODY THICKNESS RATIO =                  0.10000E 00
EQUIVALENT BODY MAXIMUM THICKNESS AT X/L =         0.50000E 00
EXPONENT N FOR EQUIVALENT BODY ORDINATES =         0.20000E 01
SEB''(X) = 0 AT X/L =                              0.21132E 00
WING MAX. THICKNESS AT XBAR/C =                    0.50000E 00
WING THICKNESS/CHORD RATIO =                       0.40000E-01
EXPONENT M FOR WING ORDINATES =                    0.20000E 01
LEADING EDGE OF WING ROOT CHORD AT X/L =           0.25000E 00
TRAILING EDGE OF WING ROOT CHORD AT X/L =          0.55000E 00
PLANFORM TAPER RATIO =                             0.40000E 00
LEADING EDGE PIERCES BODY AT X/L =                 0.29129E 00
TRAILING EDGE PIERCES BODY AT X/L =                0.57156E 00
BODY BASE AT X/L =                                 0.86000E 00
LEADING EDGE SWEEP ANGLE (DEG) =                   0.45000E 02
TRAILING EDGE SWEEP ANGLE (DEG) =                  0.23755E 02
LOCATION OF WINGTIP LEADING EDGE AT X/L =          0.57150E 00
LOCATION OF WINGTIP TRAILING EDGE AT X/L =         0.69150E 00
NORMALIZED MAX. SEMISPAN SSMAX/L =                 0.32150E 00
ANGLE OF ATTACK ALPHA (DEG) =                      0.00000
RATIO OF SPECIFIC HEATS =                          0.14000E 01
FREE STREAM MACH NUMBER =                          0.10000E 01
```

START OF INTEGRATION FROM SEB''(X) = 0 TO NOSE

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0333	0.0000	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2113	0.0333	9.0000E 01	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2003	0.0320	0.0000	2.6813E-02	3.8171E-02	3.7124E-02	3.6069E-02	3.5247E-02	3.4587E-02	3.4039E-02
0.2003	0.0320	9.0000E 01	2.6813E-02	3.8171E-02	3.7124E-02	3.6069E-02	3.5247E-02	3.4587E-02	3.4039E-02
0.1503	0.0255	0.0000	6.6692E-02	5.9471E-02	4.7472E-02	4.0071E-02	3.4755E-02	3.0613E-02	2.7221E-02
0.1503	0.0255	9.0000E 01	6.6692E-02	5.9471E-02	4.7472E-02	4.0071E-02	3.4755E-02	3.0613E-02	2.7221E-02
0.1003	0.0181	0.0000	1.1374E-01	7.5672E-02	5.0874E-02	3.6118E-02	2.5607E-02	1.7441E-02	1.0765E-02
0.1003	0.0181	9.0000E 01	1.1374E-01	7.5672E-02	5.0874E-02	3.6118E-02	2.5607E-02	1.7441E-02	1.0765E-02
0.0503	0.0096	0.0000	1.8138E-01	7.9464E-02	4.0133E-02	1.7038E-02	6.3612E-04	-1.2090E-02	-2.2490E-02
0.0503	0.0096	9.0000E 01	1.8138E-01	7.9464E-02	4.0133E-02	1.7038E-02	6.3612E-04	-1.2090E-02	-2.2490E-02
0.0043	0.0009	0.0000	3.7747E-01	4.6107E-02	-7.9117E-03	-3.9512E-02	-6.1932E-02	-7.9323E-02	-9.3533E-02
0.0043	0.0009	9.0000E 01	3.7747E-01	4.6107E-02	-7.9117E-03	-3.9512E-02	-6.1932E-02	-7.9323E-02	-9.3533E-02

START OF INTEGRATION FROM SEB''(X) = 0 TO TAIL

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0333	0.0000	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2113	0.0333	9.0000E 01	2.1307E-02	3.3159E-02	3.4270E-02	3.4476E-02	3.4548E-02	3.4582E-02	3.4600E-02
0.2503	0.0375	0.0000	-3.4545E-03	1.4990E-02	2.3030E-02	2.7311E-02	3.0278E-02	3.2559E-02	3.4413E-02
0.2503	0.0375	9.0000E 01	-3.4545E-03	1.4990E-02	2.3030E-02	2.7311E-02	3.0278E-02	3.2559E-02	3.4413E-02
0.3003	0.0420	0.0000	6.4617E-02	2.2911E-02	1.9734E-02	2.3766E-02	2.7481E-02	3.0616E-02	3.3278E-02
0.3003	0.0420	9.0000E 01	-7.8405E-02	-1.1963E-02	1.1116E-02	1.9936E-02	2.5327E-02	2.9237E-02	3.2320E-02
0.3503	0.0440	0.0000	-1.4081E-01	1.0000E 06	-4.9468E-03	1.9090E-03	9.4194E-03	1.5928E-02	2.1505E-02
0.3503	0.0440	9.0000E 01	-1.3218E-01	-6.5572E-02	-2.4868E-02	-6.7934E-03	4.5358E-03	1.2803E-02	1.9335E-02
0.4003	0.0434	0.0000	-1.6553E-01	-1.2126E-01	-8.0282E-03	-9.6651E-03	-2.2625E-03	5.0103E-03	1.1470E-02
0.4003	0.0434	9.0000E 01	-1.4164E-01	-9.0492E-02	-4.6757E-02	-2.4610E-02	-1.0504E-02	-2.3866E-04	7.8318E-03

Figure 11.- Sample input/output for a wing-body combination having a circular body and with $TR \neq 0$, $\beta_{te} > 0$, $\alpha = 0$.

0.4503	0.0413	0.0000	-1.5553E-01	-1.5350E-01	1.0000E 06	-1.3361E-02	-5.5258E-03	-3.4178E-03	2.4071E-03
0.4503	0.0413	9.0000E 01	-5.9678E-02	-8.2432E-02	-5.2027E-02	-3.2544E-02	-1.9330E-02	-9.5426E-03	-1.8102E-03
0.5003	0.0391	0.0000	-7.4455E-02	-1.4372E-01	-1.2943E-01	-1.6070E-02	-1.9618E-02	-1.4648E-02	-5.4307E-03
0.5003	0.0391	9.0000E 01	-1.5436E-02	-5.1925E-02	-4.5891E-02	-3.5565E-02	-2.5932E-02	-1.8148E-02	-1.1739E-02

START OF SUPERSONIC CALCULATION

SUPERSONIC CALCULATION STARTS AT X/L = 0.54232E 00

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.5503	0.0390	0.0000	5.1569E-03	-1.2736E-01	-1.9753E-01	1.0000E 06	-4.5947E-02	-3.8580E-02	-3.1144E-02
0.5503	0.0390	9.0000E 01	7.3914E-02	-1.9654E-02	-4.5427E-02	-4.4275E-02	-1.8527E-02	-3.2306E-02	-2.6452E-02
0.6003	0.0418	0.0000	5.6515E-02	1.8414E-01	-1.3863E-01	-2.1714E-01	-1.0602E-01	-7.6655E-02	-6.1058E-02
0.6003	0.0418	9.0000E 01	1.8844E-02	1.5786E-02	-1.2692E-02	-2.6592E-02	-3.3319E-02	-3.1867E-02	-3.0599E-02
0.6503	0.0439	0.0000	-2.6776E-01	-1.5106E-01	3.9488E-02	-1.2196E-01	-6.9199E-02	-6.0871E-02	-5.5779E-02
0.6503	0.0439	9.0000E 01	-2.7797E-01	-1.7927E-01	-1.1552E-01	-8.9898E-02	-7.6099E-02	-6.7084E-02	-6.0440E-02
0.7003	0.0420	0.0000	-1.3105E-01	-1.0776E-01	-9.2537E-02	-8.3971E-02	-7.7950E-02	-7.3257E-02	-6.9502E-02
0.7003	0.0420	9.0000E 01	-1.3105E-01	-1.0776E-01	-9.2537E-02	-8.3971E-02	-7.7950E-02	-7.3257E-02	-6.9502E-02
0.7503	0.0375	0.0000	-1.1609E-01	-9.7726E-02	-8.9791E-02	-8.5572E-02	-8.2649E-02	-8.0463E-02	-7.8576E-02
0.7503	0.0375	9.0000E 01	-1.1609E-01	-9.7726E-02	-8.9791E-02	-8.5572E-02	-8.2649E-02	-8.0463E-02	-7.8576E-02
0.8003	0.0320	0.0000	-8.8124E-02	-7.8922E-02	-8.0097E-02	-8.1727E-02	-8.2102E-02	-8.2804E-02	-8.3386E-02
0.8003	0.0320	9.0000E 01	-8.8124E-02	-7.8922E-02	-8.0097E-02	-8.1727E-02	-8.2102E-02	-8.2804E-02	-8.3386E-02
0.8503	0.0255	0.0000	-3.9535E-02	-4.7040E-02	-5.9191E-02	-6.6680E-02	-7.2057E-02	-7.6247E-02	-7.9679E-02
0.8503	0.0255	9.0000E 01	-3.9535E-02	-4.7040E-02	-5.9191E-02	-6.6680E-02	-7.2057E-02	-7.6247E-02	-7.9679E-02

DRAG COEFFICIENT = 0.10440E 00

(b) Output.

Figure 11.- Concluded.

```
&TRANIN AMACH=1.,MOPT=1,TAUB=.1,TAUW=.04,XMTB=.5,XMTW=.5,ANGLE=45.,
SSMAX=.338,XRLE=.25,TR=.4,CRT=.4,XLBASE=.86,XLOUTP=.05,AL=3.,&END
```

(a) Input.

```
CALCULATION OF SURFACE AND FLOW FIELD PRESSURE DISTRIBUTIONS
FOR FLOW AT FREE STREAM MACH NUMBERS AT OR NEAR ONE, BELOW
THE LOWER CRITICAL, OR ABOVE THE UPPER CRITICAL ABOUT A FINITE
THICKNESS WING-INCENTEC BODY COMBINATION WITH THE BODY HAVING
AN ELLIPTIC CROSS SECTION THAT MAINTAINS A CONSTANT RATIO OF
MAJOR/MINOR AXES ALONG THE ENTIRE BODY LENGTH WITH THE EQUI-
VALENT BODY OF REVOLUTION EITHER USER-SPECIFIED OR HAVING OR-
DINATES R PROPORTIONAL TO X/L-(X/L)**N OR 1-X/L-(1-X/L)**N,
THE WING HAVING A CONSTANT THICKNESS/CHORD RATIO, TAPER RATIO
BETWEEN 0 AND 1, AND WITH ORDINATES Z PROPORTIONAL TO XBAR/C
-(XBAR/C)**M OR 1-XBAR/C-(1-XBAR/C)**M BY USING THE TRANSONIC
EQUIVALENCE RULE AND THE LOCAL LINEARIZATION METHOD
```

```
WING-BODY COMBINATION GEOMETRY AND FLOW FIELD CHARACTERISTICS

RATIO OF SEMIMAJOR/SEMIMINOR AXIS =           C.30000E 01
EQUIVALENT BODY THICKNESS RATIO =             C.10000E 00
EQUIVALENT BODY MAXIMUM THICKNESS AT X/L =    C.50000E 00
EXPONENT N FOR EQUIVALENT BODY ORDINATES =    C.20000E 01
SEB**(X) = 0 AT X/L =                         C.21132E 00
WING MAX. THICKNESS AT XBAR/C =               C.50000E 00
WING THICKNESS/CHORD RATIO =                  C.40000E-01
EXPONENT M FOR WING ORDINATES =               C.20000E 01
LEADING EDGE OF WING ROOT CHORD AT X/L =      C.25000E 00
TRAILING EDGE OF WING ROOT CHORD AT X/L =     C.55000E 00
PLANFORM TAPER RATIO =                        C.40000E 00
LEADING EDGE PIERCES BODY AT X/L =            C.32613E 00
TRAILING EDGE PIERCES BODY AT X/L =           C.53919E 00
BODY BASE AT X/L =                            C.86000E 00
LEADING EDGE SWEEP ANGLE (DEG) =              C.45000E 02
TRAILING EDGE SWEEP ANGLE (DEG) =             C.25054E 02
LOCATION OF WINGTIP LEADING EDGE AT X/L =     C.58800E 00
LOCATION OF WINGTIP TRAILING EDGE AT X/L =    C.70800E 00
NORMALIZED MAX. SEMISPAN SSMAX/L =            C.33800E 00
ANGLE OF ATTACK ALPHA (DEG) =                 C.C0000
RATIO OF SPECIFIC HEATS =                     C.14000E 01
FREE STREAM MACH NUMBER =                     0.10000E 01

START OF INTEGRATION FROM SEB**(X) = 0 TO NOSE
```

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0577	0.0000	4.7974E-02	3.7658E-02	3.5267E-02	3.4921E-02	3.4797E-02	3.4740E-02	3.4710E-02
0.2113	0.0192	9.0000E 01	1.2418E-02	3.0253E-02	3.3360E-02	3.4053E-02	3.4306E-02	3.4426E-02	3.4491E-02
0.2003	0.0555	0.0000	5.7104E-02	4.2837E-02	3.8188E-02	3.6535E-02	3.5508E-02	3.4754E-02	3.4154E-02
0.2003	0.0185	9.0000E 01	1.8787E-02	3.5020E-02	3.6160E-02	3.5623E-02	3.4993E-02	3.4423E-02	3.3929E-02
0.1503	0.0442	0.0000	1.0313E-01	6.3999E-02	4.8544E-02	4.0543E-02	3.5020E-02	3.0782E-02	2.7338E-02
0.1503	0.0147	9.0000E 01	5.0961E-02	5.5879E-02	4.6461E-02	3.9611E-02	3.4495E-02	3.0446E-02	2.7104E-02
0.1003	0.0313	0.0000	1.5958E-01	7.8793E-02	5.1634E-02	3.6454E-02	2.5796E-02	1.7562E-02	1.0848E-02
0.1003	0.0104	9.0000E 01	9.1420E-02	7.2880E-02	5.0134E-02	3.5786E-02	2.5419E-02	1.7321E-02	1.0681E-02
0.0503	0.0166	0.0000	2.3788E-01	8.0611E-02	4.0418E-02	1.7164E-02	7.0711E-04	-1.2045E-02	-2.2459E-02
0.0503	0.0055	9.0000E 01	1.5160E-01	7.8353E-02	3.9851E-02	1.6912E-02	5.6524E-04	-1.2136E-02	-2.2522E-02
0.0043	0.0015	0.0000	4.4490E-01	4.6119E-02	-7.9088E-03	-3.9510E-02	-6.1932E-02	-7.9323E-02	-9.3532E-02
0.0043	0.0005	9.0000E 01	3.4004E-01	4.6096E-02	-7.9147E-03	-3.9513E-02	-6.1933E-02	-7.9324E-02	-9.3533E-02

```
START OF INTEGRATION FROM SEB**(X) = 0 TO TAIL
```

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= 6.00)
0.2113	0.0577	0.0000	4.7974E-02	3.7658E-02	3.5267E-02	3.4921E-02	3.4797E-02	3.4740E-02	3.4710E-02
0.2113	0.0192	9.0000E 01	1.2418E-02	3.0253E-02	3.3360E-02	3.4053E-02	3.4306E-02	3.4426E-02	3.4451E-02
0.2503	0.0650	0.0000	1.7943E-02	1.8326E-02	2.3756E-02	2.7627E-02	3.0455E-02	3.2671E-02	3.4452E-02
0.2503	0.0217	9.0000E 01	-8.6543E-03	1.3269E-02	2.2413E-02	2.7017E-02	3.0109E-02	3.2449E-02	3.4337E-02
0.3003	0.0728	0.0000	-1.5819E-02	-7.5985E-03	7.1287E-03	1.5062E-02	2.1651E-02	2.6359E-02	3.0170E-02
0.3003	0.0243	9.0000E 01	-3.2910E-02	-6.0892E-03	6.9124E-03	1.5553E-02	2.1627E-02	2.6317E-02	3.0140E-02
0.3503	0.0784	0.0000	-6.1022E-02	1.0000E 06	1.3079E-03	7.1848E-03	1.3604E-02	1.9148E-02	2.3851E-02
0.3503	0.0261	9.0000E 01	-1.1392E-01	-4.7325E-02	-1.4910E-02	1.2753E-04	9.6454E-03	1.6616E-02	2.2133E-02

Figure 12.- Sample input/output for a wing-body combination having an elliptical body and with $TR \neq 0$, $\beta_{te} > 0$, $\alpha = 0$.

0.4003	0.0793	0.0000	-1.8302E-01	-1.4084E-01	-1.7435E-02	-1.6409E-02	-7.2094E-03	1.3027E-03	8.7467E-03
0.4003	0.0264	9.0000E 01	-1.6235E-01	-9.8391E-02	-5.3754E-02	-3.0199E-02	-1.4855E-02	-3.5634E-03	5.3760E-03
0.4503	0.0762	0.0000	-1.7884E-01	-1.6717E-01	1.0000E 06	-2.0253E-02	[illegible]	-7.5245E-03	[illegible]
0.4503	0.0256	9.0000E 01	-1.3938E-01	-9.5965E-02	-6.2977E-02	-4.0469E-02	-2.5278E-02	-1.4013E-02	-5.0541E-03
0.5003	0.0712	0.0000	[illegible]	-1.3082E-01	[illegible]	-1.1252E-02	-1.6524E-02	-1.2476E-02	-7.6672E-03
0.5003	0.0237	9.0000E 01	-4.5507E-02	-5.8198E-02	-4.7049E-02	-3.4734E-02	-2.4995E-02	-1.7335E-02	-1.1127E-02

START OF SUPERSONIC CALCULATION

SUPERSONIC CALCULATION STARTS AT X/L = 0.54232E 00

X/L	RBODY/L	THETA(DEG)	CP(BODY)	CP(R/D= 1.00)	CP(R/D= 2.00)	CP(R/D= 3.00)	CP(R/D= 4.00)	CP(R/D= 5.00)	CP(R/D= [illegible])
0.5503	0.0682	0.0000	[illegible]	-9.7047E-02	-1.7818E-01	1.0000E 06	-3.5934E-02	-3.1435E-02	-2.5918E-02
0.5503	0.0227	9.0000E 01	4.0924E-02	-2.1245E-01	[illegible]	-3.8001E-02	-4.3153E-02	-2.8077E-02	-2.3336E-02
0.6003	0.0707	0.0000	1.6044E-01	2.7124E-01	-2.4060E-01	-3.1069E-01	-1.2037E-01	-4.3937E-02	-5.5645E-02
0.6003	0.0236	9.0000E 01	9.2130E-02	5.2479E-02	5.4771E-03	-1.5577E-02	-2.4471E-02	-2.7133E-02	-2.7119E-02
0.6503	0.0738	0.0000	-1.5205E-01	-1.0202E-01	9.3140E-02	-2.1544E-01	-7.4045E-02	-7.4065E-02	-6.4270E-02
0.6503	0.0246	9.0000E 01	-1.6550E-01	-1.0851E-01	-8.0352E-02	-6.9336E-02	-6.2852E-02	-5.7995E-02	-5.3649E-02
0.7003	0.0726	0.0000	-4.7124E-01	-3.4774E-01	-1.4115E-01	9.9602E-02	1.8350E-02	-2.3764E-02	-3.8094E-02
0.7003	0.0242	9.0000E 01	-4.9148E-01	-3.1974E-01	-2.0443E-01	-1.5415E-01	-1.2397E-01	-1.0524E-01	-9.2079E-02
0.7503	0.0647	0.0000	-9.4607E-02	-9.4394E-02	-8.9059E-02	-8.5253E-02	-8.2471E-02	-8.0289E-02	-7.8458E-02
0.7503	0.0216	9.0000E 01	-1.2134E-01	-9.9409E-02	-9.0414E-02	-8.5969E-02	-8.2820E-02	-8.0513E-02	-7.8654E-02
0.8003	0.0554	0.0000	-5.9736E-02	-7.4248E-02	-7.9031E-02	-8.0760E-02	-8.1841E-02	-8.2637E-02	-8.3270E-02
0.8003	0.0185	9.0000E 01	-5.8217E-02	-8.2048E-02	-8.1064E-02	-8.1674E-02	-8.2357E-02	-8.2968E-02	-8.3500E-02
0.8503	0.0441	0.0000	-2.9815E-03	-4.2523E-02	-5.8121E-02	-6.6209E-02	-7.1753E-02	-7.6076E-02	-7.9562E-02
0.8503	0.0147	9.0000E 01	-5.5345E-02	-5.0627E-02	-6.0200E-02	-6.7139E-02	-7.2317E-02	-7.6414E-02	-7.9795E-02

DRAG COEFFICIENT = 0.54817E-01

(b) Output.

Figure 12.- Concluded.

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546