

AD-A101 944

MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO
THREE-DIMENSIONAL TRANSONIC FLOW ANALYSIS.(U)

F/G 20/4

JUN 80 G E CHMIELEWSKI; F W SPAID

F44620-76-C-0096

UNCLASSIFIED

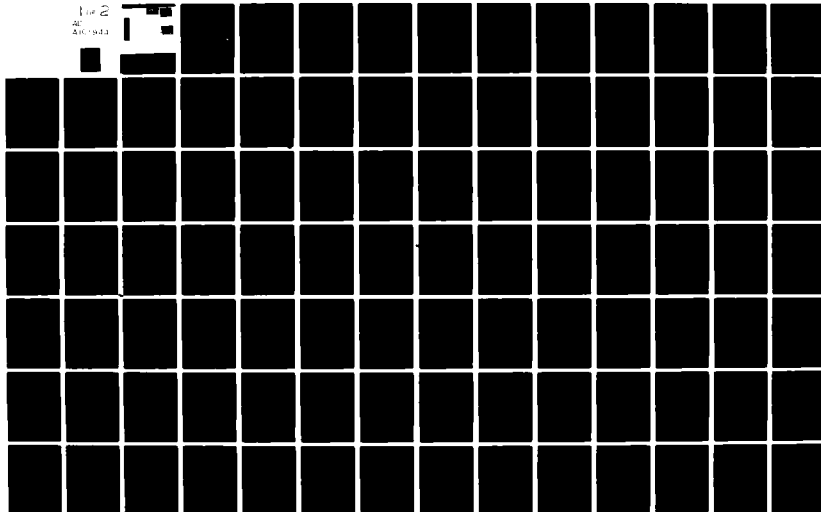
MDC-Q0722

AFOSR-TR-81-0572

NL

1 of 2

AD-A101 944



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR-81-0572	2. GOVT ACCESSION NO. AD-A101944	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THREE-DIMENSIONAL TRANSONIC FLOW ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED <i>Final</i>
7. AUTHOR(s) G. E. Chmielewski		6. PERFORMING ORG. REPORT NUMBER MDC-Q0722
9. PERFORMING ORGANIZATION NAME AND ADDRESS McDonnell Douglas Research Laboratories McDonnell Douglas Corporation St. Louis, Missouri 63166		8. CONTRACT OR GRANT NUMBER(s) F44620-76-C-0096
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Bldg. 410 Bolling Air Force Base, D.C. 20332		10. PROGRAM ELEMENT, PROJECT, AREA & WORK UNIT NUMBERS <i>2307/A1161102A</i>
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 30 June 1980
		13. NUMBER OF PAGES 98
		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Transonic flow	Wing-fuselage configuration	
Potential flow	Finite-difference relaxation scheme	
Aircraft aerodynamics	Non-surface-fitted coordinates	
Computational aerodynamics		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>A computer program was written to calculate the steady, inviscid, transonic flow about an unyawed wing/body configuration. The code is based on the nonconservative form of the full potential equation formulated in global, wing-adapted coordinates that are not surface-conforming. Numerical solution of the governing equations is accomplished via application of type-dependent, rotated finite-difference operators and the method of line relaxation. Exact wing/body surface conditions are enforced by an imaging scheme that involves both surface control points and surface-adjacent computation points. Coordinate</p>		

NOTIC
NOTE
JUL 24 1981

C

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

stretching permits far-field boundary conditions to be treated exactly; an embedded relaxation scheme computes the downwash field in the Trefftz plane. Present geometric capability includes a quite general wing attached at mid-height to a blunt-nosed, semi-infinite cylinder of varying crossplane radius. Representative solutions computed on a grid of moderate density are presented for a checkout configuration that includes ONERA Wing M6; comparisons with wing-alone experimental data are shown also. Code limitations and modifications required to further develop the program are discussed. A code listing and a user's guide summary are provided.

UNCLASSIFIED

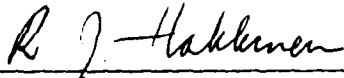
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

PREFACE

This report was prepared by the McDonnell Douglas Research Laboratories (MDRL) for the Air Force Office of Scientific Research, Bolling AFB, D.C., under Contract No. F44620-76-C-0096. The AFOSR Program Monitor was Lt. Col. Robert C. Smith during the early part of the contract period and Dr. James D. Wilson during the latter part.

The work was performed in the Flight Sciences Department of MDRL under the supervision of Dr. Raimo J. Hakkinen. Dr. Frank W. Spaid was the Principal Investigator during the first half of the contract period and Dr. Gerald E. Chmielewski during the latter half.

This report has been reviewed and is approved.



R. J. Hakkinen
Chief Scientist, Flight Sciences
McDonnell Douglas Research Laboratories



D. P. Ames
Staff Vice President
McDonnell Douglas Research Laboratories

Accession For	<input checked="" type="checkbox"/>
GRARI	<input type="checkbox"/>
Classification	<input type="checkbox"/>
Distribution/	<input type="checkbox"/>
Availability Codes	<input type="checkbox"/>
Avail and/or	<input type="checkbox"/>
Special	<input type="checkbox"/>
A	

TABLE OF CONTENTS

	<u>Page</u>
1.0 INTRODUCTION.....	1
2.0 GEOMETRY AND COORDINATES.....	3
3.0 GOVERNING EQUATIONS.....	10
3.1 Physical Domain.....	10
3.2 Computational Domain.....	13
3.3 Local Streamline Coordinates.....	16
3.4 Transformation Derivatives.....	19
4.0 NUMERICAL SOLUTION SCHEME.....	21
4.1 Finite-Difference Approximations.....	21
4.2 Boundary Conditions.....	24
4.3 Computation Procedure.....	28
5.0 RESULTS.....	29
6.0 CONCLUDING REMARKS.....	35
ACKNOWLEDGEMENT.....	37
REFERENCES.....	38
APPENDIX A. USER'S GUIDE TO COMPUTER PROGRAM.....	40
APPENDIX B. LISTING OF COMPUTER PROGRAM.....	47

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Page</u>
1 Wing/body geometry.....	3
2 Coordinate relationships.....	6
3 Schematic of wing-adapted coordinates.....	8
4 Schematic of a computation grid in the physical domain.....	8
5 Schematic diagram of computational domain showing boundary conditions.....	15
6 Grid-point array used to numerically enforce the boundary condition at a typical wing/body surface point.....	26
7 Arrangement of boundary-point arrays relative to locus of zero chordwise slope.....	27
8 Computation schematic: wing-section plane. Illustrates multi-valued dummy points.....	27
9 Calculated surface-pressure distributions on a configuration composed of ONERA Wing M6 attached at mid-height to a hemisphere- cylinder fuselage: $M_\infty = 0.84$, $\alpha = 0^\circ$. Grid dimension: 49 x 19 x 25. Every third spanwise station is shown.....	29
10 Calculated surface-pressure distributions on a configuration com- posed of ONERA Wing M6 attached at mid-height to a hemisphere- cylinder fuselage: $M_\infty = 0.84$, $\alpha = 3.06^\circ$. Grid dimension: 49 x 19 x 25. Every third spanwise station is shown.....	30
11 Comparison of wing/body calculations at $M_\infty = 0.84$, $\alpha = 0^\circ$ with wing- alone experimental data for ONERA Wing M6 at $M_\infty = 0.8399$, $\alpha = 0.04^\circ$. Data are from Reference 19. Parenthetic numbers denote corresponding spanwise positions on the wing/body configuration.....	31
12 Comparison of wing/body calculations at $M_\infty = 0.84$, $\alpha = 3.06^\circ$ with wing-alone experimental data for ONERA Wing M6 at $M_\infty = 0.8395$, $\alpha = 3.06^\circ$. Data are from Reference 19. Parenthetic numbers denote corresponding spanwise positions on the wing/body configuration.....	32
13 Two-grid arrangement for wing/body flow analysis.....	36
A1 Subroutine structure of the wing/body program.....	41

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Comparison of Surface-Slope Values at Two Semispan Stations. DZDX = Prescribed Slope; SLOPE = Calculated Velocity Slope.....	34
A1	Subprogram List.....	42
A2	Glossary of Input Data.....	43
A2	(Continued) Glossary of Input Data.....	44

1.0 INTRODUCTION

Effective transonic operation is required of modern-day transport and fighter aircraft. This consideration has stimulated development of numerical methods to compute transonic flowfields about increasingly complex geometries. In order to be useful for design and performance analysis, such methods must contain accurate, reliable schemes for solving the nonlinear equations governing the flowfields as well as the capability to handle realistic configurations.

Various finite-difference methods have been developed to calculate three-dimensional, transonic potential flowfields about isolated wings and wing/body configurations. To date, these methods have used formulations that require alignment of computing coordinates with appropriate geometric surfaces in order to apply surface boundary conditions. The codes of Ballhaus and Bailey¹⁻³ and Boppe,⁴⁻⁶ which solve various forms of the transonic small-disturbance potential equation, use mean-plane and nearest-point approximations to represent wing and fuselage geometries, respectively; the approximating surface is chosen to coincide with a cartesian grid and is the location where a linearized flow-tangency boundary conditions is enforced. The codes of Jameson and Caughey⁷⁻⁹ are based on the full potential equation and require application of an exact surface condition using analytic and numerical mappings which generate computational grids that conform to the entire surface geometry. Although the small-disturbance approach has so far produced the most extensive geometry-handling capability,⁵⁻⁶ comparative calculations indicate that methods based on the full potential equation provide greater accuracy in matching experimental data.¹⁰

This report presents an alternative approach to transonic wing/body calculations in which the full potential equation is solved using coordinates that, in general, do not coincide with the configuration surface. The wing/body surface appears as a curved boundary within a rectangular computing grid. In order to enforce the flow-tangency condition exactly on this curved boundary, an imaging scheme is used which involves surface control points and surface-adjacent grid points. At the expense of some computational detail, the approximations inherent in a small-disturbance formulation and the

complicated mappings required to produce a surface-conforming grid are both avoided. Schemes similar to the one outlined have been applied previously to calculate flows about simpler, two-dimensional planar and axisymmetric configurations.¹⁰⁻¹⁴

The objective of this contract was to develop a computer program, based on a global system of non-surface-fitted coordinates, to calculate transonic flowfields about wing/finite-fuselage configurations and to compare calculated solutions with relevant experimental data. It was determined part way into the contract period that the coordinate formulation originally proposed as the basis for this effort was inadequate and a more complicated system, involving a higher degree of nonorthogonality, was needed to achieve acceptable solutions. The delay associated with this reformulation as well as some initial difficulty in achieving stable operation of the program have resulted in a geometry capability that consists of a general wing attached to fuselage of semi-infinite length.

Sections 2.0 and 3.0 of this report outline the formulation of the wing/body problem. Section 4.0 describes the numerical solution scheme. A discussion of results is given in Section 5.0, and remarks relating to required code improvements and possible future directions are contained in Section 6.0. Appendix A is a summary user's guide for the computer program, and Appendix B is a program listing.

2.0 GEOMETRY AND COORDINATES

Consider the configuration shown in Figure 1. A wing with multiple sweep, taper, and dihedral is attached at a mid-height position to a blunt-nosed cylinder of semi-infinite length. The wing leading and trailing edges are comprised of straight-line segments; each edge may (but need not) contain a single break at a spanwise location that differs from the other. The wing varies arbitrarily along its span in section profile and section twist angle. It is assumed, however, that the variation of wing properties is piecewise linear between defining sections. The fuselage component of the configuration has an asymmetric side profile and a circular crossplane section of varying radius along its length.

For convenience in formulating the numerical scheme to calculate the flowfield about this class of configurations, a global system of wing-adapted coordinates is introduced followed by a series of coordinate stretchings. This accomplishes several purposes. The wing/body geometry is transformed to a standard and relatively simple form to ease computational bookkeeping. Also, the infinite physical domain is mapped to a finite, rectangular

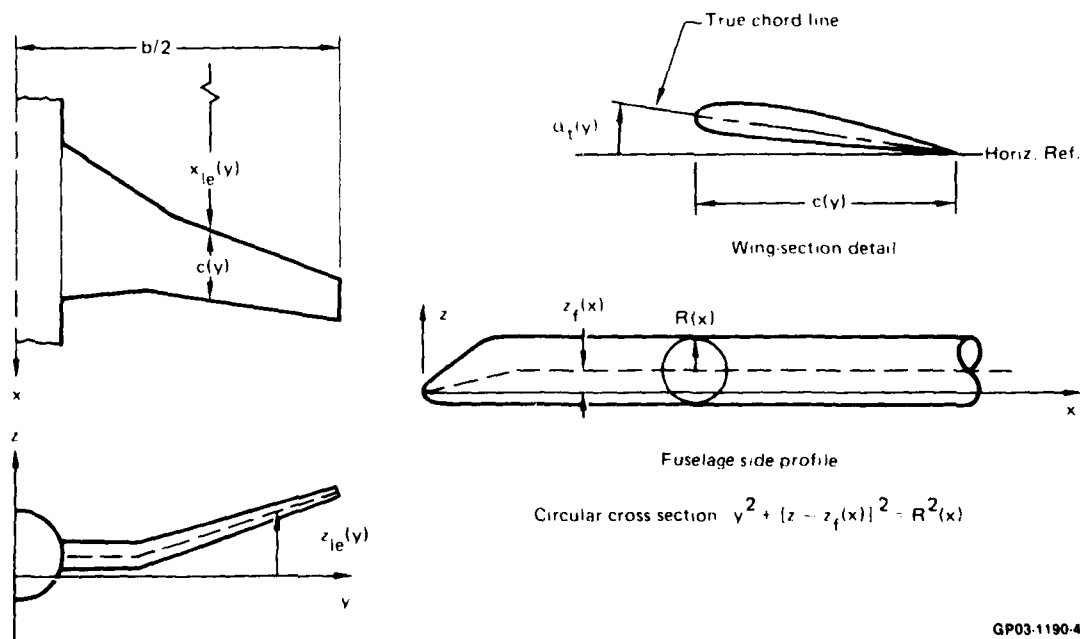


Figure 1. Wing/body geometry.

GP03-1190-4

parallelepiped along whose outer boundaries exact far-field boundary conditions can be applied. (However, the transformed wing/body surface remains embedded within the parallelepiped as an irregular, interior boundary.) Finally, grid-point spacing can be arranged to provide satisfactory surface resolution, at least on the wing at the present stage of development, as well as efficient distribution of points throughout the computational domain (dense near the configuration surface and progressively more sparse as distance from the surface increases).

The wing-adapted coordinates are defined by the following relations:

$$X = \frac{x - x_{1e}(y)}{c(y)} - \frac{1}{2}, \quad Y = \frac{y}{b}, \quad Z = \frac{z - z_m(x,y)}{\tau(y) c(y)}, \quad (1)$$

where $x_{1e}(y)$ represents the streamwise position (sweep) of the wing leading edge, $c(y)$ is the wing-section chord, b is the wing span, $\tau(y)$ is the wing-section thickness-to-chord ratio normalized by its value at the wing root, and $z_m(x,y)$ is a nonplanar mean surface determined by the wing geometry. This mean surface is specified as follows:

$$z_m(x,y) = z_{1e}(y) \quad [x < x_{1e}(y)], \quad (2a)$$

$$= z_{1e}(y) - [x - x_{1e}(y)] \tan \alpha_t(y) \quad [x_{1e}(y) < x < x_{te}(y)], \quad (2b)$$

$$= z_{1e}(y) - c(y) \tan \alpha_t(y) \quad [x > x_{te}(y)], \quad (2c)$$

where $z_{1e}(y)$ denotes the vertical position (dihedral) of the wing leading edge, $\alpha_t(y)$ is the wing-section twist angle, and $x_{te}(y)$ gives the streamwise location of the wing trailing edge. Note that $c(y) = x_{1e}(y) - x_{te}(y)$.

The transformation Equations (1) effectively shear out wing sweep, taper in both chord and thickness, dihedral, and twist. In the (X,Y,Z) coordinates, the wing appears to be planar with a rectangular planform and uniform thickness over its entire span. The constant in the x-transformation shifts the X-origin to the wing mid-chord locus; the leading and trailing edges are at $X = \pm 1/2$, respectively. Lines of constant X coincide with constant-percentage-chord stations along the span. The Y origin is at the wing/body centerline, and the wingtip is at $Y = 1/2$. The Z-origin is on the mean

surface $z_m(x,y)$. Except for the nonplanar wing mean surface $z_m(x,y)$, the (X,Y,Z) coordinates are similar to those used in transonic small-disturbance formulations.

The spanwise region containing the fuselage and the region outboard of the wingtip are treated by extending the wing leading and trailing edges while holding the section twist angle fixed at its wing root and wingtip values, respectively. The fuselage transforms in a regular but not well-defined manner with distortion in length, profile, and crossplane section. In dealing with the outboard region, the normalizing chord length in Equation (1) is held fixed at a constant value $c(y_3)$ beyond an arbitrarily specified spanwise position $y = y_3$ to prevent crossing of the extended wing edges.

In order to produce a finite computational domain, each of the (X,Y,Z) coordinates is stretched individually as described below. The stretching formulas are adaptations to the present application of functions used successfully to compute the transonic flow about an airfoil.¹² Figure 2 shows the relationship between the chordwise and spanwise coordinates and also indicates the different regions of the physical, intermediate, and stretched domains.

• X-coordinate to ξ -coordinate

$$\text{Region II: } X = \xi(a_1 + a_2 \xi^2) \quad (3a)$$

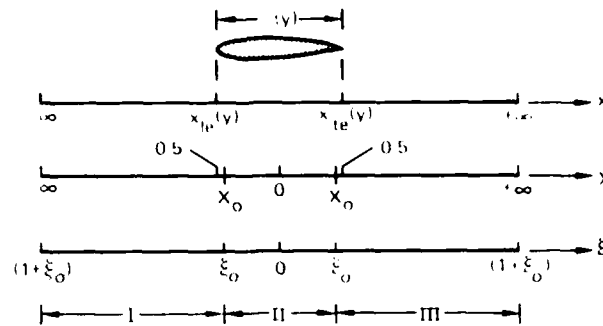
$$\text{Region I,III: } X = \mp X_0 + A_1 \tan \left[\frac{\pi}{2}(\xi \pm \xi_0) \right] + A_2 \tan \left[\frac{\pi}{2}(\xi \pm \xi_0)^3 \right] \quad (3b)$$

In Equation (3b), the upper set of signs refers to region I and the lower set to region III. The three streamwise regions encompass the following ranges:

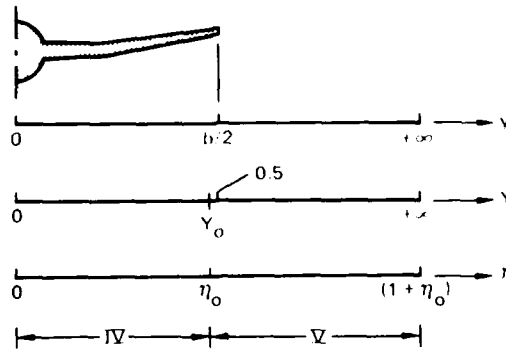
$$\text{I: } -\infty < X < -X_0, \quad -(1 + \xi_0) < \xi < -\xi_0, \quad (4a)$$

$$\text{II: } -X_0 < X < X_0, \quad -\xi_0 < \xi < \xi_0, \quad (4b)$$

$$\text{III: } X_0 < X < \infty, \quad \xi_0 < \xi < (1 + \xi_0). \quad (4c)$$



(a) Wing-section plane



(b) Wing/body crossplane

GP03-1190-15

Figure 2. Coordinate relationships.

Regions I and III nominally represent the regions upstream of the wing leading edge and downstream of the trailing edge, respectively, while Region II covers the wing-section chord. This stretching is symmetric about the origin. Constants a_1 and a_2 are determined by the conditions $X = X_0$ and $dX/d\xi = \pi A_1/2$ at $\xi = \xi_0$. The transition stations $\pm X_0$ between the cubic and tangent stretching functions are chosen to occur a small distance inside the wing leading and trailing edges. Parameter ξ_0 determines how much of the ξ -domain is confined between the wing edges. For an evenly spaced ξ -grid, the number of ξ -steps in Region II compared with the total number of ξ -steps will be in the ratio $\xi_0/(1 + \xi_0)$. Constants A_1 and A_2 mainly allow control over grid-point spacing in regions I and III but also provide some control over the uniformity of spacing in Region II.

• Y-coordinate to η -coordinate

$$\text{Region IV: } Y = \eta(b_1 + b_2\eta^2) \quad (5a)$$

$$\text{Region V: } Y = Y_0 + B_1 \tan \left[\frac{\pi}{2}(\eta - \eta_0) \right] + B_2 \tan \left[\frac{\pi}{2}(\eta - \eta_0)^3 \right] \quad (5b)$$

Spanwise regions cover the following ranges:

$$\text{IV: } 0 < Y < Y_0, \quad 0 < \eta < \eta_0, \quad (6a)$$

$$\text{V: } Y_0 < Y < \infty, \quad \eta_0 < \eta < (1 + \eta_0). \quad (6b)$$

This spanwise stretching is similar to the one used in the streamwise direction but is applied only to the half-space because only unyawed wing/body configurations are considered. Region IV extends over the wing/body semispan, and Region V encompasses the domain outboard of the wingtip. Constants b_1 and b_2 follow from the requirements that $Y = Y_0$ and $dY/d\eta = \pi B_1/2$ at $\eta = \eta_0$. The transition station Y_0 is set slightly inside the wingtip, and parameter η_0 determines the fraction of the η -domain confined to the semispan region. Constants B_1 and B_2 control spacing of the grid outboard of the wing tip, in region V.

• Z-coordinate to ζ -coordinate

$$Z = C_1 \tan (\pi\zeta/2) \quad (7)$$

Constant C_1 controls vertical grid-point spacing near the wing mean surface. The range of this stretching is:

$$-\infty < Z < \infty, \quad -1 < \zeta < 1. \quad (8)$$

Figure 3 is a schematic representation of the nonorthogonal (ξ, η, ζ) coordinate system which indicates its relationship to the wing geometry.

Figure 4 shows a schematic representation in the physical (x, y, z) domain of a grid-point distribution that is uniformly spaced in each of the (ξ, η, ζ)

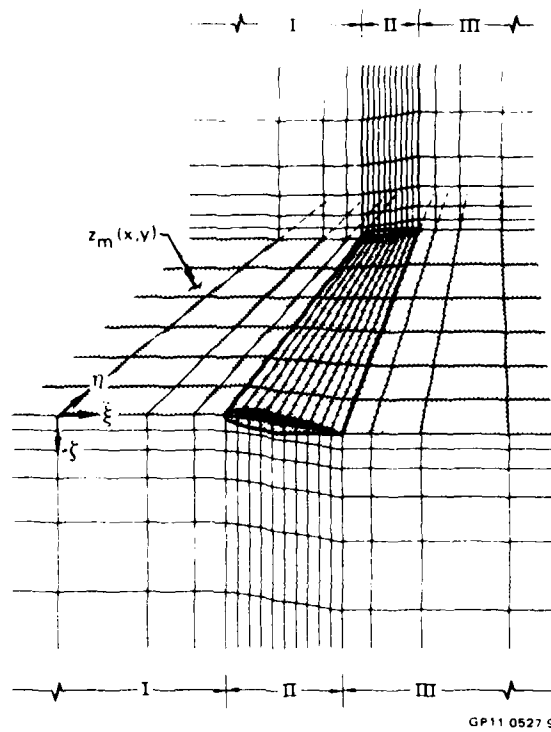


Figure 3. Schematic of wing-adapted coordinates.

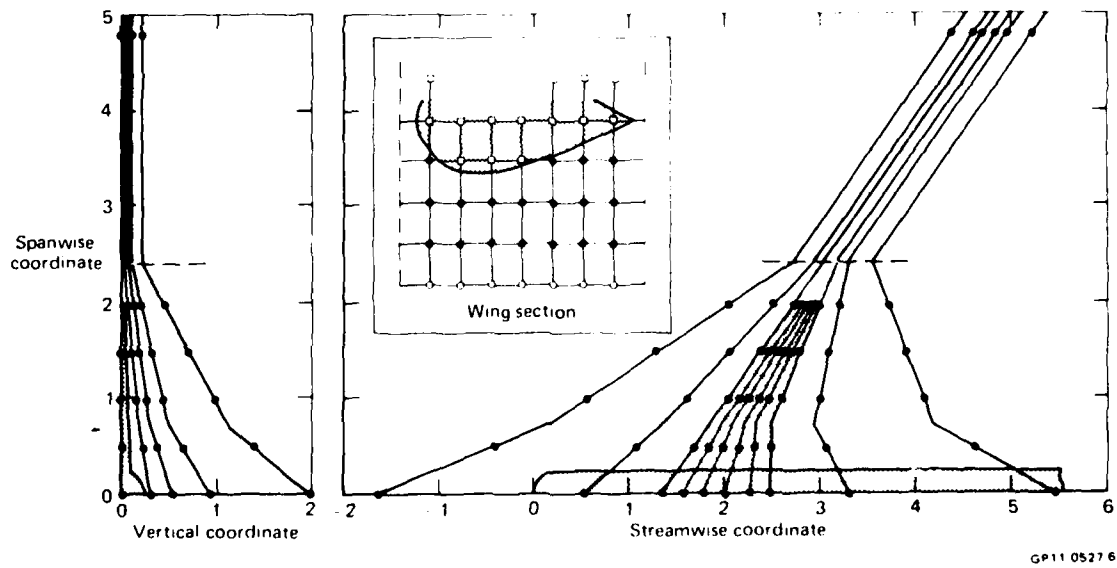


Figure 4. Schematic of a computation grid in the physical domain.

directions. The configuration shown involves a planar wing with a straight leading edge. Each point in the schematic actually represents a line of grid points. Grid taper and the clustering of points near the wing mean surface and between the wing edges are clearly evident. The dashed line corresponds to the spanwise station $y = y_3$ discussed previously. The planform and crossplane views of Figure 4 emphasize the wing orientation of the present coordinates and illustrate their lack of suitability for representing the fuselage region. A method for improving resolution about the fuselage is described in Section 5.0. The wing-section detail emphasizes the nonconformity between the computation coordinates and the geometry surface.

For later use in writing the transformed differential equations that govern the wing/body flowfield, the stretching functions are denoted symbolically as

$$\xi = \xi(X), \quad \eta = \eta(Y), \quad \zeta = \zeta(Z). \quad (9)$$

Stretching derivatives are then defined by the relations

$$f(\xi) = d\xi/dX, \quad g(\eta) = d\eta/dY, \quad h(\zeta) = d\zeta/dZ. \quad (10)$$

Evaluation of these derivatives follows from Equations (3), (5), and (7), respectively.

3.0 GOVERNING EQUATIONS

3.1 Physical Domain

The steady, inviscid, transonic flow past a wing/body configuration can be considered isentropic, and hence irrotational, if only weak shock waves are present. Such a flow can be characterized by a velocity potential $\phi(x,y,z)$ that is related to the streamwise x-component, the spanwise y-component, and the vertical z-component of velocity by the relations

$$u = \phi_x, \quad v = \phi_y, \quad w = \phi_z, \quad (11)$$

where coordinate-symbol subscripts denote partial differentiation. In Equation (11) and below, it is assumed that all velocities are normalized by the freestream speed q_∞ .

Since ϕ is singular at infinity, it is convenient to introduce a perturbation potential $\phi(x,y,z)$ according to the expression

$$\phi = x \cos \alpha + z \sin \alpha + \phi, \quad (12)$$

where α is the angle of attack of the incident flow measured in a wing-section (x-z) plane. Then, the governing equation for the flowfield, written in cartesian coordinates, has the form

$$\begin{aligned} (a^2 - u^2)\phi_{xx} + (a^2 - v^2)\phi_{yy} + (a^2 - w^2)\phi_{zz} - 2uv\phi_{xy} \\ - 2vw\phi_{yz} - 2uw\phi_{xz} = 0, \end{aligned} \quad (13)$$

where

$$u = \cos \alpha + \phi_x, \quad (14a)$$

$$v = \phi_y, \quad (14b)$$

$$w = \sin \alpha + \phi_z. \quad (14c)$$

The local speed of sound a is determined by the relation

$$a^2 = \frac{1}{M_\infty^2} + \left(\frac{\gamma - 1}{2} \right) (1 - q^2), \quad (15)$$

where $q^2 = u^2 + v^2 + w^2$, M_∞ is the freestream Mach number, and γ represents the ratio of specific heats of the medium (for air, $\gamma = 1.4$).

The pressure coefficient at any point in the flowfield is given by the expression

$$C_p = \frac{2}{\gamma M_\infty^2} \left\{ \left[1 + \left(\frac{\gamma - 1}{2} \right) M_\infty^2 (1 - q^2) \right]^{\gamma/(\gamma-1)} - 1 \right\}. \quad (16)$$

The boundary condition for Equation (13) on the wing/body requires that the flow be tangent to the surface and can be written as

$$\left[(\cos \alpha + \phi_x) \mathcal{S}'_x + \phi_y \mathcal{S}'_y - (\sin \alpha + \phi_z) \right]_{\text{surface}} = 0; \quad (17)$$

derivatives \mathcal{S}'_x and \mathcal{S}'_y are streamwise and spanwise surface slopes, respectively. In addition, the Kutta condition requires that a circulation $\Gamma(y_0)$ must exist at each spanwise wing station y_0 which is of such magnitude that the flow passes smoothly off the sharp trailing edge. A vortex sheet extends behind the wing whose strength corresponds to the spanwise variation of the circulation. Using a linearized model that neglects roll-up, we assume that the vortex sheet coincides with the wing mean surface, defined by Equation (2c), between the trailing edge and downstream infinity. There is a jump $\Gamma(y_0)$ in the potential function across the sheet which is constant along lines lying in the sheet that are parallel to the freestream direction. The normal component of velocity and the pressure must be continuous through the sheet, however. Thus, on the vortex sheet,

$$\phi(x_{te}, y_0, z_m^+) - \phi(x_{te}, y_0, z_m^-) = \Gamma(y_0), \quad \phi_z \text{ continuous.} \quad (18)$$

Far from the wing/body, the flow is undisturbed except in the downstream Trefftz plane where the vortex sheet induces a two-dimensional downwash flow. Thus,

$$\phi(\infty) = 0 \quad (19)$$

everywhere except on the y-z plane at downstream infinity where the potential function satisfies the equation

$$(a_\infty^2 - v^2) \phi_{yy} - 2vw\phi_{yz} + (a_\infty^2 - w^2) \phi_{zz} = 0 \quad (20)$$

subject to the boundary conditions

$$\left[\phi_y \mathcal{P}_y - (\sin \alpha + \phi_z) \right]_{\text{surface}} = 0, \quad (21a)$$

$$\phi(\infty, y, z_m^+) - \phi(\infty, y, z_m^-) = \Gamma(y) \text{ on the Kutta slit}, \quad (21b)$$

$$\phi \rightarrow 0 \quad \text{at} \quad \sqrt{y^2 + z^2} \rightarrow \infty. \quad (21c)$$

The Kutta slit corresponds to the crossplane profile of the vortex sheet at infinity. From Equation (15), it follows that $a_\infty = 1/M_\infty$. Boundary condition (21a) applies on a cross-section of the semi-infinitely long fuselage.

For the unyawed case, only half of the configuration need be considered, and the symmetry condition $v = 0$ is imposed on the vertical plane $y = 0$, which contains the fuselage centerline.

Equations (13)-(21) collectively define the problem for the wing/body flowfield. The two-dimensional problem for the downwash field in the Trefftz plane is embedded within the overall three-dimensional problem.

3.2 Computational Domain

The computational problem is formulated by transforming the equations of Section 3.1 into (ξ, η, ζ) coordinates. Substituting Equations (1) and (9) into Equations (13) - (14) and using the symbolic definitions given in Equation (10) yields the equation for the perturbation potential:

$$A(f\phi_{\xi})_{\xi} + B(g\phi_{\eta})_{\eta} + C(h\phi_{\zeta})_{\zeta} + D\phi_{\xi\eta} + E\phi_{\eta\zeta} + F\phi_{\xi\zeta} = J, \quad (22)$$

where

$$A = \left[\frac{(a^2 - u^2)}{c^2(\eta)} + (a^2 - v^2) G^2(\xi, \eta) - 2uv \frac{G(\xi, \eta)}{c(\eta)} \right] f(\xi),$$

$$B = (a^2 - v^2) g(\eta)/b^2,$$

$$C = \left\{ (a^2 - u^2) \bar{G}^2(\xi, \eta) + (a^2 - v^2) \tilde{G}^2(\xi, \eta, \zeta) + \frac{(a^2 - w^2)}{\tau^2(\eta)c^2(\eta)} \right. \\ \left. - 2uv\bar{G}(\xi, \eta)\tilde{G}(\xi, \eta, \zeta) - \frac{2w}{\tau(\eta)c(\eta)} \left[u\bar{G}(\xi, \eta) + v\tilde{G}(\xi, \eta, \zeta) \right] \right\} h(\zeta),$$

$$D = \frac{2}{b} \left[(a^2 - v^2) G(\xi, \eta) - \frac{uv}{c(\eta)} \right] f(\xi)g(\eta),$$

$$E = \frac{2}{b} \left[(a^2 - v^2) \tilde{G}(\xi, \eta, \zeta) - uv\bar{G}(\xi, \eta) + \frac{vw}{\tau(\eta)c(\eta)} \right] g(\eta)h(\zeta),$$

$$F = 2 \left\{ (a^2 - u^2) \frac{\bar{G}(\xi, \eta)}{c(\eta)} + (a^2 - v^2) G(\xi, \eta)\tilde{G}(\xi, \eta, \zeta) - uv \right. \\ \left. \cdot \left[\frac{\tilde{G}(\xi, \eta, \zeta)}{c(\eta)} + \bar{G}(\xi, \eta)G(\xi, \eta) \right] - \frac{w}{\tau(\eta)c(\eta)} \left[vG(\xi, \eta) + \frac{u}{c(\eta)} \right] \right\} f(\xi)h(\zeta),$$

$$\begin{aligned}
J = & \left[2uvI(n) - (a^2 - v^2) H(\xi, n) \right] \\
& \cdot \left[u - \cos \alpha - \bar{G}(\xi, n) (w - \sin \alpha) \tau(n)c(n) \right] c(n) \\
& + \left\{ 2v \left[u\tilde{I}(\xi, n) + w\bar{I}(n) \right] - (a^2 - v^2) \tilde{H}(\xi, n, \zeta) \right\} (w - \sin \alpha) \tau(n)c(n),
\end{aligned}$$

and

$$u = \cos \alpha + \frac{f(\xi)}{c(n)} \phi_\xi + \bar{G}(\xi, n) h(\zeta) \phi_\zeta, \quad (23a)$$

$$v = G(\xi, n) f(\xi) \phi_\xi + \frac{g(n)}{b} \phi_n + \tilde{G}(\xi, n, \zeta) h(\zeta) \phi_\zeta, \quad (23b)$$

$$w = \sin \alpha + \frac{h(\zeta)}{\tau(n)c(n)} \phi_\zeta. \quad (23c)$$

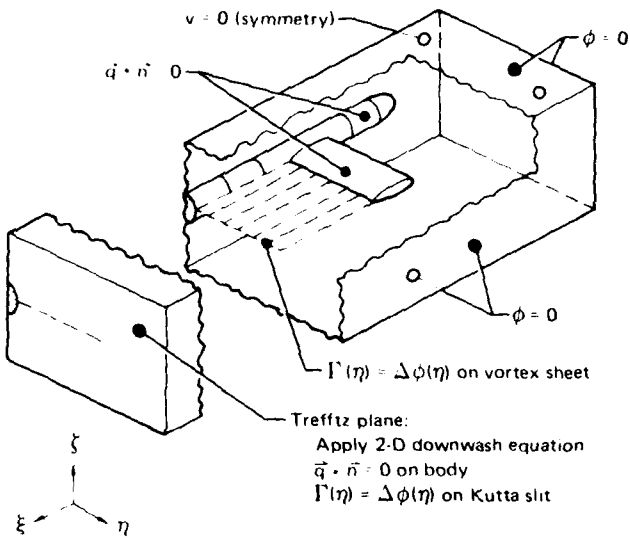
The functions $G(\xi, n)$, $\bar{G}(\xi, n)$, $\tilde{G}(\xi, n, \zeta)$, $H(\xi, n)$, $\tilde{H}(\xi, n, \zeta)$, $I(n)$, $\bar{I}(n)$ and $\tilde{I}(\xi, n)$ are transformation derivatives (see Section 3.4); $\tilde{G}(\xi, n, \zeta)$ and $\tilde{H}(\xi, n, \zeta)$ break into sums of two-dimensional functions so that the potential function $\phi(\xi, n, \zeta)$ remains as the only three-dimensional quantity in Equation (22).

Figure 5 is a schematic diagram of the finite (ξ, n, ζ) domain and shows the conditions applicable on its various boundaries. The surface condition (17) takes the form:

$$\left[K\phi_\xi + L\phi_n + M\phi_\zeta + N \right]_{\text{surface}} = 0, \quad (24)$$

where

$$K = \left[\frac{\mathcal{L}_x(\xi, n)}{c(n)} + G(\xi, n) \mathcal{L}_y(\xi, n) \right] f(\xi),$$



GP03-1190-5

Figure 5. Schematic diagram of computational domain showing boundary conditions.

$$L = \mathcal{L}'_y(\xi, \eta)g(\eta)/b,$$

$$M = \left[\tilde{G}(\xi, \eta) \mathcal{L}'_x(\xi, \eta) + \tilde{G}(\xi, \eta, \zeta) \mathcal{L}'_y(\xi, \eta) - \frac{1}{\tau(\eta)c(\eta)} \right] h(\zeta),$$

$$N = (\cos \alpha) \mathcal{L}'_x(\xi, \eta) - \sin \alpha.$$

The far-field boundary condition remains $\phi = 0$ on all edges of the computational domain corresponding to infinity except the transformed Trefftz plane where the governing equation becomes

$$B^T(g\phi_{\eta})_{\eta} + C^T(h\phi_{\zeta})_{\zeta} + E^T\phi_{\eta\zeta} = J^T, \quad (25)$$

with

$$B^T = (a_{\infty}^2 - v^2) g(\eta)/b^2,$$

$$C^T = \left[(a_\infty^2 - v^2) \tilde{G}^2(1+\xi_0, n, \zeta) + \frac{(a_\infty^2 - w^2)}{\tau^2(n)c^2(n)} - 2vw \frac{\tilde{G}(1+\xi_0, n, \zeta)}{\tau(n)c(n)} \right] h(\zeta),$$

$$E^T = \frac{2}{b} \left[(a_\infty^2 - v^2) \tilde{G}(1+\xi_0, n, \zeta) - \frac{vw}{\tau(n)c(n)} \right] g(n)h(\zeta),$$

$$J^T = \left\{ 2vw\bar{I}(n) - (a_\infty^2 - v^2) \tilde{H}(1+\xi_0, n, \zeta) \right\} (w - \sin \alpha) \tau(n)c(n),$$

and velocities v and w follow from Equations (23b) and (23c) with $\phi_\xi = 0$. Equation (25) is subject to the boundary conditions

$$\left\{ \mathcal{P}_y(1+\xi_0, n) \frac{g(n)}{b} \phi_n + \left[\tilde{G}(1+\xi_0, n, \zeta) \mathcal{P}_y(1+\xi_0, n) - \frac{1}{\tau(n)c(n)} \right] h(\zeta) \phi_\zeta - \sin \alpha \right\}_{\text{surface}} = 0, \quad (26a)$$

$$\phi(\xi_{te}, n, 0^+) - \phi(\xi_{te}, n, 0^-) = \Gamma(n) \text{ on Kutta slit}, \quad (26b)$$

$$\phi = 0 \quad \text{at} \quad n = 1+n_0 \quad \text{or} \quad \zeta = \pm 1. \quad (26c)$$

On the wing/body symmetry plane, the condition $v = 0$ is enforced using Equation (23b).

3.3 Local Streamline Coordinates

For use in applying upwind-biased finite differences (Section 4.1), the potential equation is rewritten in coordinates that are locally aligned with the stream direction, which is denoted by S . In such coordinates, the principal part of Equation (13) takes the form:

$$(a^2 - q^2) \phi_{SS} + a^2 (\Delta\phi - \phi_{SS}) = 0, \quad (27)$$

where

$$\phi_{SS} = q^{-2} (u^2 \phi_{xx} + v^2 \phi_{yy} + w^2 \phi_{zz} + 2uv \phi_{xy} + 2vw \phi_{yz} + 2uw \phi_{xz}) \quad (28)$$

and $\Delta\phi$ represents the Laplacian:

$$\Delta\phi = \phi_{xx} + \phi_{yy} + \phi_{zz}. \quad (29)$$

In computational (ξ, η, ζ) coordinates,

$$\phi_{SS} = q^{-2} \left[P(f\phi_{\xi})_{\xi} + Q(g\phi_{\eta})_{\eta} + R(h\phi_{\zeta})_{\zeta} + S\phi_{\xi\eta} + T\phi_{\eta\zeta} + V\phi_{\xi\zeta} + W \right], \quad (30)$$

with

$$P = \left[\frac{u^2}{c^2(\eta)} + v^2 G^2(\xi, \eta) + 2uv \frac{G(\xi, \eta)}{c(\eta)} \right] f(\xi),$$

$$Q = v^2 g(\eta) / b^2$$

$$R = \left\{ u^2 \bar{G}^2(\xi, \eta) + v^2 \tilde{G}^2(\xi, \eta, \zeta) + \frac{w^2}{\tau^2(\eta)c^2(\eta)} + 2uv \bar{G}(\xi, \eta) \tilde{G}(\xi, \eta, \zeta) \right. \\ \left. + \frac{2w}{\tau(\eta)c(\eta)} \left[u \bar{G}(\xi, \eta) + v \tilde{G}(\xi, \eta, \zeta) \right] \right\} h(\zeta),$$

$$S = \frac{2}{b} \left[v^2 G(\xi, \eta) + \frac{uv}{c(\eta)} \right] f(\xi)g(\eta),$$

$$T = \frac{2}{b} \left[v^2 \tilde{G}(\xi, \eta, \zeta) + uv \bar{G}(\xi, \eta) + \frac{vw}{\tau(\eta)c(\eta)} \right] g(\eta)h(\zeta),$$

$$V = 2 \left\{ \frac{u^2 \bar{G}(\xi, \eta)}{c(\eta)} + v^2 G(\xi, \eta) \tilde{G}(\xi, \eta, \zeta) + uv \left[\frac{\tilde{G}(\xi, \eta, \zeta)}{c(\eta)} + \bar{G}(\xi, \eta) G(\xi, \eta) \right] \right\}$$

$$\begin{aligned}
& + \frac{w}{\tau(n)c(n)} \left[v G(\xi, n) + \frac{u}{c(n)} \right] \left\{ f(\xi)h(\zeta), \right. \\
W = & \left[2uvI(n) + v^2 H(\xi, n) \right] \left[u \cos \alpha - \bar{G}(\xi, n) (w - \sin \alpha) \tau(n)c(n) \right] c(n) \\
& + \left\{ 2v \left[u\bar{I}(\xi, n) + w\bar{I}(n) \right] + v^2 \tilde{H}(\xi, n, \zeta) \right\} (w - \sin \alpha) \tau(n)c(n).
\end{aligned}$$

The Laplacian becomes

$$\Delta \phi = P_1 (f\phi_\xi)_\xi + Q_1 (g\phi_\zeta)_\zeta + R_1 (h\phi_\zeta)_\zeta + S_1 \phi_{\xi n} + T_1 \phi_{\zeta \zeta} + V_1 \phi_{\xi \zeta} + W_1, \quad (31)$$

where

$$P_1 = \left[\frac{1}{c^2(n)} + G^2(\xi, n) \right] f(\xi),$$

$$Q_1 = g(n)/b^2,$$

$$R_1 = \left[\bar{G}^2(\xi, n) + \tilde{G}^2(\xi, n, \zeta) + \frac{1}{\tau^2(n)c^2(n)} \right] h(\zeta),$$

$$S_1 = 2G(\xi, n)f(\xi)g(n)/b,$$

$$T_1 = 2\tilde{G}(\xi, n, \zeta)g(n)h(\zeta)/b,$$

$$V_1 = 2 \left[\frac{\bar{G}(\xi, n)}{c(n)} + G(\xi, n) \tilde{G}(\xi, n, \zeta) \right] f(\xi)h(\zeta),$$

$$W_1 = H(\xi, n) \left[u - \cos \alpha - \bar{G}(\xi, n) (w - \sin \alpha) \tau(n)c(n) \right] c(n) \\ + \tilde{H}(\xi, n, \zeta) (w - \sin \alpha) \tau(n)c(n).$$

3.4 Transformation Derivatives

In Sections 3.2 and 3.3, various quantities appear which represent transformation derivatives between the (x, y, z) and (X, Y, Z) coordinates. These derivatives follow from Equation (1) according to the following definitions:

$$G(x, y) = \frac{\partial X}{\partial y} \longrightarrow G(\xi, n) \quad (32)$$

$$H(x, y) = \frac{\partial^2 X}{\partial y^2} \longrightarrow H(\xi, n) \quad (33)$$

$$I(y) = \frac{\partial^2 X}{\partial x \partial y} \longrightarrow I(n) \quad (34)$$

$$\tilde{G}(x, y, z) = \frac{\partial Z}{\partial y} \longrightarrow \tilde{G}(\xi, n, \zeta) = \tilde{G}_1(n, \zeta) + \tilde{G}_2(\xi, n) \quad (35)$$

$$\tilde{H}(x, y, z) = \frac{\partial^2 Z}{\partial y^2} \longrightarrow \tilde{H}(\xi, n, \zeta) = \tilde{H}_1(n, \zeta) + \tilde{H}_2(\xi, n) \quad (36)$$

$$\tilde{I}(x, y) = \frac{\partial^2 Z}{\partial x \partial y} \longrightarrow \tilde{I}(\xi, n) \quad (37)$$

$$\bar{G}(x, y) = \frac{\partial Z}{\partial x} \longrightarrow \bar{G}(\xi, n) \quad (38)$$

$$\bar{I}(y) = \frac{\partial^2 Z}{\partial y \partial z} \longrightarrow \bar{I}(n) \quad (39)$$

Evaluation of these transformation derivatives in the computational domain uses the one-to-one correspondence that exists between grid points in the (x,y,z) , (X,Y,Z) , and (ξ,η,ζ) coordinate systems. The splitting of the functional dependence shown in Equations (35) and (36) actually occurs in the (X,Y,Z) system. Transformation derivatives which are not included among Equations (32) - (39) either are zero or have been explicitly evaluated in the equations of the previous sections.

4.0 NUMERICAL SOLUTION SCHEME

4.1 Finite-Difference Approximations

The type-dependent finite-difference concept introduced by Murman and Cole¹⁶ is the basis for numerical schemes to compute steady-state transonic flowfields. Central differences are used to approximate the potential equation at subsonic points of the solution domain, and upwind-biased differences are used at supersonic points. Thus, the mathematical character of the equation is properly represented as it changes type from elliptic to hyperbolic. The original application involved the transonic small-disturbance potential equation.

In order to apply upwind differences to the full potential equation, it is necessary to take into account the misalignment between coordinate lines and the velocity vector at any given point of the flowfield. The principal part of the equation is recast in a form that constitutes an effective rotation to the local stream direction, denoted by S :¹⁷

$$(1 - M^2)\phi_{SS} + (\Delta\phi - \phi_{SS}) = 0, \quad (40)$$

where $M = q/a$ is the local Mach number and the quantities ϕ_{SS} and $\Delta\phi$ are defined in Section 3.3. Then, upwind differences are used to approximate contributions to the first term on the lefthand side of Equation (40), and central differences are applied to factors associated with the second term.

Jameson has shown further that the relaxation procedure for the finite-difference counterpart of Equation (40) can be viewed in terms of a damped, three-dimensional wave equation involving an artificial time.⁷ The time dependence arises because of the appearance in difference formulas at each grid point of both a new solution value $\phi_{i,j,k}^{(n+1)}$, from the current relaxation step, and an old value $\phi_{i,j,k}^{(n)}$, from the previous relaxation step. Thus, timelike terms occur implicitly which correspond to ϕ_{St} . These terms play a role in controlling the stability of the relaxation process but do not affect the final solution since they vanish as the process converges. Sometimes the damping inherent in the finite-difference analog of Equation (40) must be

augmented; this can be accomplished by adding to Equation (40) a term of the form

$$\begin{aligned} \bar{A}\varphi_{St} &= -\varepsilon \frac{u}{q} \frac{\Delta t}{\Delta x} \left(\frac{u}{q} \varphi_{xt} + \frac{v}{q} \varphi_{yt} + \frac{w}{q} \varphi_{zt} \right) \\ &= -\varepsilon \Delta t \left[P_2 f(\xi) \varphi_{\xi t} + Q_2 g(\eta) \varphi_{\eta t} + R_2 h(\zeta) \varphi_{\zeta t} \right], \end{aligned} \quad (41)$$

with

$$P_2 = \frac{f(\xi)u}{q^2 c(\eta) \Delta \xi} \left[\frac{u}{c(\eta)} + vG(\xi, \eta) \right],$$

$$Q_2 = \frac{f(\xi)u}{q^2 c(\eta) \Delta \xi} \frac{v}{b},$$

$$R_2 = \frac{f(\xi)u}{q^2 c(\eta) \Delta \xi} \left[\bar{u}G(\eta) + \tilde{v}G(\xi, \eta, \zeta) + \frac{w}{\tau(\eta)c(\eta)} \right].$$

In Equation (41), the value of parameter ε is arbitrary and can be adjusted to control the amount of damping augmentation.

In writing difference approximations, central differences based on old values are used to calculate first derivatives required to evaluate the velocity components in Equations (23a) - (23c). At grid points where the flow is subsonic, Equation (22) is used, and second derivatives are represented by central differences. Typical formulas are

$$\begin{aligned}
(f\psi_\xi)_\xi &= \frac{1}{2(\Delta\xi)^2} \left\{ (f_{i+1,j,k} + f_{i,j,k}) \phi_{i+1,j,k}^{(n)} \right. \\
&\quad - (f_{i+1,j,k} + 2f_{i,j,k} + f_{i-1,j,k}) \left[\frac{1}{\omega} \phi_{i,j,k}^{(n+1)} + \left(1 - \frac{1}{\omega}\right) \phi_{i,j,k}^{(n)} \right] \\
&\quad \left. + (f_{i,j,k} + f_{i-1,j,k}) \phi_{i-1,j,k}^{(n+1)} \right\}, \tag{42}
\end{aligned}$$

$$\begin{aligned}
(g\psi_\eta)_\eta &= \frac{1}{2(\Delta\eta)^2} \left[(g_{i,j+1,k} + g_{i,j,k}) (\phi_{i,j+1,k}^{(n)} - \phi_{i,j,k}^{(n)}) \right. \\
&\quad \left. - (g_{i,j,k} + g_{i,j-1,k}) (\phi_{i,j,k}^{(n+1)} - \phi_{i,j-1,k}^{(n+1)}) \right], \tag{43}
\end{aligned}$$

$$\psi_{\xi\xi} = \frac{1}{4\Delta\xi\Delta\xi} \left[\phi_{i+1,j,k+1}^{(n)} - \phi_{i+1,j,k-1}^{(n)} + \phi_{i-1,j,k-1}^{(n+1)} - \phi_{i-1,j,k+1}^{(n+1)} \right]. \tag{44}$$

In Equation (42), the relaxation factor ω has been incorporated into the difference expression. Also, in Equations (42) and (43), stretching function values required at half-step points have been evaluated as the average of the values at grid points on either side; the averaging is a second-order approximation to the stretching function at the half-step location. At grid points where evaluation of the local velocity indicates the flow to be supersonic, Equation (40) is used, and second derivatives contributing to ψ_{SS} in the first term are upwind differenced. For example, in the case where velocity components u , v , and w are all positive,

$$\begin{aligned}
(f\psi_\xi)_\xi &= \frac{1}{2(\Delta\xi)^2} \left[(f_{i,j,k} + f_{i-1,j,k}) (2\phi_{i,j,k}^{(n+1)} - \phi_{i,j,k}^{(n)} - \phi_{i-1,j,k}^{(n+1)}) \right. \\
&\quad \left. - (f_{i-1,j,k} + f_{i-2,j,k}) (\phi_{i-1,j,k}^{(n+1)} - \phi_{i-2,j,k}^{(n)}) \right], \tag{45}
\end{aligned}$$

$$\psi_{\xi\zeta} = \frac{1}{\Delta\xi\Delta\zeta} \left[\psi_{i,j,k}^{(n+1)} - \psi_{i,j,k-1}^{(n+1)} + \psi_{i-1,j,k-1}^{(n+1)} - \psi_{i-1,j,k}^{(n+1)} \right]. \quad (46)$$

Derivatives associated with the ϕ_{St} terms are approximated by upwind differences; a representative form is

$$\Delta t (f\phi_{\xi t}) = \frac{1}{2\Delta\xi} (f_{i,j,k} + f_{i-1,j,k}) \left[(\phi_{i,j,k}^{(n+1)} - \phi_{i,j,k}^{(n)}) - (\phi_{i-1,j,k}^{(n+1)} - \phi_{i-1,j,k}^{(n)}) \right]. \quad (47)$$

If u , v , or w is negative, Equations (45) - (47) are revised to reflect the appropriate upwind direction. All second-derivative contributions to the term $(\Delta\phi - \phi_{SS})$ in Equation (41) are central differenced in a manner similar to Equations (43) and (44).

4.2 Boundary Conditions

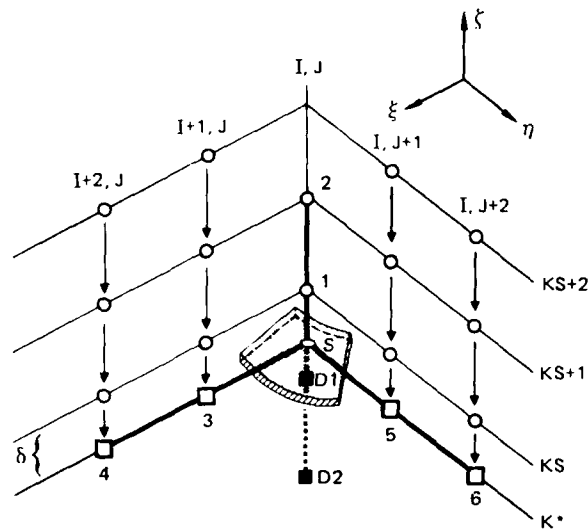
Since the present formulation produces a finite computational domain (Figure 5), application of far-field conditions is straight-forward on those faces that represent infinity and on which $\phi = 0$. The boundary condition in the Trefftz plane (downstream infinity face) depends on the circulation distribution and is not known a priori; therefore, it must be calculated by an embedded relaxation procedure which solves the two-dimensional problem for the downwash field. This calculation is based on Equation (25) with boundary conditions (26a) - (26c); central-difference approximations are used similar to those at subsonic grid points. The Trefftz-plane calculation coincides with updates of the circulation distribution. Finite differences in the ζ -

direction that traverse the vortex sheet are adjusted to incorporate the potential jump across it. At the wing/body symmetry plane, where $v = 0$, Equation (23b) is used to compute values of the potential function at image points located one grid-step beyond the computational domain. In order to compensate for coordinate nonorthogonality at this plane, derivatives are replaced by one-sided differences that are oriented to correspond with the skewed intersection of spanwise grid lines. The present symmetry-plane scheme is an extension of one used by Boppe⁴ and has proved to be numerically stable.

At the wing/body surface, special treatment is required to enforce the flow tangency condition given by Equation (24). A scheme is used that is based on ideas derived from previous numerical applications of non-surface-fitted coordinates.^{11,12} The Neumann form in Equation (24) is replaced by an equivalent Dirichlet condition that is applied at image points below the wing/body surface. Referring to Figure 6, point S is a typical surface control point defined by the intersection of a vertical grid line. Using the corner-shaped boundary-point array shown, one-sided differences based on the Lagrange interpolation formula for three unevenly spaced points are substituted into Equation (24) to obtain a Dirichlet value for the potential function at point S. Extrapolation along line S-1-2 then transfers this value to the uniformly spaced image point D1 located below the surface boundary. Thus,

$$\phi(D1) = \mathcal{F} \left[\mathcal{S}'_x(S), \mathcal{S}'_y(S), \phi_1 \dots \phi_6, \Delta\xi, \Delta\eta, \Delta\zeta, \delta \right] , \quad (48)$$

where $\mathcal{S}'_x(S)$, $\mathcal{S}'_y(S)$ are known surface slopes, $\phi_1 \dots \phi_6$ are potential values at indicated points of the boundary array associated with point S, $(\Delta\xi, \Delta\eta, \Delta\zeta)$ are grid stepsizes in the respective coordinate directions, and δ is the offset distance between point S and the grid point above it. Repeated applications of the Lagrange formula to the sets of points connected by arrows in Figure 6 provide the inter-grid values $\phi_3 \dots \phi_6$. The potential-function value at image point D1 fixes the boundary condition during relaxation of the solution at exterior points on the vertical line D1-2. After each relaxation sweep of the exterior field, the image-point potential value is recomputed. In the event that upwind differencing at a surface-adjacent point necessitates a second



- S Surface point at which Neuman flow-tangency condition is enforced
- 1-6 Exterior points used to calculate equivalent Dirichlet potential at surface point S
- D1, D2 Interior image points used in numerical solution scheme
- δ Grid-point offset distance

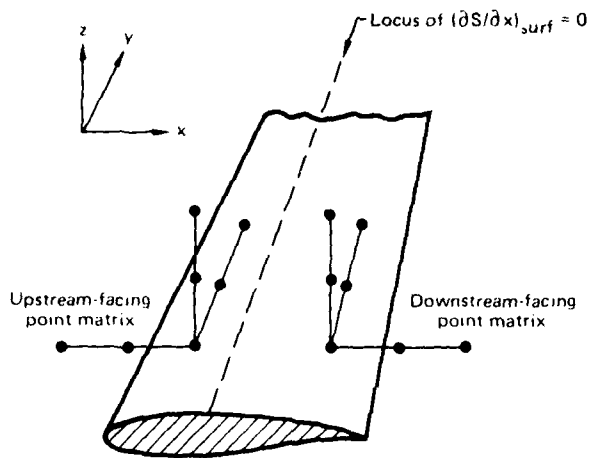
GP03-1190-6

Figure 6. Grid-point array used to numerically enforce the boundary condition at a typical wing/body surface point.

image point D2, the associated potential value is obtained by linear extrapolation in the vertical direction.

In order to treat each control point consistently, the corresponding boundary-point array is usually directed to the exterior side of the wing/body surface. Thus, a reorientation of the array occurs at a locus of zero streamwise slope on a convex part of the surface, as shown in Figure 7. This arrangement has the effect of eliminating direct coupling between neighboring image points in the present methodology. The boundary-point array is not reoriented in concave surface regions, however.

Additionally, it has been found useful to permit surface image points to be multi-valued in order to more accurately represent the surface condition at configuration edges. Figure 8 indicates schematically the situation in a wing-section plane. Image point P represents the leading-edge condition during relaxation of the solution on the line ILE-1 upstream of the section. Then, for computation on the segment of line ILE below the section, point P represents a lower-surface condition. In the same way, point P can also be



GP03-1190-17

Figure 7. Arrangement of boundary-point arrays relative to locus of zero chordwise slope.

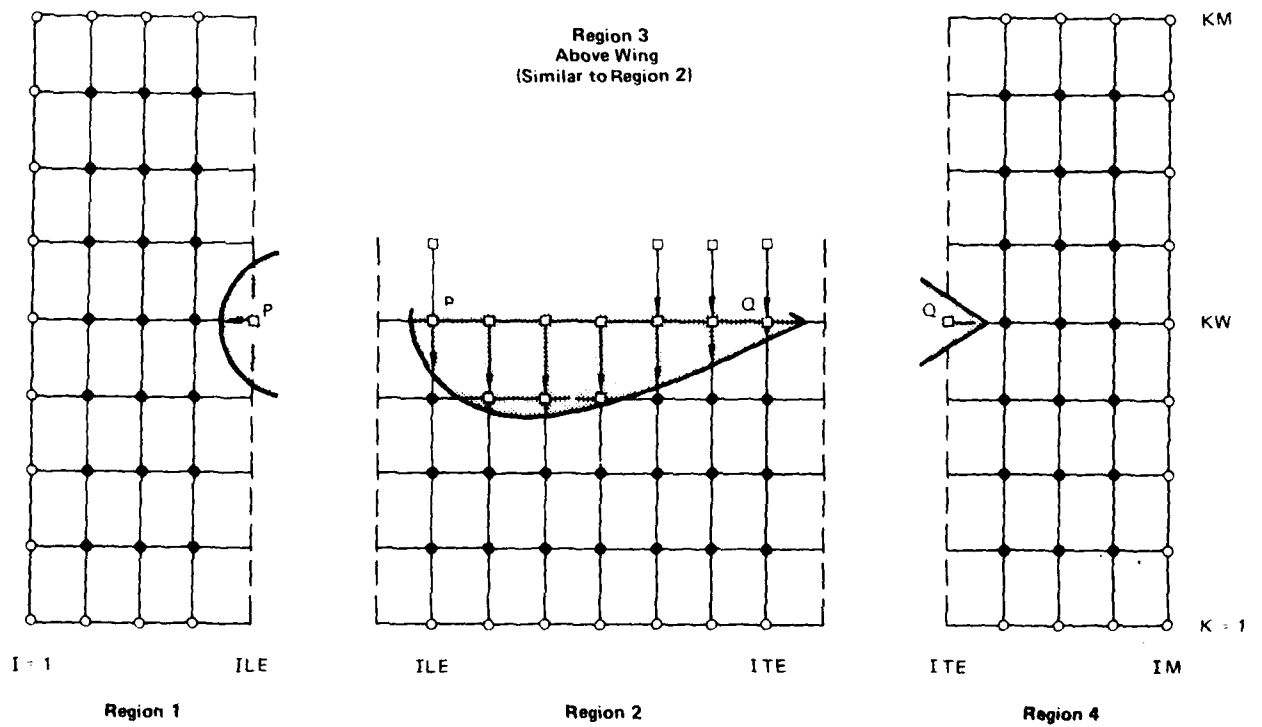


Figure 8. Computation schematic: wing-section plane. Illustrates multi-valued dummy points.

GP03-1190-18

assigned an upper-surface boundary value. Image point Q at the section trailing edge is treated similarly. Other such multi-valued image points occur at the fuselage nose, along the side-edge locus of the fuselage, and at the wing tip.

4.3 Computation Procedure

The system of algebraic difference equations for the wing/body problem is solved iteratively by systemically sweeping through the computational space (Figure 5) and relaxing the solution along vertical columns of grid points. In the present methodology, the domain is swept by crossplanes beginning at the upstream-infinity boundary and proceeding to the downstream-infinity boundary. Each crossplane is swept by vertical lines from the configuration symmetry plane to the spanwise-infinity face of the domain. In a crossplane that intersects the wing/body surface, all column segments beneath the configuration are relaxed first, followed by column segments above the geometry, and then by columns in the outboard region of the crossplane. The relaxation procedure is continued until either a prescribed convergence criterion is satisfied or a specified number of domain sweeps have been completed.

During a particular iterative cycle, the first step is to fix the surface-boundary condition by calculating image-point potential values associated with each surface control point. The solution is then relaxed throughout the domain as described above. At designated intervals, the circulation distribution is updated by applying Equation (26b) at the wing trailing edge. Following each circulation update, the Trefftz-plane boundary condition is recalculated by an embedded relaxation of Equations (25) - (26) for the downwash field. The cycle is then repeated.

During the sweep process, image-point values for the wing/body surface condition are substituted sequentially, as required, into the solution array that stores the potential function. This procedure simplifies program logic by effectively eliminating the distinction between surface-adjacent grid points and interior field points in computing finite differences. On vertical grid lines that intersect the wing/body configuration, grid benchmarks for the surface-adjacent points are used to designate the length of column-segments on which the solution is relaxed.

5.0 RESULTS

Representative calculations have been made for a configuration comprised of ONERA Wing M6 attached at mid-height to a hemisphere-cylinder fuselage. Details of the wing geometry are given in References 18 and 19. The wing has a planform with a leading-edge sweep angle of 30° , a taper ratio of 0.56, and a uniform section whose thickness ratio is 0.098. The fuselage radius is taken to be 25% of the exposed wing semispan. A representation of the configuration planform is shown in Figures 9 and 10.

Figures 9 and 10 also show the calculated pressure distributions at several spanwise stations on the configuration for a freestream Mach number of 0.84 and angles of attack of 0° and 3.06° , respectively. The sharp peaks in the wing leading-edge region are the consequence of grid coarseness (see below). There is an indication of a double-shock structure in the midspan

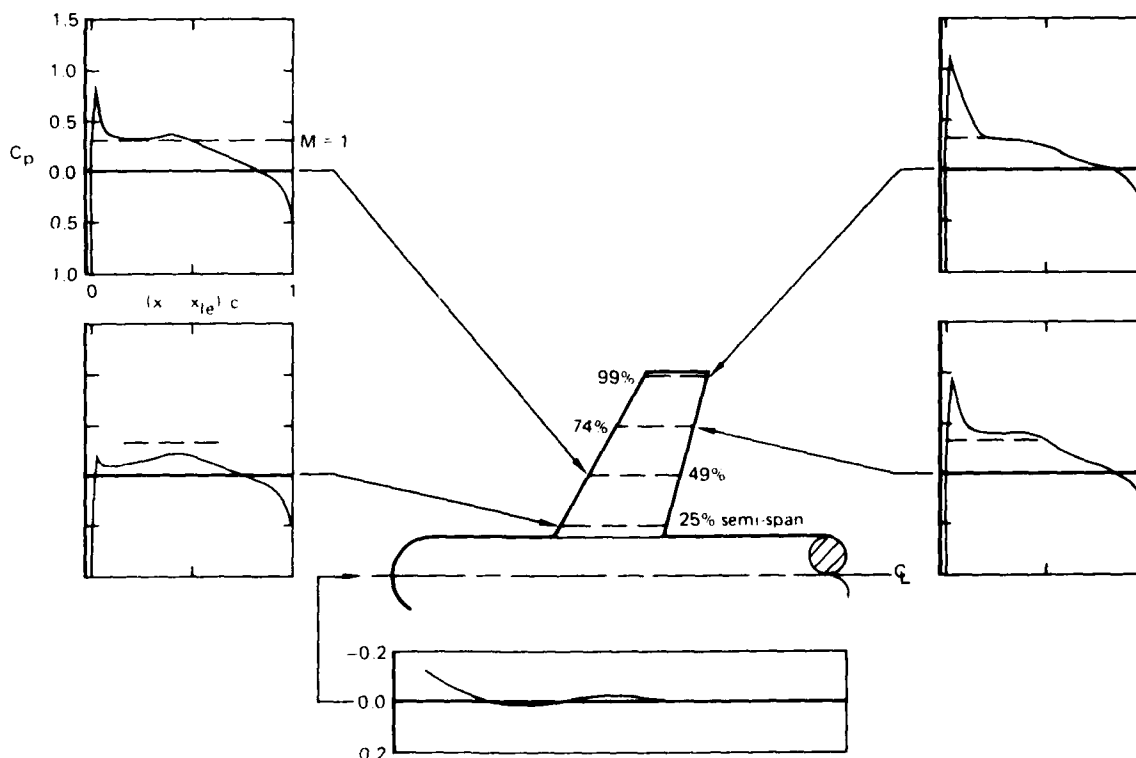
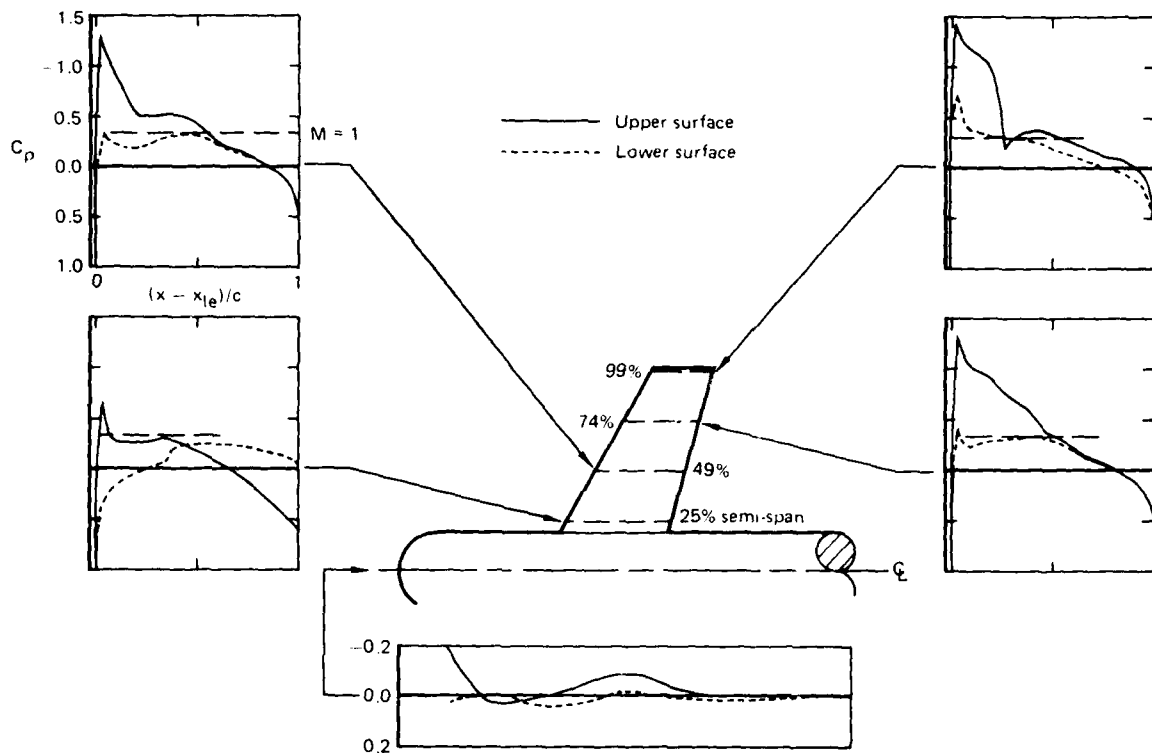


Figure 9. Calculated surface-pressure distributions on a configuration composed of ONERA Wing M6 attached at mid-height to a hemisphere-cylinder fuselage: $M_\infty = 0.84$, $\alpha = 0^\circ$. Grid dimension: $49 \times 19 \times 25$. Every third spanwise station is shown.



GP11 0527.4

Figure 10. Calculated surface-pressure distributions on a configuration composed of ONERA Wing M6 attached at mid-height to a hemisphere-cylinder fuselage: $M_\infty = 0.84$, $\alpha = 3.06^\circ$. Grid dimension: $49 \times 19 \times 25$. Every third spanwise station is shown.

region of the wing, which coalesces to a single strong shock near the wing tip; this effect is particularly evident for the lifting case in Figure 10. In both cases, the pressure distribution at the fuselage centerline is modified by the presence of the wing, again with a more pronounced effect in the lifting case.

In order to assess solution accuracy, the calculations are compared with available experimental data for ONERA Wing M6 obtained in a wing-alone test (References 18 and 19). Since spanwise grid stations and test stations on the wing do not coincide, the computed results are interpolated linearly along constant-percentage-chord lines to the positions of the data measurements. The comparisons are shown in Figures 11 and 12. In general, agreement is reasonably good especially near the wingtip where the presence of the fuselage in the calculations has the least effect. Differences near the wing leading edge and the poor resolution of the double-shock structure along the wing can be attributed to grid coarseness in the computed results, while trailing-edge discrepancies are more probably the consequence of viscous effects in the data. (Note the definite shock-induced separation exhibited by the test data at the wingtip station in Figure 12.) Anomalous behavior such as the

inflection at the downstream end of the shock wave at $y/b = 0.95$ in Figure 12 is probably caused by the interpolation procedure; it is not evident in the actual computation results.

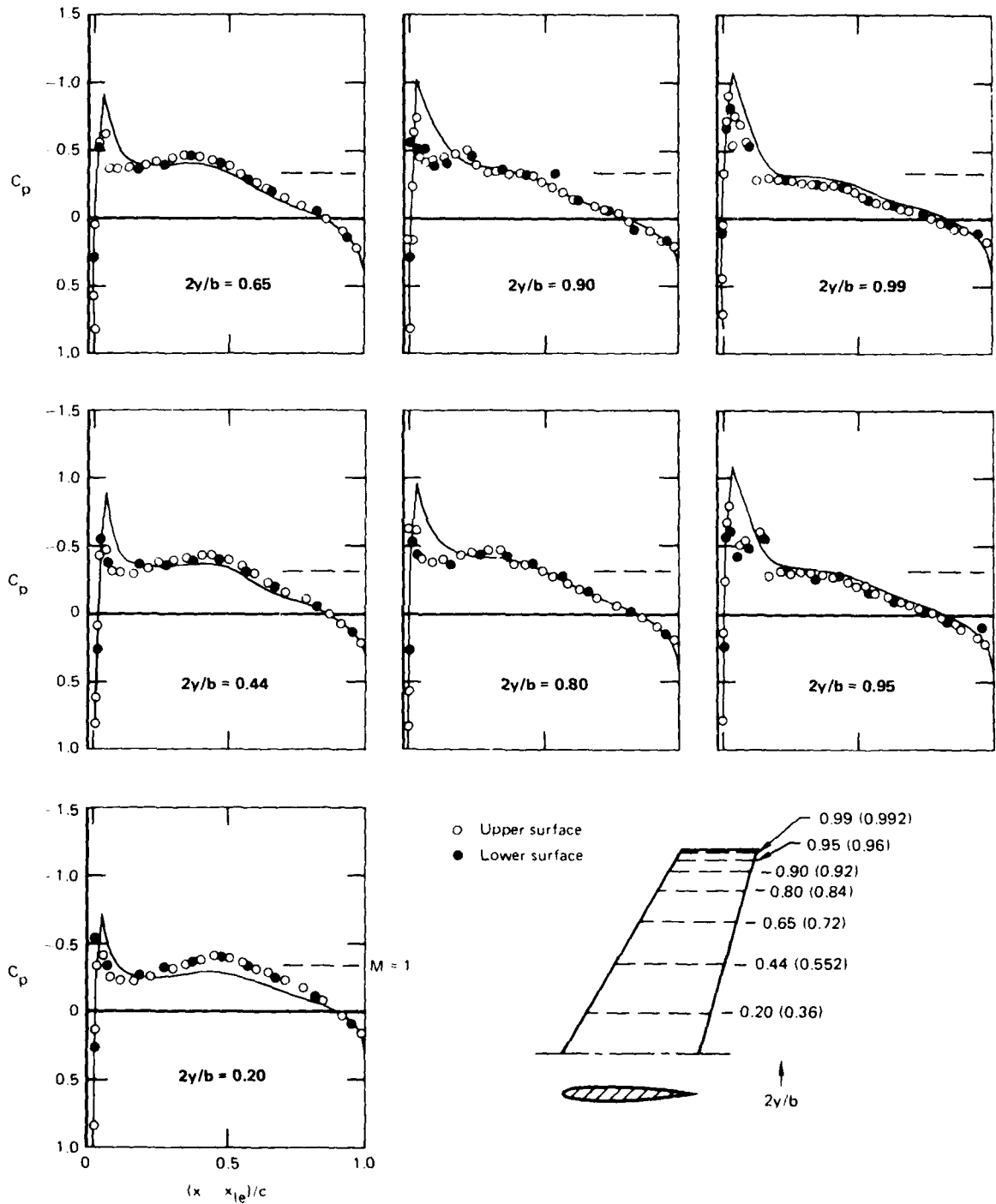
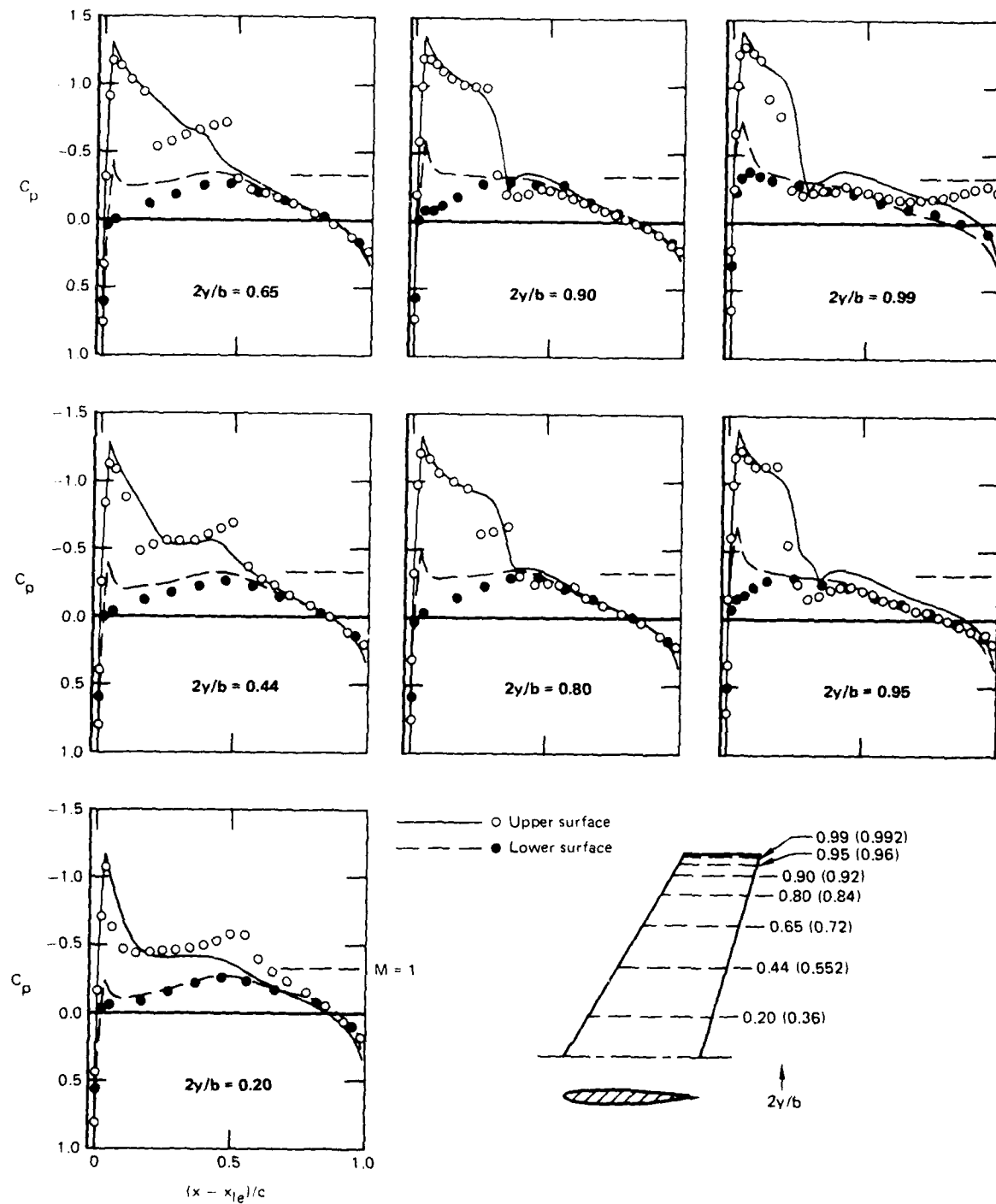


Figure 11. Comparison of wing/body calculations at $M_\infty = 0.84$, $\alpha = 0^\circ$ with wing-alone experimental data for ONERA Wing M6 at $M_\infty = 0.8399$, $\alpha = 0.04^\circ$. Data are from Reference 19. Parenthetic numbers denote corresponding spanwise positions on the wing/body configuration.

GP11 0527 2



GP11-0527-1

Figure 12. Comparison of wing/body calculations at $M_\infty = 0.84$, $\alpha = 3.06^\circ$ with wing-alone experimental data for ONERA Wing M6 at $M_\infty = 0.8395$, $\alpha = 3.06^\circ$. Data are from Reference 19. Parenthetic numbers denote corresponding spanwise positions on the wing/body configurations.

The calculations for these cases were carried out on a 49 x 19 x 25 grid in the chordwise, spanwise, and vertical directions, respectively. Thirty-three vertical grid lines intersect each wing section chord, thus providing sixty-six (upper plus lower) surface control points on each section profile. Control points are spaced at approximately every 3% of section chord with an edge-offset of 0.5% chord at the leading- and trailing-edge points. The pressure peaks in Figures 9 and 10 correspond to the second chordwise control point from the leading edge on each section. In the spanwise direction, 13 grid stations occur on the configuration semispan.

The calculations were performed on a Control Data CYBER 175 computer with FTN(OPT=2) compiler. The nonlifting case of Figure 9 required 2.10 min of CPU execution time, and the lifting case of Figure 10 required 5.23 min. Convergence is based on a 10^{-4} cut-off limit on the maximum correction to the potential function over the solution domain between the final two iterative sweeps. Decreasing the cut-off limit to 10^{-5} was found to approximately double solution times.

Table 1 shows the accuracy of the scheme used to numerically apply the surface boundary condition. At two semispan positions on the wing, velocity slopes computed from a converged solution are compared with prescribed streamwise surface slopes. Agreement between corresponding slope values extends to at least the third decimal place, one order-of-magnitude greater than the cut-off limit on the potential function.

In order to explore the stability of the computer program, additional calculations were performed for a configuration with the planform shown in Figure 4 in which the wing has a uniform NACA 0012 section. Convergent operation without the use of damping augmentation was achieved in nonlifting cases for freestream Mach numbers up to 0.99 and at selected (supercritical) Mach numbers for angles of attack up to 6° . These results are not included in this report since no comparison data exist for the particular geometry.

TABLE 1. COMPARISON OF SURFACE-SLOPE VALUES AT TWO SEMISPAN STATIONS.
 DZDX = PRESCRIBED SLOPE; SLOPE = CALCULATED VELOCITY SLOPE.

(X-XLE)/C	2Y/B =	.2457	2Y/B =	.9900
	SLOPE U	DZDXU	SLOPE U	DZDXU
.0050	1.191182	1.192676	1.192739	1.192676
.0364	.199445	.199572	.199583	.199572
.0677	.137938	.138035	.138046	.138035
.0989	.109128	.109208	.109222	.109208
.1300	.087512	.087580	.087594	.087580
.1611	.071034	.071092	.071107	.071092
.1921	.058087	.058136	.058151	.058136
.2230	.047324	.047365	.047380	.047365
.2539	.037729	.037764	.037777	.037764
.2848	.028551	.028580	.028593	.028580
.3156	.019317	.019339	.019351	.019339
.3464	.009725	.009742	.009751	.009742
.3771	-.000267	-.000277	-.000303	-.000277
.4079	-.010586	-.010561	-.010520	-.010561
.4386	-.021286	-.021235	-.021118	-.021235
.4693	-.031790	-.031724	-.031526	-.031724
.5000	-.041930	-.041359	-.041578	-.041959
.5307	-.051393	-.051328	-.050765	-.051328
.5614	-.059962	-.059711	-.059473	-.059911
.5921	-.067439	-.067410	-.066705	-.067410
.6229	-.073775	-.073773	-.073210	-.073773
.6536	-.079030	-.079056	-.078448	-.079056
.6844	-.083393	-.083447	-.082903	-.083447
.7152	-.087188	-.087269	-.086598	-.087269
.7461	-.090812	-.090918	-.090227	-.090719
.7770	-.094731	-.094861	-.094151	-.094861
.8079	-.099350	-.099501	-.098771	-.099501
.8389	-.104321	-.105092	-.104338	-.105092
.8700	-.111292	-.111480	-.110598	-.111480
.9011	-.117739	-.117940	-.117130	-.117940
.9323	-.122597	-.122804	-.121977	-.122804
.9636	-.123584	-.123781	-.122973	-.123781
.9950	-.123582	-.123776	-.122919	-.123776

(a) Convergence limit on $\Delta\phi$: 10^{-4}

GP11 0527 5

(b) Nonlifting calculation for ONERA Wing M6/hemisphere-cylinder fuselage

6.0 CONCLUDING REMARKS

The present work substantiates the use of non-surface-fitted coordinates for numerical computation of transonic wing/body flowfields. However, a number of modifications and improvements remain to be incorporated in order to make the present computer program useful for engineering applications.

The capability of the program to perform calculations on a series of progressively finer grids needs to be completed in order to improve computing efficiency. This modification involves two requirements. Verification of the grid-halving subroutine in the program must be completed, and the array that stores the potential function must be moved to disk storage, with a sequential transfer into central memory of only those array segments required at each stage of the computation process.

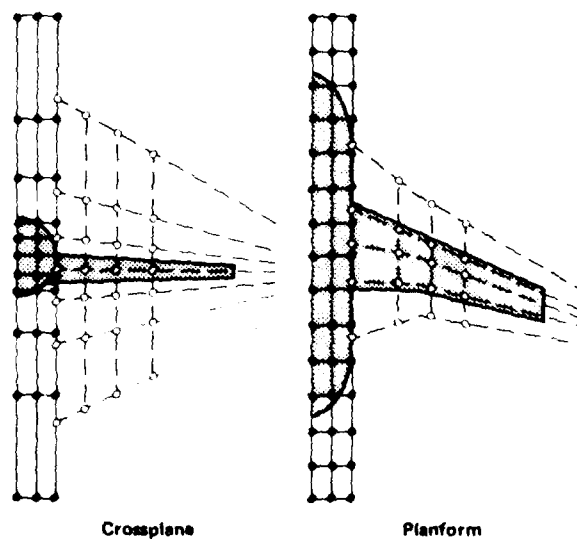
The chordwise stretching function used between the leading and trailing edges of the wing is symmetric about the mid-chord locus. In order to improve resolution of the blunt leading-edge region within a fixed grid dimension, the introduction of an asymmetric stretching that clusters more points near the leading edge than the trailing edge would be helpful. This modification would provide a more efficient means of improving solution accuracy in the leading-edge region than a simple increase in the number of chordwise grid points.

Incorporation into the program of a finite-fuselage capability would provide a better geometry model for engineering applications. An attempt to do this during the present contract was unsuccessful when the resulting version of the code proved to be nonconvergent. The principal difficulty seemed to involve the question of how to properly treat the vortex sheet (circulation distribution) in the region behind the fuselage.

Finally, there exists the problem of improving flowfield and geometry resolution around the fuselage. A method for accomplishing this, depicted in Figure 13, involves a two-coordinate formulation in which the present coordinate arrangement is retained about the wing and better-suited coordinates -- perhaps, a simple stretched, cartesian system -- are introduced in a vertical slab whose width coincides with that of the fuselage. This scheme would require interpolation within an overlap region (possibly, only a

single plane) common to both coordinate systems. In order to conveniently accommodate the two-coordinate formulation, it may be preferable to revise the computation strategy such that the domain is swept by wing-section planes beginning at spanwise infinity and moving to the wing/body centerline. The vertical-line sweep in each section plane would begin at the upstream boundary and proceed to the downstream boundary.

The baseline computer program described in Appendix A, when modified as discussed above, is expected to provide a framework for treatment of complex wing/body configurations. Further extension to include add-on components such as nacelles, stores, or additional lifting surfaces should be possible also.



GP11 0527 8

Figure 13. Two-grid arrangement for wing/body flow analysis.

ACKNOWLEDGEMENT

The spline interpolation scheme used for fitting of geometry data was developed by Prof. Chaman L. Sabharwal, Department of Mathematics, St. Louis University, under a concurrent program funded by National Science Foundation Faculty Research Participation Grant No. SPI-7907397.

REFERENCES

1. W. F. Ballhaus and F. R. Bailey, Numerical Calculation of Transonic Flow About Swept Wings, AIAA Paper No. 72-677, June 1972.
2. F. R. Bailey and W. F. Ballhaus, Relaxation Methods for Transonic Flow About Wing-Cylinder Combinations and Lifting Swept Wings, Lecture Notes in Physics, (Springer-Verlag, 1972) Vol. 19.
3. W. F. Ballhaus, F. R. Bailey and J. Frick, Improved Computational Treatment of Transonic Flow About Swept Wings, NASA CP-2001, November 1976.
4. C. W. Boppe, Calculation of Transonic Wing Flows by Grid Embedding, AIAA Paper No. 77-207, January 1977.
5. C. W. Boppe, Computational Transonic Flow About Realistic Aircraft Configuration, AIAA Paper No. 78-104, January 1978.
6. C. W. Boppe and M. A. Stern, Simulated Transonic Flows for Aircraft with Nacelles, Pylons, and Winglets, AIAA Paper No. 80-0130, January 1980.
7. A. Jameson, Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1, Comm. Pure and Appl. Math. 27, 283 (1974).
8. A. Jameson and D. A. Caughey, Numerical Calculation of the Transonic Flow Past a Swept Wing, NYU Report C00-3077-140, June 1977.
9. A. Jameson and D. A. Caughey, A Finite Volume Method for Transonic Potential Flow Calculations, AIAA Paper No. 77-635, June 1977.
10. P. A. Henne and R. M. Hicks, Transonic Wing Analysis Using Advanced Computational Methods, AIAA Paper No. 78-105, January 1978.
11. R. Parker and C. Y. Ma, Normal Gradient Boundary Condition in Finite Difference Calculations, Intern. J. Numer. Meth. in Engrg. 1, 395 (1976).
12. L. A. Carlson, Transonic Airfoil Design and Analysis Using Cartesian Coordinates, AIAA J. 13, 349 (1976).
13. T. A. Reyhner, Cartesian Mesh Solution for Axisymmetric Transonic Potential Flow Around Inlets, AIAA J. 15, 624 (1977).
14. T. A. Reyhner, Transonic Potential Flow Around Axisymmetric Inlets and Bodies at Angle of Attack, AIAA J. 15, 1299 (1977).

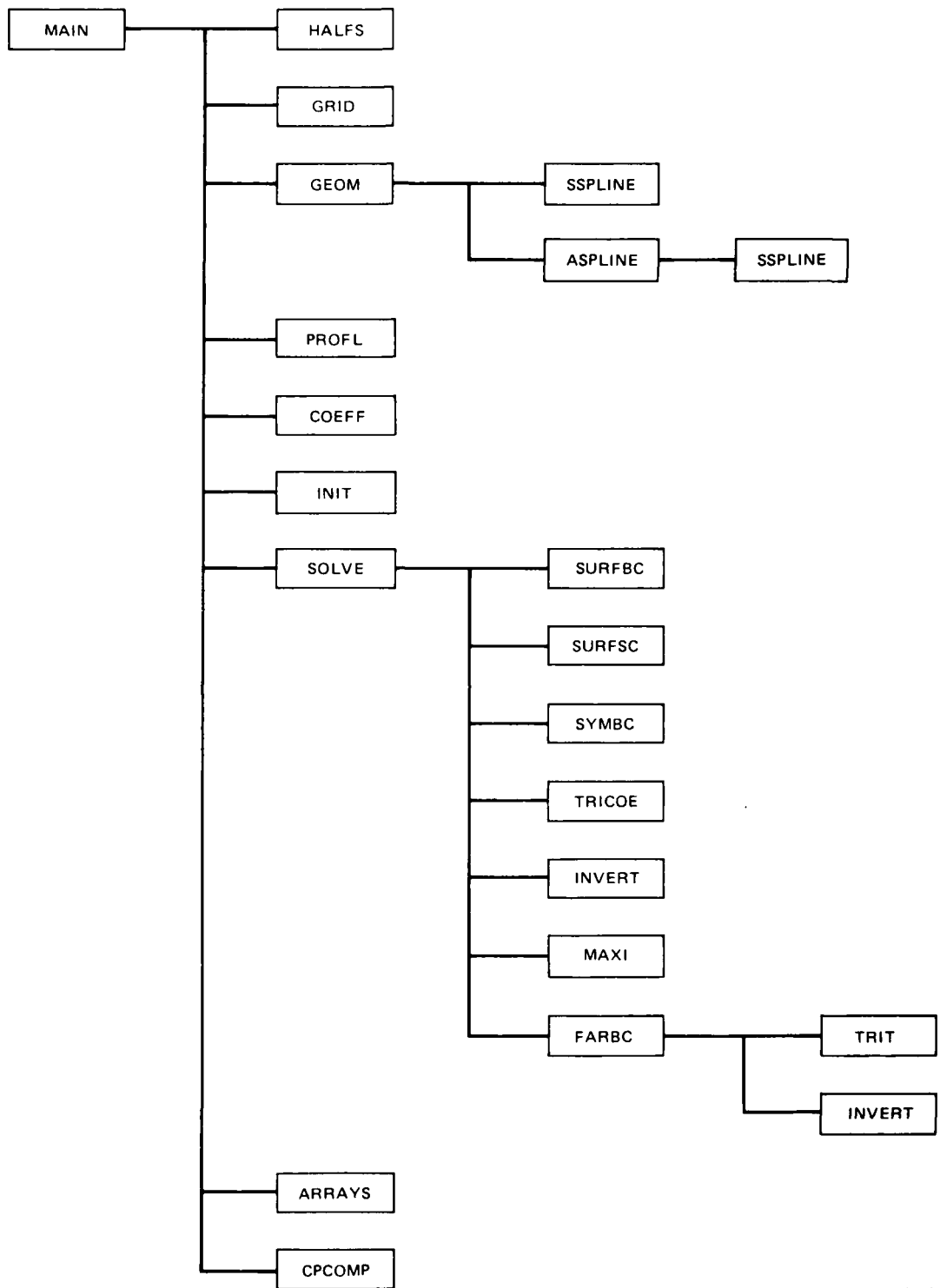
15. W. J. Rae and G. F. Homicz, A Rectangular-Coordinate Method for Calculating Nonlinear Transonic Potential Flowfields in Compressor Cascades, AIAA Paper No. 78-248, 1978.
16. E. M. Murman and J. D. Cole, Calculation of Plane Steady Transonic Flows, AIAA J. 9, 114 (1971).
17. J. C. South, Jr. and A. Jameson, Relaxation Solutions for Inviscid Axisymmetric Transonic Flow Over Blunt or Pointed Bodies, AIAA Computational Fluid Dynamics Conf., July 1973.
18. B. Monnerie and F. Charpin, Essais de Buffeting d'une Aile en Fleche en Transsonique, 10^e Colloque d'Aerodynamique Appliquee, Lille, France, November 1973.
19. V. Schmitt and F. Charpin, Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers, AGARD-AR-138, May 1979.

APPENDIX A. USER'S GUIDE TO COMPUTER PROGRAM

A computer program based on the formulation presented in the main body of this report is listed in Appendix B. The subroutine structure of the program is shown in Figure A1, and a summary of subroutine functions is given in Table A1. The code has been run on a Control Data CYBER 175 computer with an FTN compiler. As dimensioned, it requires 260k (octal) storage locations to load and execute. The present form of the program does not use peripheral storage devices.

The solutions presented in this report have been computed on a single grid. Some logic is contained in the code to permit a calculation to be made on a series of progressively finer grids, but it has not been fully implemented. In particular, subroutine HALFS interpolates the potential array $P(I,J,K)$ from an initial grid onto one that is half-spaced in each coordinate direction of the computational domain. The interpolated array is then used as the starting solution for continuation of the relaxation process on the new grid. Input parameter NHALF specifies the number of grid-halving cycles to be performed. Full implementation of this capability will require that the array $P(I,J,K)$ be transferred to disk storage and that provision be made for a sequential, plane-by-plane transfer of $P(I,J,K)$ values into central memory to coincide with each relaxation sweep through the computational domain.

Figure 1 defines the class of wing/body geometries that can be represented by the program. Input data are smoothed to ensure that the wing leading and trailing edges are piecewise straight lines. One break (kink) is permitted in each edge but need not be present. If both leading- and trailing-edge breaks occur, they may be at different spanwise positions. Although the formulation outlined in Section 2.0 places no restriction on wing attachment to the fuselage, code logic assumes that the horizontal mid-plane of the grid ($\zeta = 0$) coincides with the maximum width position on the fuselage. Combined with the specification of circular fuselage cross-sections, this computational arrangement effectively limits application to mid-wing configurations at present. This restriction can be removed by incorporating a two-coordinate formulation as discussed in Section 5.0.



GP03-1190-13

Figure A1. Subroutine structure of the wing/body program.

TABLE A1. SUBPROGRAM LIST

NAME	FUNCTION
MAIN	Reads input data; controls overall program logic
HALFS	Interpolates solution arrays onto a half-spaced grid. (Has not been verified.)
GRID	Defines computational grid; calculates wing configuration data at spanwise grid stations
GEOM	Calculates wing/body coordinates and slopes at planform grid stations in the physical domain
ASPLINE	Parameterizes input geometry data in terms of arc length along the curve prior to spline interpolation
SSPLINE	Interpolates input geometry data
PROFL	Calculates geometry-defining quantities in the computational domain
COEFF	Calculates fixed quantities which appear in the finite-difference equations
INIT	Initializes solution arrays
SOLVE	Executes one relaxation sweep of the computation domain; updates the circulation distribution and solution boundary values
FARBC	Computes the Trefftz-plane boundary condition
SURFBC	Calculates boundary values at control points on the wing/body surface
SURFSC	(Entry) calculates boundary values at side-edge points of the configuration
SYMBC	(Entry) Calculates image values at the wing/body symmetry plane
TRICOE	Calculates coefficients of the finite-difference potential equation at grid points along a specified vertical grid-line segment
TRIT	(Entry) calculates coefficients of the finite-difference downwash equation in the Trefftz plane
INVERT	Solves the finite-difference potential equation along a vertical grid-line segment to obtain updated solution values
MAXI	Determines the maximum potential-value increment along a specified vertical grid-line segment between the current and preceding relaxation sweeps
ARRAYS	Prints out solution arrays to specified iteration intervals: circulation distribution, potential distribution, surface boundary values and associated image-point values, symmetry-plane image-point values. (Used mainly for diagnostic purposes.)
CPCOMP	Calculates and writes the pressure-coefficient distribution on the wing/body surface

GP03-1190-1

Table A2 summarizes the sequence and format of input data required by the program. Data categories are as follows:

- case title,
- computational grid parameters,
- code execution parameters,
- case specification (Mach number, angle of attack),
- fuselage configuration data, and
- wing configuration data.

Suggested values for grid and execution parameters are included in the table. Also given are parameter-value limits imposed by current code dimensions.

TABLE A2. GLOSSARY OF INPUT DATA.

CARD	COLUMNS	VARIABLE	EXPLANATION	
1	1-80	TITLE	Case title (written to output)	
2	1-10	NXI	Number of chordwise grid steps at start of calculation. Maximum: 48	
	11-20	A1	X - ξ stretching constant in Eq. (3b). Suggested value: 0.16	
	21-30	A2	X - ξ stretching constant in Eq. (3b). Suggested value: 2.75	
	31-40	XIO	Stretching transition point of ξ - coordinate; see Figure 2. Ratio XIO/(1 + XIO) determines fraction of chordwise grid steps which occur on each wing-section chord	
3	41-50	XCAPO	Stretching transition point of X-coordinate; see Figure 2. Must be inside wing-section edge. Suggested value: 0.495	
	1-10	NETA	Number of spanwise grid steps at start of calculation. Maximum: 18	
	11-20	B1	Y - η stretching constant in Eq. (5b). Suggested value: 0.16	
	21-30	B2	Y - η stretching constant in Eq. (5b). Suggested value: 2.75	
4	31-40	ETAO	Stretching transition point of η - coordinate; see Figure 2. Ratio ETAO/(1 + ETAO) determines fraction of spanwise grid steps which occur between the wing/body symmetry plane and the wingtip.	
	41-50	YCAPO	Stretching transition point of Y-coordinate; see Figure 2. Must be inside wingtip. Suggested value: 0.49995	
	1-10	NZETA	Number of grid steps in vertical direction at start of calculation. Maximum: 24	
5	11-20	C1	Z - ζ stretching constant in Eq. (7). Suggested value: 0.45	
	1-10	ITERM	Maximum number of iterations to be executed on the initial grid	
	11-20	NHALF	Number of grid-halving cycles. Set NHALF = 0 for single-grid calculation. (Note: Subroutine HALFS has not been fully verified.)	
6	21-30	NPRINT	Iteration frequency for execution of subroutine ARRAYS, which prints complete solution arrays for diagnostic purposes. If NPRINT = 0, only the circulation distribution is printed upon completion of the relaxation procedure. If NPRINT > ITERM, complete solution arrays are printed after the relaxation process.	
	1-10	WE	Relaxation parameter for potential at elliptic field points. Suggested value: 1.70	
	11-20	WG	Relaxation parameter for circulation. Suggested value: 1.00	
	21-30	DPLIM	Convergence cut-off limit for maximum potential-value increment between successive iterations. Suggested value: 10^{-4}	
7	31-40	EPSI	Damping factor in potential difference equation used at hyperbolic field points. Suggested value: 0.00, increase if instability occurs	
	1-10	ZMACH	Freestream Mach number	
	11-20	ALPHA	Angle of attack of the wing reference plane (in degrees)	
8	1-10	NF	Number of fuselage coordinate sets to be read from the following cards (one XF, ZF, RF set per card)	
	1-NF	1-10	XF	x coordinate along fuselage beginning at nose; see Figure 1
	11-20	ZF	z coordinate of fuselage side-profile reference line; see Figure 1	
9	21-30	RF	Fuselage crossplane radius; see Figure 1	
	1-10	JSECT	Number of wing-section data sets to be read subsequently. Maximum: 5	
	11-20	JBL	Sequence number of wing data set at break in leading edge. If no break exists, set JBL = 0.	
10	21-30	JBT	Sequence number of wing data set at break in trailing edge. If no break exists, set JBT = 0.	

The first card uses alphanumeric format 20A4. All remaining cards use repeated floating-point format 8F10.5. Conversion of data to integer mode is performed within the program as required.

GP11 0527

TABLE A2. (CONTINUED) GLOSSARY OF INPUT DATA

CARD	COLUMNS	VARIABLE	EXPLANATION
10	1-10	YS	Spanwise station of wing section
	11-20	XLES	x-coordinate of wing-section leading edge
	21-30	ZLES	z-coordinate of wing-section leading edge
	31-40	CS	Wing-section chord length
	41-50	ATS	Wing-section twist angle relative to wing reference plane (in degrees)
	51-60	TS	Wing-section thickness-to-chord ratio
	61-70	FS	Repeat indicator. If FS = 0, wing-section coordinate data from the previous span station is used; the next card specifies parameters for the wing section at the next span station (go to Card 12). If FS = 1, coordinates for a new section profile are read from the following data cards.
11	1-10	NSU	Number of wing-section upper-surface coordinate sets to be read from the following cards (one XSU, ZSU set per card)
	11-20	NSL	Number of wing-section lower-surface coordinate sets to be read from the following cards (one XSL, ZSL set per card)
	21-30	KSYM	Wing-section symmetry indicator. If KSYM = 0, the section is asymmetric; both upper- and lower-surface coordinates must be given. If KSYM = 1, the section is symmetric, and only the upper-surface coordinates are required.
1-NSU	1-10	XSU	Upper-surface coordinates of wing section from leading to trailing edge. One set per card
	11-20	ZSU	
1-NSL	1-10	XSL	Lower-surface coordinates of wing section from leading to trailing edge. One set per card. (Required only if KSYM = 0)
	11-20	ZSL	
12	—	—	Repeat of Data Card 10 for the next wing section
13	—	—	Repeat of Data Card 11 and data sets 1-NSU and 1-NSL for the wing section defined by Data Card 12. (Required only if FS = 1 on Card 12)
14-15 } 16-17 } 18-19 }	— — —	— — —	Up to three additional wing-section definition cards and coordinate data sets. Total number of data sets must correspond to JSECT on Data Card 9.

GP03-1190-21

Fuselage data consist of sets of side-profile reference-line coordinates and crossplane radii given at streamwise stations beginning at the nose and proceeding aft. These data are spline interpolated to obtain required values at grid-point stations. Along a constant-radius segment of the fuselage, a number of values must be given to ensure accuracy of the spline fit, and the data must be more closely spaced near the ends of the segment than in its central region. A final table value at streamwise location $XF > 25$ is required to specify the semi-infinite fuselage length.

Wing data are read section-by-section beginning at the wing root and moving outboard to the wingtip. In the data set for each section, the first card specifies section properties: spanwise position, leading-edge coordinates, chord length, twist angle relative to a horizontal reference

line, thickness-to-chord ratio, and an indicator flag to designate either that new profile coordinates are to be read or that the previous section profile is to be repeated. The next card specifies the numbers of coordinate pairs in the upper- and lower-surface data tables for the section and an indicator flag to designate whether the section is symmetric or asymmetric. Next, if a new section profile is being given, the upper-surface coordinate pairs of the profile are read proceeding from leading edge to trailing edge, followed by the lower-surface coordinate pairs in the same order. If the profile at the new section is identical to that of the previous section, surface-coordinate data are read internally by the program and need not be repeated in the input data list. Also, if a section profile is symmetric, only the upper-surface coordinates are required in the input data set; the lower-surface coordinates are set within the program.

The code provides the following tabular output:

- input data listing,
- computational coordinates, stretching derivatives and grid benchmark values,
- wing configuration data at grid stations,
- maps of surface-adjacent grid points (upper surface, lower surface, side edge in planform view),
- iteration summary,
- final circulation distribution,
- detailed solution arrays (if requested), and
- pressure-coefficient and surface-slope distributions.

The input listing, coordinate tables, and wing data provide a check of problem set-up. The grid-point maps define the wing/body configuration in the computational domain. The iteration summary printed after each complete relaxation cycle lists the value and location of the maximum potential increment between the current and previous iterations, the number of supersonic points detected, the current wing-root value of circulation, and information about the embedded iteration process required to update the Trefftz-plane boundary condition. The solution arrays, when requested via input parameter NPRINT, include the complete potential array as well as control-point and image-point potential values which arise in the numerical application of the surface boundary condition. These array print-outs are

useful mainly for code diagnostic purposes. Finally, in addition to the pressure-coefficient distribution, both prescribed surface slopes and velocity slopes calculated from the converged potential field are printed to provide an accuracy check on the surface boundary conditions.

APPENDIX B. LISTING OF COMPUTER PROGRAM

```

PROGRAM TWB3(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C
C
C
C
SOLVES THE FULL POTENTIAL EQUATION FOR TRANSONIC FLOW PAST A
GENERAL-WING/SEMI-INFINITE-FUSELAGE CONFIGURATION.

DIMENSION TITLE(20),XTES(5)
*
* COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,AI,CSA,RX1,RX2,
SNA,TDETA,TOXI,TUZETA,DETA,DXI,ZZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
PLCB2,PLCB3
*
* COMMON/DATAX/ A1,A2,A3,A4,A5,ETA0,NETA,NXI,NZETA,XCAPO,XIO,YCAPO,
NF,XF(150),ZF(150),RF(150),JSECT,JBL,JBT,NSU(5),
* NSL(5),YS(5),XLES(5),ZLES(5),CS(5),ATS(5),TS(5),
FS(5),XSU(150,5),XSL(150,5),ZSU(150,5),ZSL(150,5)
*
* COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERI,
* JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
* PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)

MSTART=0
NCYCLE=0
READ (5,901) (TITLE(N),N=1,20)
READ (5,902) FNXI,A1,A2,XIO,XCAPO
READ (5,902) FNETA,A3,A4,ETA0,YCAPO
READ (5,902) FNZETA,A5
NXI=FNXI
NETA=FNETA
NZETA=FNZETA
READ (5,902) FITERM,FNHALF,FNPRINT
ITERM=FITERM
NHALF=FNHALF
NPRINT=FNPRINT
READ (5,902) WE,WG,DPLIM,EPSI
READ (5,902) ZMACH,ALPHA
WRITE (6,903) (TITLE(N),N=1,20),ZMACH,ALPHA
WRITE (6,904) ITERM,NHALF,NPRINT,WE,WG,DPLIM,EPSI
WRITE (6,905) NXI,A1,A2,XIO,XCAPO,NETA,A3,A4,ETA0,YCAPO,NZETA,A5

C
READ (5,902) FNF
NF=FNF
IF (NF.EQ.0) GO TO 150
DO 110 N=1,NF
110 READ (5,902) XF(N),ZF(N),RF(N)
WRITE (6,911)
DO 120 N=1,NF
120 WRITE (6,912) N,XF(N),ZF(N),RF(N)
150 CONTINUE

C
READ (5,902) FJSECT,FJBL,FJBT
JSECT=FJSECT
JBL=FJBL
JBT=FJBT
J=0
211 J=J+1

```

```

IF (J.GT.JSECT) GO TO 237
READ (5,902) YS(J),XLES(J),ZLES(J),CS(J),ATS(J),TS(J),FS(J)
IF (FS(J).EQ.0.) GO TO 230
READ (5,902) FNSU,FNSL,FKSYM
NSU(J)=FNSU
NSL(J)=FNSL
KSYM=FKSYM
NU=NSU(J)
DO 218 I=1,NU
READ (5,902) XSU(I,J),ZSU(I,J)
218 CONTINUE
NL=NSL(J)
IF (KSYM.EQ.1) GO TO 220
DO 219 I=1,NL
READ (5,902) XSL(I,J),ZSL(I,J)
219 CONTINUE
GO TO 211
220 DO 222 I=1,NL
XSL(I,J)=XSU(I,J)
ZSL(I,J)=-ZSU(I,J)
222 CONTINUE
GO TO 211
230 NSU(J)=NU
DO 235 I=1,NU
XSU(I,J)=XSU(I,J-1)
ZSU(I,J)=ZSU(I,J-1)
235 CONTINUE
NSL(J)=NL
DO 236 I=1,NL
XSL(I,J)=XSL(I,J-1)
ZSL(I,J)=ZSL(I,J-1)
236 CONTINUE
GO TO 211
237 CONTINUE
WRITE (6,921) JSECT,JBL,JBT
DO 380 J=1,JSECT
WRITE (6,922) J,YS(J),XLES(J),ZLES(J),CS(J),ATS(J),TS(J),FS(J)
IF (FS(J).NE.0.) GO TO 370
WRITE (6,923)
GO TO 380
370 NU=NSU(J)
WRITE (6,924) NU
WRITE (6,926) (XSU(I,J),ZSU(I,J),I=1,NU)
NL=NSL(J)
WRITE (6,925) NL
WRITE (6,926) (XSL(I,J),ZSL(I,J),I=1,NL)
380 CONTINUE
C
DO 438 J=1,JSECT
438 XTES(J)=XLES(J)+CS(J)
IF (JBL.NE.0) GO TO 440
JSECTX=JSECT-1
XSLOP=(XLES(JSECT)-XLES(1))/(YS(JSECT)-YS(1))
ZSLOP=(ZLES(JSECT)-ZLES(1))/(YS(JSECT)-YS(1))
DO 439 J=2,JSECTX
XLES(J)=XLES(1)+(YS(J)-YS(1))*XSLOP
439 ZLES(J)=ZLES(1)+(YS(J)-YS(1))*ZSLOP
GO TO 445
440 JBLX=JBL-1
XSLOP=(XLES(JBL)-XLES(1))/(YS(JBL)-YS(1))
ZSLOP=(ZLES(JBL)-ZLES(1))/(YS(JBL)-YS(1))
DO 441 J=2,JBLX
XLES(J)=XLES(1)+(YS(J)-YS(1))*XSLOP
441 ZLES(J)=ZLES(1)+(YS(J)-YS(1))*ZSLOP
JBLP=JBL+1
JSECTX=JSECT-1
XSLOP=(XLES(JSECT)-XLES(JBL))/(YS(JSECT)-YS(JBL))
ZSLOP=(ZLES(JSECT)-ZLES(JBL))/(YS(JSECT)-YS(JBL))
DO 442 J=JBLP,JSECTX
XLES(J)=XLES(JBL)+(YS(J)-YS(JBL))*XSLOP
442 ZLES(J)=ZLES(JBL)+(YS(J)-YS(JBL))*ZSLOP

```

```

445 IF (JRT.NE.0) GO TO 447
    XSLOP=(XTES(JSECT)-XTES(1))/(YS(JSECT)-YS(1))
    DO 446 J=2,JSECTX
446 XTES(J)=XTES(1)+(YS(J)-YS(1))*XSLOP
    GO TO 450
447 JBTX=JBT-1
    XSLOP=(XTES(JBT)-XTES(1))/(YS(JBT)-YS(1))
    DO 448 J=2,JBTX
448 XTES(J)=XTES(1)+(YS(J)-YS(1))*XSLOP
    JBTP=JBT+1
    XSLOP=(XTES(JSECT)-XTES(JBT))/(YS(JSECT)-YS(JBT))
    DO 449 J=JBTP,JSECTX
449 XTES(J)=XTES(JBT)+(YS(J)-YS(JBT))*XSLOP
450 CONTINUE
    DO 451 J=1,JSECT
451 CS(J)=XTES(J)-XLES(J)
    BSPAN=2*YS(JSECT)

```

C

```

    WRITE (6,931)
    ITER1=1
    GO TO 520
510 WRITE (6,932)
    ITER1=ITERM
    CALL HALFS(ITERM,NPRINT)
520 ITER2=ITER1+ITERM-1
    CALL GRID
    CALL GEOM
    CALL PROFL
    CALL COEFF
    CALL INIT(MSTART)
    WRITE (6,941)
    DO 590 ITER=ITER1,ITER2
    CALL SOLVE(ITER)
    IF (NPRINT.EQ.0) GO TO 550
    IF (ITER/NPRINT*ITERM.NE.ITER) GO TO 550
    CALL ARRAYS(NPRINT)
    CALL CPCOMP
550 IF (OPMAX.LE.OPLIM) GO TO 600
590 CONTINUE
600 CALL ARRAYS(NPRINT)
    CALL CPCOMP
    IF (NCYCLE.EQ.NHALF) STOP
    MSTART=1
    GO TO 510

```

C

```

901 FORMAT (20A4)
902 FORMAT (8F10.5)
903 FORMAT (33H1TRANSONIC WING/BODY PROGRAM TWB3
*      52X43H CODED BY: G. E. CHMIELEWSKI
*      85X43H MCDONNELL DOUGLAS RESEARCH LABS./
*      85X43H ST. LOUIS, MISSOURI 63166
*      85X43H JUNE 1980
*      /1X20A4///
*      5X40HMACH NUMBER ZMACH = F8.3/
*      5X40HANGLE OF ATTACK ALPHA = F8.3/
*      3X7HDEGREES////)
904 FORMAT (21H-EXECUTION PARAMETERS//
*      5X40HMAXIMUM ITERATIONS ITERM = I8/
*      5X40HGRID HALVING CYCLES NHALF = I8/
*      5X40HSOLUTION ARRAY PRINT CYCLE NPRINT = I8////
*      1X21HRELAXATION PARAMETERS//
*      5X40HELLIPTIC POINT WE = F8.3/
*      5X40HCIRCULATION WG = F8.3//
*      5X40HCONVERGENCE LIMIT OPLIM = E12.3/
*      5X40HUAMPING FACTOR EPSI = F8.3//)
905 FORMAT (22H-CUORDINATE PARAMETERS//
*      5X40HSTREAMWISE DIRECTION NXI = I8/
*      5X40H A1 = F8.3/
*      5X40H A2 = F8.3/
*      5X40H X10 = F8.3/
*      5X40H XCAPO = F8.3//)

```



```

*      5X40H SPANWISE DIRECTION          NETA   = 18/
*      5X40H                              A3     = F8.3/
*      5X40H                              A4     = F8.3/
*      5X40H                              ETAO  = F8.3/
*      5X40H                              YCAPO  = F8.3//
*      5X40H VERTICAL DIRECTION          NZETA  = 18/
*      5X40H                              A5     = F8.3)
911 FORMAT (1H-///1X13HFUSELAGE DATA///9X1HN,8X2HXF,8X2HZF,8X2HRF/)
912 FORMAT (I10,8F10.4)
921 FORMAT (1H-///10H WING DATA///20H NUMBER OF SECTIONS ,I10/
*      22H SECTION AT L.E. BREAK,I8/
*      22H SECTION AT T.E. HREAK,I8)
922 FORMAT (1H-///13H WING SECTION,I10//
*      1X5HYS = F9.6,5X5XLES= F9.6,5X5HZLES= F9.6,
*      5X5HCS = F9.6,5X5HATS = F9.6,5X5HTS = F9.6,
*      5X5HFS = F9.6)
923 FORMAT (43H-SURFACE DATA IDENTICAL TO PREVIOUS SECTION)
924 FORMAT (1H-/26H UPPER SURFACE (X-Z PAIRS)//1X5HNSU = I8/)
925 FORMAT (1H-/20H LOWER SURFACE (X-Z PAIRS)//1X5HNSL = I8/)
926 FORMAT (I2F11.6)
931 FORMAT (1H-///45H INITIAL START *****
*      51H*****
*      40H*****
932 FORMAT (1H-///45H CONTINUATION ON HALF-SPACED GRID *****
*      51H*****
*      40H*****
94) FORMAT (1H-///1X17H ITERATION SUMMARY//
*      6X4HITER,10X5HDPMAX,4X1MI,4X1HJ,4X1HK,6X4HNSUP,
*      14X6HGAMMAR,5X5HITERT,7X6HDPMAXT,3X2HJT,3X2HKT/)
C
END

```

SUBROUTINE HALFS(ITERM,NPRINT)

INTERPOLATES POTENTIAL AND CIRCULATION ARRAYS ONTO A HALF-SPACED COMPUTATIONAL GRID.

```

C
C
C
COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPST,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
*      SNA,TDETA,TDXI,TUZETA,OTETA,OXI,DZETA,ILE,IM,INOSE,
*      ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
*      PLCB2,PLCB3
COMMON/DATAX/  AI,A2,A3,A4,A5,ETAO,NETA,NXI,NZETA,XCAPO,XIO,YCAPO,
*      NF,XF(150),ZF(150),RF(150),JSECT,JBL,JBT,NSU(5),
*      NSL(5),YS(5),XLES(5),ZLES(5),CS(5),ATS(5),TS(5),
*      FS(5),XSU(150,5),XSL(150,5),ZSU(150,5),ZSL(150,5)
COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
*      DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERT,
*      JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
*      PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
*      PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)
COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZDXL(49,19),
*      DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
*      HWBU(49,19),IZL(19),IZU(19),JBOD(49),KBODL(49,19),
*      KBODU(49,19),ZL(49,19),ZU(49,19)
C
C
C

```

```

NXI=2,*NXI
NETA=2,*NETA
NZETA=2,*NZETA
ITERM=ITERM/2
NPRINT=2*NPRINT

```

```

IMN=2*IM-1
JMN=2*JM-1
KMN=2*KM-1

```

```

      IMNX=IMN-1
      JMNX=JMN-1
      KMNX=KMN-1
      DO 200 I=1,IM
      JE=JROD(I)
      JEX=JE-1
      DO 130 J=JE,JM
      DO 110 K=1,KM
      KN=KMN-(2*K-1)+1
110  P(I,J,KN)=P(I,J,K)
      DO 120 K=2,KMNX,2
120  P(I,J,K)=0.5*(P(I,J,K-1)+P(I,J,K+1))
130  CONTINUE
      IF (JE.EQ.1) GO TO 200
      DO 190 J=1,JEX
      KE=KROUJ(1,J)-1
      P(I,J,KE)=DU1(I,J)
      DO 140 K=Kw,KM
      KN=KMN-(2*K-1)+1
140  P(I,J,KN)=P(I,J,K)
      KWN1=2*Kw
      DO 150 K=KWN1,KMNX,2
150  P(I,J,K)=0.5*(P(I,J,K-1)+P(I,J,K+1))
      KE=KRODL(1,J)+1
      P(I,J,KE)=DL1(I,J)
      DO 160 K=1,Kw
      KN=KMN-(2*K-1)+1
160  P(I,J,KN)=P(I,J,K)
      KWN1=2*Kw-2
      DO 170 K=2,KWN1,2
170  P(I,J,K)=0.5*(P(I,J,K-1)+P(I,J,K+1))
190  CONTINUE
200  CONTINUE
C
      KWN=2*Kw-1
      DO 210 I=INOSE,IM
      JEX=JROD(I)-1
210  P(I,JEX,KWN)=DS1(I)
      P(INOSE,1,KWN)=PNOSE
      DO 220 J=1,JTIP
      P(ILE,J,KWN)=PLE1(J)
220  P(ITE,J,KWN)=PTE1(J)
      DO 260 K=1,KMN
      DO 250 J=1,JM
      DO 230 I=1,IM
      IN=IMN-(2*I-1)+1
230  P(IN,J,K)=P(I,J,K)
      DO 240 I=2,IMNX,2
240  P(I,J,K)=0.5*(P(I-1,J,K)+P(I+1,J,K))
250  CONTINUE
260  CONTINUE
C
      P(2*INOSE-1,1,KWN)=DS1(INOSE)
      P(2*ILE,JTIP,KWN)=DS1(ILE)
      P(2*ITE,JTIP,KWN)=DS1(ITE)
      DO 340 K=1,KMN
      DO 330 I=1,IMN
      DO 310 J=1,JM
      JN=JMN-(2*JM-1)+1
310  P(I,JN,K)=P(I,J,K)
      DO 320 J=2,JMNX,2
320  P(I,J,K)=0.5*(P(I,J-1,K)+P(I,J+1,K))
330  CONTINUE
340  CONTINUE
C
      DO 410 J=1,JM
      JN=JMN-(2*J-1)+1
410  GAMMA(JN)=GAMMA(J)
      DO 420 J=2,JMNX,2
420  GAMMA(J)=0.5*(GAMMA(J-1)+GAMMA(J+1))
      JTIPN=2*JTIP-1
      GAMMA(JTIPN+1)=0.

```

```

JROOTN=2*JROOT-1
GAMMA(JROOTN-2)=0.
RETURN
END

```

SUBROUTINE GRID

CALCULATES GRID-POINT COORDINATES, STRETCHING DERIVATIVES, AND WING CONFIGURATION DATA.

DIMENSION XI(49),YCAP(19)

```

COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,A12,CSA,RX1,RX2,
SNA,TDETA,TDXI,TUZETA,DETA,DXI,OZETA,ILE,IM,INOSE,
ITAIL,ITE,JM,JROOT,OTIP,KM,KW,JB1,JB2,JB3,PLCB1,
PLCB2,PLCB3
COMMON/COORD/ CHORD(19),DCDY(19),UTDY(19),DXLEDY(19),ETA(19),F(49),
FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
ZCAP(25),ZLE(19),OZLEDY(19),ALPHAT(19),DADY(19),GWT
COMMON/DATAX/ A1,A2,A3,A4,A5,ETAO,NETA,NXI,NZETA,XCAPO,XIO,YCAPO,
NF,XF(150),ZF(150),RF(150),JSECT,JRL,JBT,NSU(5),
NSL(5),YS(5),XLES(5),ZLES(5),CS(5),ATS(5),TS(5),
FS(5),XSU(150,5),XSL(150,5),ZSU(150,5),ZSL(150,5)

```

```

102 FORMAT (1H-///1X27HSTREAMWISE GRID COORDINATES///
1 1X8HDXI = F14.6//
2 9X1HI,11X4HXCAP,13X2HXI,14X1HF/)
108 FORMAT (I10,3(1X,F14.6))
294 FORMAT (1H0/1X8HFLE = F14.6/1X8HGAMLE = F14.6/
1 1X8HXCAPLE = F14.6/1X8HXILE = F14.6//1X8HFTE = F14.6/
2 1X8HGAMTE = F14.6/1X8HXCAPTE = F14.6/1X8HXITE = F14.6)
355 FORMAT (//1X8HFN = F14.6/1X8HGAMN = F14.6/1X8HXCAPN = F14.6
1 //1X8HXIN = F14.6)
357 FORMAT (//1X8HINOSE = I14/1X8HILE = I14/1X8HITE = I14/
1 1X8HITAIL = I14/1X8HIM = I14)
411 FORMAT (1H-///1X25HSPANWISE GRID COORDINATES///
1 1X8HDETA = F14.6//
2 9X1HJ,11X4HYCAP,12X3HETA,14X1HG/)
595 FORMAT (1H0/1X8HGWT = F14.6/1X8HGAMWT = F14.6/
1 1X8HYCAPWT = F14.6/1X8HETAWT = F14.6)
655 FORMAT (//1X8HGF = F14.6/1X8HGAMF = F14.6/1X8HYCAPF = F14.6
1 //1X8HETAF = F14.6)
657 FORMAT (//1X8HJROOT = I14/1X8HJTIP = I14/1X8HJM = I14)
734 FORMAT (1H-///1X25HVERTICAL GRID COORDINATES///
1 1X8HDZETA = F14.6//
2 9X1HK,11X4HZCAP,11X4HZETA,14X1HH/)
7749 FORMAT (//1X8HKW = I14/1X8HKM = I14)
801 FORMAT (I8,10(1X,F10.6))
836 FORMAT (1H-///1X35HWING PLANFORM AREA S = F8.4/
1 1X35HWING ASPECT RATIO AR = F8.4///
2 1X35HSTATION AT L.E. BREAK YB1 = F8.4,
3 10X5HJB1 = I4,10X7HETAB1 = F8.4,10X7HPLCB1 = F8.4/
4 1X35HSTATION AT T.E. BREAK YB2 = F8.4,
5 10X5HJB2 = I4,10X7HETAB2 = F8.4,10X7HPLCB2 = F8.4/
6 1X35HCONSTANT-CHORD STATION YB3 = F8.4,
7 10X5HJB3 = I4,10X7HETAB3 = F8.4,10X7HPLCB3 = F8.4)
871 FORMAT (1H-///1X41HWING CONFIGURATION DATA (PHYSICAL DOMAIN)//
1 7X1HJ,8X3HXLE,4X7HDXLE/DY,6X5HCHORD,6X5HDC/DY,8X3HTAU,
2 6X5HDT/DY,8X3HZLE,4X7HDZLE/DY,5X6HALPHAT,6X5HDA/DY/)

```

C
C

XCAP-STRETCHING

```

HPI=2.*ATAN(1.)
XIM=1.+XIO
DXI=2.*XIM/NXI
IM=NXI+1
IO=NXI/2+1
ID=XIO/DXI
ILE=IO-ID
ITE=IO+ID
XI(IO)=0.
II=IO+1
DO 50 I=II,IM
50 XI(I)=XI(I-1)+DXI
DO 60 I=2,IO
IX=IO-I+1
60 XI(IX)=XI(IX+1)-DXI
ALC1=0.5*(3.*XCAPO/XIO-HPI*A1)
ALC2=0.5*(HPI*A1-XCAPO/XIO)/(XIO*XIO)
WRITE (6,102) DXI
XCAP(I)=-1.E+99
F(I)=0.
ILX=ILE-1
DO 110 I=1,ILX
IF (I.EQ.1) GO TO 109
PP=XI(I)+XIO
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
XCAP(I)=-XCAPO+A1*TN2+A2*TN3
F(I)=1./(HPI*(A1*(1.+TN2*TN2)+3.*A2*PP*PP*(1.+TN3*TN3)))
109 WRITE (6,108) I,XCAP(I),XI(I),F(I)
110 CONTINUE
DO 120 I=ILE,ITE
PP2=XI(I)**2
XCAP(I)=XI(I)*(ALC1+ALC2*PP2)
F(I)=1./(ALC1+3.*ALC2*PP2)
WRITE (6,108) I,XCAP(I),XI(I),F(I)
120 CONTINUE
XCAP(IM)=1.E+99
F(IM)=0.
ITEP=ITE+1
DO 130 I=ITEP,IM
IF (I.EQ.IM) GO TO 129
PP=XI(I)-XIO
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
XCAP(I)=XCAPO+A1*TN2+A2*TN3
F(I)=1./(HPI*(A1*(1.+TN2*TN2)+3.*A2*PP*PP*(1.+TN3*TN3)))
129 WRITE (6,108) I,XCAP(I),XI(I),F(I)
130 CONTINUE
C
IF (XCAPO.NE.0.5) GO TO 200
XILE=-XIO
XITE=XIO
GO TO 290
200 XI1=XIO+DXI
210 PP=XI1-XIO
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
FUN=A1*TN2+A2*TN3-0.5*XCAPO
FUNPR=HPI*(A1*(1.+TN2*TN2)+3.*A2*PP*PP*(1.+TN3*TN3))
DEL=FUN/FUNPR
XI2=XI1-DEL
IF (DEL.LT.1.E-10) GO TO 250
XI1=XI2
GO TO 210
250 XITE=XI2
XILE=-XI2

```

```

290 GAMLE=ABS(XILE-XI(ILF-1))
   GAMTE=ABS(XI(ITE+1)-XITE)
   PP=XITE-XIO
   TN2=TAN(HPI*PP)
   PP3=PP**3
   TN3=TAN(HPI*PP3)
   XCAPTE=XCAPO+A1*TN2+A2*TN3
   XCAPLE=-XCAPTE
   FTE=1./(HPI*(A1*(1.+TN2*TN2)+3.*A2*PP*PP*(1.+TN3*TN3)))
   FLE=FTE
   WRITE (6,294) FLE,GAMLE,XCAPLE,XILE,FTE,GAMTE,XCAPTE,XITE

```

C

```

   DCS=CS(2)-CS(1)
   DXS=XLES(2)-XLES(1)
   DYS=YS(2)-YS(1)
   IF(DYS.EQ.0.) STOP 'ERROR YS(1)=YS(2)'
   XREF=XLES(1)-YS(1)*DXS/DYS
   DXS=DXS+DCS
   CREF=XLES(1)+CS(1)-YS(1)*DXS/DYS-XREF
   XCAPN=-XREF/CREF-0.5
   DO 300 I=1,ILE
   DIFF=XCAP(I)-XCAPN
   IF (DIFF.GE.0.) GO TO 305
300 CONTINUE
305 INOSE=I
   XI1=XI(INOSE-1)
310 PP=XI1*XIO
   TN2=TAN(HPI*PP)
   PP3=PP**3
   TN3=TAN(HPI*PP3)
   FUN=A1*TN2+A2*TN3-XCAPN-XCAPO
   FUNPR=HPI*(A1*(1.+TN2*TN2)+3.*A2*PP*PP*(1.+TN3*TN3))
   DEL=FUN/FUNPR
   XI2=XI1-DEL
   DELT=ABS(DEL)
   IF (DELT.LT.1.E-10) GO TO 350
   XI1=XI2
   GO TO 310
350 XIN=XI2
   GAMN=ABS(XI(INOSE-1)-XIN)
   PP=XIN*XIO
   TN2=TAN(HPI*PP)
   PP3=PP**3
   TN3=TAN(HPI*PP3)
   FN=1./(HPI*(A1*(1.+TN2*TN2)+3.*A2*PP*PP*(1.+TN3*TN3)))
   WRITE (6,355) FN,GAMN,XCAPN,XIN
   FL=XF(NF)-XF(1)
   XCAPT=(FL-XREF)/CREF-.5
   DO 360 I=1,IM
   IF(XCAP(I).GT.XCAPT) GO TO 370
360 CONTINUE
370 ITAIL=I-1
   IF(XF(NF).GT.25.) ITAIL=IM
   WRITE (6,357) INOSE,ILE,ITE,ITAIL,IM

```

C
C
C

```

YCAP=STRETCHING

   ETAM=1.+ETA0
   UETA=ETAM/NETA
   JM=NETA+1
   JTIP=1.+ETA0/UETA
   ETA(1)=0.
   DO 400 J=2,JM
400 ETA(J)=ETA(J-1)+DETA
   ALC3=0.5*(3.*YCAPO/ETA0-HPI*A3)
   ALC4=0.5*(HPI*A3-YCAPO/ETA0)/(ETA0*ETA0)
   WRITE (6,411) DETA
   DO 420 J=1,JTIP
   PP2=ETA(J)**2
   YCAP(J)=ETA(J)*(ALC3+ALC4*PP2)
   G(J)=1./(ALC3+3.*ALC4*PP2)
   WRITE (6,108) J,YCAP(J),ETA(J),G(J)

```

```

420 CONTINUE
YCAP(JM)=1.E+99
G(JM)=0.
JTIPP=JTIP+1
DO 450 J=JTIPP,JM
IF (J.EQ.JM) GO TO 445
PP=ETA(J)-ETAU
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
YCAP(J)=YCAPO+A3*TN2+A4*TN3
G(J)=1./(HPI*(A3*(1.+TN2*TN2)+3.*A4*PP*PP*(1.+TN3*TN3)))
445 WRITE (6,108) J,YCAP(J),ETA(J),G(J)
450 CONTINUE
C
IF (YCAPO.NE.0.5) GO TO 500
ETAWT=ETAU
GO TO 590
500 ETA1=ETA0+DETA
510 PP=ETA1-ETA0
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
FUN=A3*TN2+A4*TN3-0.5*YCAPO
FUNPR=HPI*(A3*(1.+TN2*TN2)+3.*A4*PP*PP*(1.+TN3*TN3))
DEL=FUN/FUNPR
ETA2=ETA1-DEL
IF (DEL.LT.1.E-10) GO TO 550
ETA1=ETA2
GO TO 510
550 ETAWT=ETA2
590 GAMWT=ABS(ETA(JTIP+1)-ETAWT)
PP=ETAWT-ETA0
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
YCAPWT=YCAPO+A3*TN2+A4*TN3
GWT=1./(HPI*(A3*(1.+TN2*TN2)+3.*A4*PP*PP*(1.+TN3*TN3)))
WRITE (6,595) GWT,GAMWT,YCAPWT,ETAWT
C
YFS=RF(1)
DO 460 N=2,NF
IF (YFS.LT.RF(N)) YFS=RF(N)
460 CONTINUE
YCAPF=YFS/HSPAN
DO 600 J=1,JTIP
DIFF=YCAP(J)-YCAPF
IF (DIFF.GE.0.) GO TO 605
600 CONTINUE
605 JROOT=J
ETA1=ETA(JROOT-1)
ETA1SQ=ETA1*ETA1
610 FUN=ETA1*(ALC3+ALC4*ETA1SQ)-YCAPF
FUNPR=ALC3+3.*ALC4*ETA1SQ
DEL=FUN/FUNPR
ETA2=ETA1-DEL
IF (DEL.LT.1.E-10) GO TO 650
ETA1=ETA2
GO TO 610
650 ETAF=ETA2
GAMF=ABS(ETA(JROOT)-ETAF)
GF=1./(ALC3+3.*ALC4*ETAF*ETAF)
WRITE (6,655) GF,GAMF,YCAPF,ETAF
WRITE (6,657) JROOT,JTIP,JM
C
C
C
ZCAP-STRETCHING
ZETAM=1.0
DZETA=2.*ZETAM/NZETA
KM=NZETA+1
KW=NZETA/2+1

```

```

ZETA(KW)=0.
KK=KW+1
DO 710 K=KK,KM
710 ZETA(K)=ZETA(K-1)+DZETA
DO 720 K=2,KW
KX=KW-K+1
720 ZETA(KX)=ZETA(KX+1)-DZETA
WRITE (6,734) DZETA
ZCAP(1)=-1.E+99
H(1)=0.
ZCAP(KM)=-ZCAP(1)
H(KM)=0.
DO 750 K=1,KM
ZET=ZETA(K)
IF (K.EQ.1.OR.K.EQ.KM) GO TO 749
TN1=TAN(HPI*ZET)
ZCAP(K)=A5*TN1
H(K)=1./(A5*HPI*(1.+TN1*TN1))
749 WRITE (6,108) K,ZCAP(K),ZET,H(K)
750 CONTINUE
WRITE (6,1749) KW,KM

```

C

WING CONFIGURATION DATA

C

```

JSL=JBL
JST=JBT
IF (JSL.EQ.0) JSL=JSECT
IF (JST.EQ.0) JST=JSECT
XTIP=XLES(JSECT)
CTIP=CS(JSECT)
SWING=BSPAN*(XTIP+CTIP-XREF)
SWING=SWING-(XLES(JSL)-XREF)*YS(JSL)
SWING=SWING-(XLES(JSL)-XREF+XTIP-XREF)*(YS(JSECT)-YS(JSL))
SWING=SWING-(XLES(JST)+CS(JST)-XREF-CREF)*YS(JST)
SWING=SWING-(XTIP+CTIP-XLES(JST)-CS(JST))*(YS(JSECT)+YS(JST))
AR=BSPAN*BSPAN/SWING

```

C

```

DO 800 J=1,JM
Y(J)=BSPAN*YCAP(J)
800 CONTINUE
CALC1=YS(JSECT)-YS(JSECT-1)
CALC2=XLES(JSECT)-XLES(JSECT-1)
CALC3=CS(JSECT)-CS(JSECT-1)
IF (CALC3.EQ.0.) GO TO 803
YSET=YS(JSECT)-CS(JSECT)*CALC1/CALC3
YB3=(YSET+YS(JSECT))/2
XLEB3=XLES(JSECT)+(YB3-YS(JSECT))*CALC2/CALC1
XTEB3=XLES(JSECT)+CS(JSECT)+(CALC3+CALC2)*(YB3-YS(JSECT))/CALC1
DO 802 J=1,JM
CALC=YB3-Y(J)
IF (CALC.LT.0.) GO TO 805
802 CONTINUE
803 JB3=JM
YB3=1.E+99
PLCB3=0.0
ETAB3=ETA(JM)
GO TO 827
805 JB3=J-1
YCAPB3=YB3/BSPAN
ETA1=ETA(JB3)+DETA
8825 PP=ETA1-ETA0
TN2=TAN(HPI*PP)
PP3=PP**3
TN3=TAN(HPI*PP3)
FUN=A3*TN2+A4*TN3-YCAPB3+YCAP0
FUNPR=HPI*(A3*(1.+TN2*TN2)+3.*A4*PP*PP*(1.+TN3*TN3))
DEL=FUN/FUNPR
ETA2=ETA1-DEL
IF (DEL.LT.1.E-10) GO TO 8826
ETA1=ETA2
GO TO 8825

```

```

8826 ETAB3=ETA2
    PLCH3=(ETAB3-ETA(JR3))/DETA
827 CONTINUE
C
    DO 828 J=1,JM
    IF (JRL.EQ.0) GO TO 830
    CALC=YS(JBL)-Y(J)
    IF (CALC.LT.0.) GO TO 831
828 CONTINUE
830 JB1=0
    YB1=YS(1)
    GO TO 832
831 JB1=J-1
    YB1=YS(JBL)
832 CONTINUE
    YCAPB1=YB1/BSPAN
    IF (JB1.EQ.0) JSTART=JROOT-1
    IF (JB1.NE.0) JSTART=JB1
    ETA1=ETA(JSTART)
8833 ETA1SQ=ETA1*ETA1
    FUN=ETA1*(ALC3+ALC4*ETA1SQ)-YCAPB1
    FUNPR=ALC3+3.*ALC4*ETA1SQ
    DEL=FUN/FUNPR
    ETA2=ETA1-DEL
    IF (DEL.LT.1.E-10) GO TO 8834
    ETA1=ETA2
    GO TO 8833
8834 ETAB1=ETA2
    PLCB1=(ETAB1-ETA(JSTART))/DETA
C
    DO 835 J=1,JM
    IF (JRT.EQ.0) GO TO 8835
    CALC=YS(JBT)-Y(J)
    IF (CALC.LT.0.) GO TO 8836
835 CONTINUE
8835 JB2=0
    YB2=YS(1)
    GO TO 837
8836 JB2=J-1
    YB2=YS(JBT)
837 CONTINUE
    YCAPB2=YB2/BSPAN
    IF (JB2.EQ.0) JSTART=JROOT-1
    IF (JB2.NE.0) JSTART=JB2
    ETA1=ETA(JSTART)
8838 ETA1SQ=ETA1*ETA1
    FUN=ETA1*(ALC3+ALC4*ETA1SQ)-YCAPB2
    FUNPR=ALC3+3.*ALC4*ETA1SQ
    DEL=FUN/FUNPR
    ETA2=ETA1-DEL
    IF (DEL.LT.1.E-10) GO TO 8839
    ETA1=ETA2
    GO TO 8838
8839 ETAB2=ETA2
    PLCB2=(ETAB2-ETA(JSTART))/DETA
    WRITE(6,836) SWING,AR,YB1,JB1,ETAB1,PLCB1,
1 YB2,JB2,ETAB2,PLCB2,YB3,JB3,ETAB3,PLCB3
C
    SLOPE=(YS(2)-YS(1))/(XLES(2)-XLES(1))
    DO 845 J=1,JB1
    XLE(J)=XLES(1)+(Y(J)-YS(1))/SLOPE
    DXLEDY(J)=1/SLOPE
845 CONTINUE
    CALC1=YS(JSECT)-YS(JSECT-1)
    CALC2=XLES(JSECT)-XLES(JSECT-1)
    SLOPE1=CALC1/CALC2
    JB1P=JB1+1
    DO 847 J=JB1P,JM
    XLE(J)=XLES(JSECT)+(Y(J)-YS(JSECT))/SLOPE1
    DXLEDY(J)=1/SLOPE1
847 CONTINUE
C

```



```

      CALC=(ZLES(2)-ZLES(1))/(YS(2)-YS(1))
      DO 858 J=1,JM
      IF(YS(1).LE.Y(J)) GO TO 852
      ZLE(J)=ZLES(1)+(Y(J)-YS(1))*CALC
      DZLEDY(J)=CALC
      GO TO 858
852 DO 853 JS=2,JSECT
      IF(YS(JS).GE.Y(J)) GO TO 857
853 CONTINUE
      DYS=YS(JSECT)-YS(JSECT-1)
      OZLES=ZLES(JSECT)-ZLES(JSECT-1)
      CALC=OZLES/OYS
      ZLEB3=ZLES(JSECT)+(YB3-YS(JSECT))*CALC
      IF(Y(J).GT.YB3) GO TO 854
      ZLE(J)=ZLEB3+(Y(J)-YB3)*CALC
      DZLEDY(J)=CALC
      GO TO 858
854 ZLE(J)=ZLER3
      DZLEDY(J)=0
      GO TO 858
857 CALC=(ZLES(JS)-ZLES(JS-1))/(YS(JS)-YS(JS-1))
      ZLE(J)=ZLES(JS)+(Y(J)-YS(JS))*CALC
      DZLEDY(J)=CALC
858 CONTINUE
C
      CALC=(CS(2)-CS(1))/(YS(2)-YS(1))
      DO 868 J=1,JM
      IF(YS(1).LE.Y(J)) GO TO 862
      CHORD(J)=CS(1)+(Y(J)-YS(1))*CALC
      DCDY(J)=CALC
      GO TO 868
862 DO 863 JS=2,JSECT
      IF(YS(JS).GE.Y(J)) GO TO 867
863 CONTINUE
      DYS=YS(JSECT)-YS(JSECT-1)
      DCS=CS(JSECT)-CS(JSECT-1)
      CALC=DCS/OYS
      CHORDB3=XLER3-XLER3
      IF(Y(J).GT.YB3) GO TO 864
      CHORD(J)=CHORDB3+(Y(J)-YB3)*CALC
      DCDY(J)=CALC
      GO TO 868
864 CHORD(J)=CHORUR3
      DCDY(J)=0
      GO TO 868
867 CALC=(CS(JS)-CS(JS-1))/(YS(JS)-YS(JS-1))
      CHORD(J)=CS(JS)+(Y(J)-YS(JS))*CALC
      DCDY(J)=CALC
868 CONTINUE
C
      TROOT=TS(1)
      DO 870 J=1,JSECT
870 TS(J)=TS(J)/TROOT
      CALC=(TS(2)-TS(1))/(YS(2)-YS(1))
      DO 878 J=1,JM
      IF(YS(1).LE.Y(J)) GO TO 872
      TAU(J)=TS(1)+(Y(J)-YS(1))*CALC
      DTDY(J)=CALC
      GO TO 878
872 DO 873 JS=2,JSECT
      IF(YS(JS).GE.Y(J)) GO TO 877
873 CONTINUE
      DYS=YS(JSECT)-YS(JSECT-1)
      DTS=TS(JSECT)-TS(JSECT-1)
      CALC=DTS/OYS
      TAUB3=TS(JSECT)+(YB3-YS(JSECT))*CALC
      IF(Y(J).GT.YB3) GO TO 874
      TAU(J)=TAUB3+(Y(J)-YB3)*CALC
      DTDY(J)=CALC
      GO TO 878

```

```

874 TAU(J)=TAUH3
   DTDY(J)=0
   GO TO 878
877 CALC=(TS(JS)-TS(JS-1))/(YS(JS)-YS(JS-1))
   TAU(J)=TS(JS)+(Y(J)-YS(JS))*CALC
   DTDY(J)=CALC
878 CONTINUE
C
   CALC=(ATS(2)-ATS(1))/(YS(2)-YS(1))
   DO 888 J=1,JM
   IF(YS(1).LE.Y(J)) GO TO 882
   ALPHAT(J)=ATS(1)+(Y(J)-YS(1))*CALC
   DADY(J)=CALC
   GO TO 888
882 DO 883 JS=2,JSECT
   IF(YS(JS).GE.Y(J)) GO TO 887
883 CONTINUE
   DYS=YS(JSECT)-YS(JSECT-1)
   DATS=ATS(JSECT)-ATS(JSECT-1)
   CALC=DATS/DYS
   ALPHATB3=ATS(JSECT)+(YB3-YS(JSECT))*CALC
   IF(Y(J).GT.YB3) GO TO 884
   ALPHAT(J)=ALPHATB3+(Y(J)-YB3)*CALC
   DADY(J)=CALC
   GO TO 888
884 ALPHAT(J)=ALPHATB3
   DADY(J)=0
   GO TO 888
887 CALC=(ATS(JS)-ATS(JS-1))/(YS(JS)-YS(JS-1))
   ALPHAT(J)=ATS(JS)+(Y(J)-YS(JS))*CALC
   DADY(J)=CALC
888 CONTINUE
C
   WRITE (6,871)
   DO 895 J=1,JM
895 WRITE (6,801) J,XLE(J),DXLEDY(J),CHORD(J),DCDY(J),TAU(J),DTDY(J)
   1,ZLE(J),DZLEDY(J),ALPHAT(J),DADY(J)
   DO 900 I=1,IM
   DO 900 J=1,JM
   X(I,J)=(XCAP(I)+0.5)*CHORD(J)+XLE(J)
900 CONTINUE
   RETURN
   END

```

SUBROUTINE GEUM

COMPUTES WING/BODY COORDINATES Z(X,Y) AND SURFACE SLOPES DZ/DX AND DZ/DY AT GRID-POINT STATIONS IN THE PHYSICAL DOMAIN.

DIMENSION SF(150),XSET(150),SO(150),RFS(150),ZFS(150),XFSP(150),
RFSP(150),ZFSP(150),XFSP(150),RFSP(150),ZFSP(150)

```

COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WF,WG,BSPAN,AI2,CSA,RX1,RX2,
* SNA,TDETA,TDXI,TUZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
COMMON/COORD/ CHORD(19),DCDY(19),DTDY(19),DXLEDY(19),ETA(19),F(49),
* FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
* TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
* ZCAP(25),ZLE(19),DZLEDY(19),ALPHAT(19),DADY(19),GWT
COMMON/DATAX/ A1,A2,A3,A4,A5,ETAO,NETA,NXI,NZETA,XCAPO,XIO,YCAPO,
* NF,XF(150),ZF(150),RF(150),JSECT,JBL,JBT,NSU(5),
* NSL(5),YS(5),XLES(5),ZLES(5),CS(5),ATS(5),TS(5),
* FS(5),XSU(150,5),XSL(150,5),ZSU(150,5),ZSL(150,5)
COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZOXL(49,19),
* DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
* HWBU(49,19),IZL(19),IZU(19),JBOD(49),KBODL(49,19),
* KRODU(49,19),ZL(49,19),ZU(49,19)

```

```

C
C
249 FORMAT (1H-///1X29HPLANFORM-ADJACENT GRID POINTS
1 //3X1H1,8X7HJBOD(I)/)
251 FORMAT (I4,I15)

C
C
DO 50 I=1,IM
JBOD(I)=1
DO 50 J=1,JM
ZU(I,J)=0.
ZL(I,J)=0.
DZDXU(I,J)=0.
DZDXL(I,J)=0.
DZDYU(I,J)=0.
50 DZDYL(I,J)=0.

C
C
FUSELAGE CO-ORDINATES

DO 56 I=INOSE,ITAIL
K=I-INOSE+1
DO 54 J=1,JROUT
DO 52 I1=INOSE,ITAIL
K1=I1-INOSE+1
XSET(K1)=X(I1,J)
52 CONTINUE
NO=ITAIL-INOSE+1
IF(XSET(NO).GT.XF(NF)) NO=NO-1
CALL SSPLINE(NF,XF,ZF,NO,XSET,ZFS,ZFSP,ZFSP,4)
CALL ASPLINE(NF,XF,RF,SF,NO,XSET,SO,RFS,XFSP,RFSP,XFSP,RFSP,2)
IF(K.GT.NO) K=NO
IF(Y(J).GE.RFS(K)) GO TO 55
CALC=SQRT(RFS(K)*RFS(K)-Y(J)*Y(J))
ZU(I,J)=ZFS(K)+CALC
ZL(I,J)=ZFS(K)-CALC
DRFDXF=RFSP(K)/XFSP(K)
DZFDXF=ZFSP(K)
DZDXU(I,J)=DZFDXF+RFS(K)*DRFDXF/CALC
DZDXL(I,J)=DZFDXF-RFS(K)*DRFDXF/CALC
DZDYU(I,J)=-Y(J)/CALC
DZDYL(I,J)=Y(J)/CALC
54 CONTINUE
55 JBOD(I)=J
56 CONTINUE
DO 124 I=ILE,ITE
124 JBOD(I)=JTIP+1

C
C
WING COORDINATES

DO 150 JS=1,JSECT
NU=NSU(JS)
DO 140 N=1,NU
XF(N)=XSU(N,JS)
RF(N)=ZSU(N,JS)
140 CONTINUE
DO 144 I=ILE,ITE
XSET(I)=(X(I,1)-XLE(1))/CHORD(1)
144 CONTINUE
NO=ITE-ILE+1
DO 146 I=ILE,ITE
XSET(I-ILE+1)=XSET(I)
146 CONTINUE
CALL ASPLINE(NSU(JS),XF,RF,SF,NO,XSET,SO,RFS,XFSP,RFSP,XFSP,RFSP,2)
DO 148 I=1,NO
ZSU(ILE+I-1,JS)=RFS(I)
XSU(ILE+I-1,JS)=RFSP(I)/XFSP(I)

```

```

148 CONTINUE
150 CONTINUE
DO 154 I=ILE,ITE
  JSECTX=JSECT-1
DO 154 JS=1,JSECTX
  DO 153 J=JROOT,JTIP
  IF (YS(I).EQ.Y(J).OR.(YS(JS).LT.Y(J).AND.Y(J).LE.YS(JS+1)))
  I GO TO 152
  GO TO 153
152 CALC=(ZSU(I,JS)-ZSU(I,JS+1))/(YS(JS)-YS(JS+1))
  ZU(I,J)=ZSU(I,JS)+CALC*(Y(J)-YS(JS))
  ZU(I,J)=ZU(I,J)*CHORD(J)
  CALC=(XSU(I,JS)-XSU(I,JS+1))/(YS(JS)-YS(JS+1))
  DZDXU(I,J)=XSU(I,JS)+(Y(J)-YS(JS))*CALC
153 CONTINUE
154 CONTINUE
C
DO 160 JS=1,JSECT
  NL=NSL(JS)
DO 155 N=1,NL
  XF(N)=XSL(N,JS)
  RF(N)=ZSL(N,JS)
155 CONTINUE
DO 156 I=ILE,ITE
  XSET(I)=(X(I,1)-XLE(1))/CHORD(1)
156 CONTINUE
  NO=ITE-ILE+1
DO 157 I=ILE,ITE
  XSET(I-ILE+1)=XSET(I)
157 CONTINUE
  CALL ASPLINE(NSL(JS),XF,RF,SF,NO,XSET,SO,RFS,XFSP,RFSP,XFSPP,RFSP
  1,3)
DO 158 I=1,NO
  ZSL(ILE+I-1,JS)=RFS(I)
  XSL(ILE+I-1,JS)=RFSP(I)/XFSP(I)
158 CONTINUE
160 CONTINUE
DO 164 I=ILE,ITE
  JSECTX=JSECT-1
DO 164 JS=1,JSECTX
  DO 163 J=JROOT,JTIP
  IF (YS(I).EQ.Y(J).OR.(YS(JS).LT.Y(J).AND.Y(J).LE.YS(JS+1)))
  I GO TO 162
  GO TO 163
162 CALC=(ZSL(I,JS)-ZSL(I,JS+1))/(YS(JS)-YS(JS+1))
  ZL(I,J)=ZSL(I,JS)+CALC*(Y(J)-YS(JS))
  ZL(I,J)=ZL(I,J)*CHORD(J)
  CALC=(XSL(I,JS)-XSL(I,JS+1))/(YS(JS)-YS(JS+1))
  DZDXL(I,J)=XSL(I,JS)+(Y(J)-YS(JS))*CALC
163 CONTINUE
164 CONTINUE
C
DO 172 I=ILE,ITE
DO 172 J=JROOT,JTIP
  CALC=-DXLEDY(J)/CHORD(J)
  CALC=CALC-(XCAP(I)+.5)*DCDY(J)/CHORD(J)
  CALC=CALC*CHORD(J)
  CALC=CALC*DZDXU(I,J)
  IF (J.EQ.JB1.OR.J.EQ.JB2.OR.J.EQ.JTIP) GO TO 170
  DZDYU(I,J)=CALC+(ZU(I,J)-ZU(I,J+1))/(Y(J)-Y(J+1))
  GO TO 172
170 DZDYU(I,J)=CALC+(ZU(I,J)-ZU(I,J-1))/(Y(J)-Y(J-1))
172 CONTINUE
C
DO 182 I=ILE,ITE
DO 182 J=JROOT,JTIP
  CALC=-DXLEDY(J)/CHORD(J)
  CALC=CALC-(XCAP(I)+.5)*DCDY(J)/CHORD(J)
  CALC=CALC*CHORD(J)
  CALC=CALC*DZDXL(I,J)
  IF (J.EQ.JB1.OR.J.EQ.JB2.OR.J.EQ.JTIP) GO TO 180
  DZDYL(I,J)=CALC+(ZL(I,J)-ZL(I,J+1))/(Y(J)-Y(J+1))

```

```

GO TO 182
180 DZDYL(I,J)=CALC*(ZL(I,J)-ZL(I,J-1))/(Y(J)-Y(J-1))
182 CONTINUE

```

C

```

WRITE (6,249)
DO 250 I=1,IM
250 WRITE (6,251) I,JHOD(I)
RETURN
END

```

```

SUBROUTINE ASPLINE(ND,XD,YD,SD,NU,XO,SO,YO,XP,YP,XPP,YPP,NFL)

```

C

```

PARAMETERIZES (XD,YD) DATA IN TERMS OF ARC-LENGTH SD ALONG CURVE,
THEN SPLINE FITS. XP,YP,XPP,YPP ARE DERIVATIVES W.R.T. SD.
NFL.EQ.1 FOR DX/DS=0 (INFINITE DY/DX AT LEFT END)
NFL.EQ.2 FOR DY/DS=+1 (POSITIVE DY/DX AT LEFT END)
NFL.EQ.3 FOR DY/DS=-1 (NEGATIVE DY/DX AT LEFT END)

```

C

```

DIMENSION XD(ND),YD(NU),XO(NU),YO(NU),SD(ND),SO(NU)
DIMENSION XP(NU),YP(NU),XPP(NU),YPP(NU)

```

C

```

EPSU=1.E-10
NI=ND-1
SD(1)=0
H1=0
UX1=XD(2)-XD(1)
DY1=YD(2)-YD(1)
C1=SQRT(DX1**2+DY1**2)
SD(2)=C1
IF(NFL.EQ.2)RETURN
DO 1 I=2,NI
DX1=XD(I)-XD(I-1)
DY1=YD(I)-YD(I-1)
DX2=XD(I+1)-XD(I)
DY2=YD(I+1)-YD(I)
DX=XD(I+1)-XD(I-1)
DY=YD(I+1)-YD(I-1)
C2=SQRT(DX2**2+DY2**2)
C=SQRT(UX**2+UY**2)
A=(DY1*DX-DX1*DY)/2
H=4*A/(C*C1*C2)
HAV=(H1+H)/2
DS=C1*(1+(C1/2*HAV)**2/6)
SD(I)=SD(I-1)+DS
C1=C2
H1=H
1 CONTINUE
DS=C1*(1+(C1/2*H)**2/6)
SD(ND)=SD(ND-1)+DS
CALL SSPLINE(ND,SD,XD,NO,SO,XO,XP,XPP,1)
CALL SSPLINE(ND,SD,YD,NO,SO,YO,YP,YPP,NFL)
RETURN
END

```

1

```

SUBROUTINE SSPLINE(ND,XD,YD,NO,X,Y,YP,YPP,NSWITCH)

```

C

```

SPLINE FITS ND DATA POINTS (XD,YD). ASSUMES CURVIC RIGHT END.
LOCAL ARRAYS D1Y,D2Y,D3Y MUST BE APPROPRIATELY DIMENSIONED.
NSWITCH.EQ.1 FOR CALL FROM ASPLINE WITH DX/DS=0
NSWITCH.EQ.2 FOR CALL FROM ASPLINE WITH DY/DS=+1
NSWITCH.EQ.3 FOR CALL FROM ASPLINE WITH DY/DS=-1
NSWITCH.EQ.4 FOR DIRECT CALL AND CUBIC LEFT END

```

C

```

DIMENSION XD(ND),YD(ND),X(NO),Y(NO),YP(NO),YPP(NO)
)DIMENSION D1Y(150),D2Y(150),D3Y(150)

```

C

```

EPSI1=-1.E-10
EPSI2=-EPSI1
NIM1=ND-1
DX=XD(2)-XD(1)
IF(DX.EQ.0.) GO TO 35
DF=(YD(2)-YD(1))/DX
D1Y(1)=.5
I2=2
GO TO (1,2,3,4),NSWITCH
1  D2Y(1)=3*(DF-0)/DX
   GO TO 6
2  D2Y(1)=3*(DF-1)/DX
   GO TO 6
3  D2Y(1)=3*(DF+1)/DX
   GO TO 6
4  I2=3
   DX1=XD(3)-XD(2)
   DF1=(YD(3)-YD(2))/DX1
   C=(DX1**2-DX**2)/DX1
   H=(DX+DX1)*(DX+2*DX1)/DX1
   D1Y(2)=C/H
   D2Y(2)=6*(DF1-DF)/H
   DX=DX1
   DF=DF1
6  DO 7 I=I2,NIM1
   DX1=XD(I+1)-XD(I)
   IF(DX1.EQ.0.) GO TO 36
   DF1=(YD(I+1)-YD(I))/DX1
   H=2*(DX+DX1)
   F=6*(DF1-DF)
   DENOM=H-DX*D1Y(I-1)
   D2Y(I)=(F-DX*D2Y(I-1))/DENOM
   D1Y(I)=DX1/DENOM
   DX=DX1
   DF=DF1
7  CONTINUE
   DX1=XD(ND-1)-XD(ND-2)
   CALC=(DX+DX1)*(DX+DX1*2)/DX1
   CALC1=(DX1**2-DX**2)/DX1
   DENOM=(CALC-D1Y(ND-2)*CALC1)
   D2Y(ND-1)=(F-D2Y(ND-2)*CALC1)/DENOM
   NIM2=ND-2
   DO 8 I=1,NIM2
   K=ND-I-1
   IF(NSWITCH.EQ.4.AND.K.EQ.1) GO TO 8
   D2Y(K)=D2Y(K)-D1Y(K)*D2Y(K+1)
8  CONTINUE
   D2Y(ND)=((DX+DX1)*D2Y(ND-1)-DX*D2Y(ND-2))/DX1
   IF(NSWITCH.NE.4) GO TO 9
   DX=XD(2)-XD(1)
   DX1=XD(3)-XD(2)
   D2Y(1)=((DX+DX1)*D2Y(2)-DX*D2Y(3))/DX1
9  K=ND
   DO 11 I=1,NIM1
   K=K-1
10  DX1=XD(K+1)-XD(K)
   DF1=(YD(K+1)-YD(K))/DX1
   D1Y(K+1)=DF1+DX1/6*(D2Y(K)+2*D2Y(K+1))
   D3Y(K+1)=(D2Y(K+1)-D2Y(K))/DX1
11  CONTINUE
   D1Y(1)=DF1-DX1/6*(2*D2Y(1)+D2Y(2))
   D3Y(1)=D3Y(2)
   IF(NSWITCH.EQ.1) GO TO 16
C
C
C  INTERPOLATING Y
DO 15 J=1,NO

```

```

DO 12 I=1,ND
DX=XD(I)-X(J)
IF (DX.GE.EPSI1.AND.DX.LE.EPSI2) GO TO 13
IF (DX.GE.EPSI2) GO TO 14
12 CONTINUE
GO TO 37
13 Y(J)=YU(I)
YP(J)=D1Y(I)
YPP(I)=D2Y(I)
GO TO 15
14 DX=X(J)-XD(I)
Y(J)=YD(I)+DX*(D1Y(I)+DX/2*(D2Y(I)+DX/3*D3Y(I)))
YP(J)=D1Y(I)+DX*(D2Y(I)+DX/2*D3Y(I))
YPP(J)=D2Y(I)+DX*D3Y(I)
15 CONTINUE
GO TO 23
C
C C
INTERPOLATING X
16 CONTINUE
DO 22 J=1,NO
DO 17 I=1,ND
DY=YD(I)-Y(J)
IF (DY.GE.EPSI1.AND.DY.LE.EPSI2) GO TO 18
IF (DY.GE.EPSI2) GO TO 19
17 CONTINUE
GO TO 38
18 Y(J)=YD(I)
X(J)=XD(I)
YP(J)=D1Y(I)
YPP(J)=D2Y(I)
GO TO 22
19 DX=-DY/D1Y(I)
20 YO=YD(I)+DX*(D1Y(I)+DX/2*(D2Y(I)+DX/3*D3Y(I)))
DY=YO-Y(J)
IF (DY.GE.EPSI1.AND.DY.LE.EPSI2) GO TO 21
YOP=D1Y(I)+DX*(D2Y(I)+DX/2*D3Y(I))
DELX=-DY/YOP
DX=DX+DELX
GO TO 20
21 X(J)=XD(I)+DX
YP(J)=D1Y(I)+DX*(D2Y(I)+DX/2*D3Y(I))
YPP(J)=D2Y(I)+DX*D3Y(I)
22 CONTINUE
23 CONTINUE
RETURN
35 WRITE (6,100)
WRITE (6,101) XD(1),XD(2)
STOP
36 WRITE (6,100)
WRITE (6,102) I,XD(I),XD(I+1)
STOP
37 WRITE (6,100)
WRITE (6,103) J,X(J),XD(ND)
STOP
38 WRITE (6,100)
WRITE (6,104) J,Y(J),YD(ND)
STOP
C
100 FORMAT (/5X,18HSUBROUTINE SSPLINE/)
101 FORMAT (/5X,21HERROR IN INPUT XD(1)=E12.4,5X,6HXD(2)=E12.4/)
102 FORMAT (/5X,16HERROR IN INPUT I=15,5X,6HXD(I)=E12.4,5X,
1 8HXD(I+1)=E12.4/)
103 FORMAT (/5X,23HX(J) IS OUT OF RANGE J=15,5HX(J)=E12.4,5X,
1 7HXD(ND)=E12.4/)
104 FORMAT (/5X,23HY(J) IS OUT OF RANGE J=15,5X,5HY(J)=E12.4,5X,
1 7HYD(ND)=E12.4/)
C
END

```

SUBROUTINE PROFL

CALCULATES WING/BODY AND GRID DATA IN THE XI-ETA-ZETA COMPUTATION DOMAIN.

DIMENSION ZETAL(49,19),ZETAU(49,19)

```

COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,A12,CSA,XX1,RX2,
* SNA,TDETA,TOXI,TUZETA,DETA,OXI,DZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
COMMON/COORD/ CHORD(19),DCDY(19),DUDY(19),DXLEDY(19),ETA(19),F(49)
* ,FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
* TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
* ZCAP(25),ZLE(19),DZLEDY(19),ALPHAT(19),DADY(19),GWT
COMMON/DATAX/ A1,A2,A3,A4,A5,ETA0,NETA,NXI,NZETA,XCAP0,XID,YCAP0,
* NF,XF(150),ZF(150),RF(150),JSECT,JBL,JBT,NSU(5),
* NSL(5),YS(5),XLES(5),ZLES(5),CS(5),ATS(5),TS(5),
* FS(5),XSU(150,5),XSL(150,5),ZSU(150,5),ZSL(150,5)
COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZDXL(49,19),
* DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
* HWBU(49,19),IZL(19),IZU(19),JBOD(49),KBODL(49,19),
* KBODU(49,19),ZL(49,19),ZU(49,19)

```

```

304 FORMAT (1H-///1X28HSURFACE-ADJACENT GRID POINTS///
1 1X33HUPPER SURFACE (I DOWN, J ACROSS)///
2 3X1HI,3X10HKRODU(I,J)/)
306 FORMAT (I4,(20I6))
309 FORMAT (///1X33HLOWER SURFACE (I DOWN, J ACROSS)///3X1HI,3X,
1 10HKRODL(I,J)/)
459 FORMAT (1H-///1X33HSTATIONS OF ZERO STREAMWISE SLOPE//3X1HJ,9X,
1 6HIZU(J),9X6HIZL(J)/)
461 FORMAT (I4,2I15)
601 FORMAT (1H-///1X40HNEGATIVE UPPER-SURFACE GRID-POINT OFFSET,
1 24H AT PLANFORM STATION I = I4,5H, J = I4/)
602 FORMAT (1H-///1X40HNEGATIVE LOWER-SURFACE GRID-POINT OFFSET,
1 24H AT PLANFORM STATION I = I4,5H, J = I4/)

```

```

C
C
C
HPI=2.*ATAN(1.)
DO 100 I=1,IM
DO 100 J=1,JM
ZETAU(I,J)=0.
100 ZETAL(I,J)=0.
DO 120 I=1,IM
DO 120 J=1,JM
TC=TAU(J)*CHORD(J)
ZH=ZU(I,J)/TC
ZETAU(I,J)=ATAN(ZH/A5)/HPI
ZH=ZL(I,J)/TC
120 ZETAL(I,J)=ATAN(ZH/A5)/HPI
C
AH=A5*HPI
DO 200 I=1,IM
DO 200 J=1,JM
ZH=ZETAU(I,J)
TN1=TAN(HPI*ZH)
HWBU(I,J)=1./(AH*(1.+TN1*TN1))
ZH=ZETAL(I,J)
TN1=TAN(HPI*ZH)
200 HWBL(I,J)=1./(AH*(1.+TN1*TN1))
C
DO 270 I=1,IM
DO 270 J=1,JM
KHODU(I,J)=0.
270 KHODL(I,J)=0.
DO 300 I=1,IM

```



```

      DO 300 J=1,JM
      ZH=ZETAU(I,J)
      IF (ZH.EQ.0.) GO TO 285
      DO 280 K=KW,KM
      IF (ZH.GE.ZETA(K)) GO TO 280
      KR0DU(I,J)=K
      GO TO 285
280 CONTINUE
285 ZH=ZETAL(I,J)
      IF (ZH.EQ.0.) GO TO 295
      DO 290 K=1,KW
      KK=KW-K+1
      IF (ZH.LE.ZETA(KK)) GO TO 290
      KBODL(I,J)=KK
      GO TO 295
290 CONTINUE
295 CONTINUE
300 CONTINUE
      WRITE (6,304)
      DO 305 I=1,IM
305 WRITE (6,306) I,(KR0DU(I,J),J=1,JM)
      WRITE (6,309)
      DO 310 I=1,IM
310 WRITE (6,306) I,(KBODL(I,J),J=1,JM)
C
      DO 400 J=1,JM
      IZU(J)=0.
      IZL(J)=0.
400 NFX=NF-1
      DO 365 N=1,NFX
      IF (RF(N+1).LE.RF(N)) GO TO 370
365 CONTINUE
370 XF1=XF(N)
      DO 375 I=INOSL,ITAIL
      IF (X(I,1).GE.XF1) GO TO 380
375 CONTINUE
380 IFS=I
      JR00TX=JR00T-1
      DO 410 J=1,JR00TX
      IZU(J)=IFS
410 IZL(J)=IFS
      DO 450 J=JR00T,JTIP
      DO 420 I=ILE,ITE
      IF (DZDXU(I,J).GT.0.) GO TO 420
      IZU(J)=I
      GO TO 425
420 CONTINUE
425 DO 430 I=ILE,ITE
      IF (DZDXL(I,J).LT.0.) GO TO 430
      IZL(J)=I
      GO TO 450
430 CONTINUE
450 CONTINUE
      WRITE (6,459)
      DO 460 J=1,JM
460 WRITE (6,461) J,IZU(J),IZL(J)
C
      DO 500 I=1,IM
      DO 500 J=1,JM
      DELU(I,J)=0.
500 DELL(I,J)=0.
      DO 550 I=1,IM
      DO 550 J=1,JM
      KX=KR0DU(I,J)
      IF (KX.EQ.0) GO TO 525
      DELU(I,J)=ZETA(KX)-ZETAU(I,J)
525 KX=KBODL(I,J)
      IF (KX.EQ.0) GO TO 550
      DELL(I,J)=ZETAL(I,J)-ZETA(KX)
550 CONTINUE
      DO 600 I=1,IM
      DO 600 J=1,JM

```

```

        IF (DELU(1,J).GE.0.) GO TO 590
        WRITE (6,601) I,J
        STOP
590    IF (DELL(1,J).GE.0.) GO TO 600
        WRITE (6,602) I,J
        STOP
600    CONTINUE
C
        DO 700 I=1,IM
700    DELS(I)=0.0
        DO 710 I=INOSE,ITAIL
710    DELS(I)=GAMF
        DO 720 I=ILE,ITE
720    DELS(I)=GAMWT
        IRX=IFS-1
        XF1=X(IFS,1)
        ALC3=0.5*(3.*YCAPO/ETA0-MPI*A3)
        ALC4=0.5*(MPI*A3-YCAPO/ETA0)/(ETA0*ETA0)
        YFS=RF(1)
        DO 722 N=2,NF
722    IF (YFS.LT.RF(N)) YFS=RF(N)
        CONTINUE
        DO 790 I=INOSE,IFS
        J2=JROU(I)
        J1=J2-1
        X1=X(I,J1)
        X2=X(I,J2)
        Y1=Y(J1)
        Y2=Y(J2)
        SLOPE=(YFS-Y1)/(Y2-Y1)
        XTEST=X1+SLOPE*(X2-X1)
        IF (XTEST.GE.XF1) GO TO 800
        XXX=((X1-XF1)/XF1)**2
        YF1=YFS*SQRT(1.-XXX)
        XXX=((X2-XF1)/XF1)**2
        YF2=YFS*SQRT(1.-XXX)
        RATIO=(Y2-Y1)/(YF2-YF1)
        YE=(Y1-RATIO*YF1)/(1.-RATIO)
        YCAPE=YE/BSPAN
        ETA1=ETA(J2)
730    ETALSQ=ETA1*ETA1
        FUN=ETA1*(ALC3+ALC4*ETALSQ)-YCAPE
        FUNPR=ALC3+3.*ALC4*ETALSQ
        DEL=FUN/FUNPR
        ETA2=ETA1-DEL
        IF (DEL.LT.1.E-10) GO TO 750
        ETA1=ETA2
        GO TO 730
750    ETAE=ETA2
        GAM=ABS(ETA(J2)-ETAE)
        DELS(I)=GAM
790    CONTINUE
800    CONTINUE
        RETURN
        END

```

SUBROUTINE COEFF

CALCULATES FIXED GEOMETRY- AND GRID-RELATED QUANTITIES WHICH APPEAR IN THE TRANSFORMED POTENTIAL EQUATION.

C
C
C
C
C

```

COMMON/CAPS / CAPG(49,19),CAPGT1(19,25),CAPGT2(49,19),CAPI(19),
* CAPH(49,9),CAPHT1(19,25),CAPHT2(49,19),CAPIB(19),
* CAPGB(4,9),CAPIT(49,19)
COMMON/COEF / A1(49),A2(49),A3(49),A5(49),B1(19),B2(19),B3(19),
* B5(19),B7(19),C1(25),C2(25),C3(25),C5(25),C7(25)

```

```

* COMMON/CONST/ D1,D2,E1,E2,F1,F2
ALPHA,ZMACH,DPLIM,EPSE,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
SNA,TDETA,TDXI,TDZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
PLCB2,PLCB3
* COMMON/COURD/ CHORD(19),DCDY(19),DUDY(19),DXLEDY(19),ETA(19),F(49)
,FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
* TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
* ZCAP(25),ZLE(19),DZLEDY(19),ALPHAT(19),DADY(19),GWT

```

COC

```

PI=4.*ATAN(1.)
ILX=ILE-1
ITP=ITE+1
DO 200 J=1,JM
CC=CHORD(J)
TT=TAU(J)
TC=IT*CC
DXLY=DXLEDY(J)
DZLY=DZLEDY(J)
DATY=DADY(J)*PI/180.
CIS=-DCDY(J)/CC
CTS=-DUDY(J)/IT
CAPI(J)=CIS/CC
CAPIR(J)=(CIS*CTS)/TC
DO 100 K=1,KM
ZZ=ZCAP(K)
CAPGT1(J,K)=ZZ*(CIS+CTS)
100 CAPHT1(J,K)=2.*ZZ*(CIS*CIS+CTS*CTS+CIS*CTS)
DO 120 I=1,IM
CAPG(I,J)=(XCAP(I)+0.5)*CIS-DXLY/CC
120 CAPH(I,J)=2.*CIS*CAPG(I,J)
ANGLE=ALPHAT(J)*PI/180.
TNA=TAN(ANGLE)
SCA=1./COS(ANGLE)
DO 140 I=1,ILX
CAPGB(I,J)=0.
CAPGT2(I,J)=-DZLY/TC
CAPHT2(I,J)=2.*CAPGT2(I,J)*(CIS+CTS)
140 CAPIT(I,J)=0.
DO 160 I=1,ITE
XX=XCAP(I)+0.5
CAPGB(I,J)=TNA/TC
CAPGT2(I,J)=-((DZLY-CC*XX*SCA*SCA*DADY+DXLY*TNA)/TC
CAPHT2(I,J)=-2.*SCA*SCA*DADY*(DXLY-CC*XX*TNA*DADY)/TC
+2.*CAPGT2(I,J)*(CIS+CTS)
160 CAPIT(I,J)=SCA*SCA*DADY/TC+CAPGB(I,J)*(CIS+CTS)
DO 180 I=ITP,IM
CAPGR(I,J)=0.
CAPGT2(I,J)=-((DZLY-CC*SCA*SCA*DADY+DXLY*TNA)/TC
CAPHT2(I,J)=2.*SCA*SCA*DADY*(DCDY(J)+CC*TNA*DADY)/TC
+2.*CAPGT2(I,J)*(CIS+CTS)
180 CAPIT(I,J)=0.
200 CONTINUE
C
DX=2.*DXI*DXI
DE=2.*DETA*DETA
DZ=2.*DZETA*DZETA
A1(1)=0.
A1(IM)=0.
A2(1)=0.
A2(IM)=0.
A3(1)=0.
A3(IM)=0.
IMX=IM-1
DO 210 I=2,IMX
A1(I)=(F(I+1)+F(I))/DX
A3(I)=(F(I)+F(I-1))/DX
210 A2(I)=-A1(I)-A3(I)

```

```

A5(1)=0.
A5(2)=0.
DO 220 I=3,IM
C 220 A5(I)=(F(I-1)+F(I-2))/DX
      B1(1)=(G(1)+G(2))/DE
      B3(1)=(G(1)+G(2))/DE
      B2(1)=-B1(1)-B3(1)
      H1(JM)=0.
      B2(JM)=0.
      B3(JM)=0.
      JMX=JM-1
      DO 310 J=2,JMA
      B1(J)=(G(J+1)+G(J))/DE
      B3(J)=(G(J)+G(J-1))/DE
310   B2(J)=-B1(J)-B3(J)
      H5(1)=(G(2)+G(3))/DE
      H5(2)=(G(1)+G(2))/DE
      DO 320 J=3,JM
320   H5(J)=(G(J-1)+G(J-2))/DE
      JMX2=JM-2
      DO 330 J=1,JMX2
330   H7(J)=(G(J+1)+G(J+2))/DE
      H7(JM-1)=0.
      H7(JM)=0.

```

```

C
      C1(1)=0.
      C1(KM)=0.
      C2(1)=0.
      C2(KM)=0.
      C3(1)=0.
      C3(KM)=0.
      KMX=KM-1
      DO 410 K=2,KMX
      C1(K)=(H(K+1)+H(K))/DZ
      C3(K)=(H(K)+H(K-1))/DZ
410   C2(K)=-C1(K)-C3(K)
      C5(1)=0.
      C5(2)=0.
      DO 420 K=3,KM
420   C5(K)=(H(K-1)+H(K-2))/DZ
      KMX2=KM-2
      DO 430 K=1,KMX2
430   C7(K)=(H(K+1)+H(K+2))/DZ
      C7(KM-1)=0.
      C7(KM)=0.

```

```

C
      D1=1./(4.*DXI*DETA)
      D2=-D1
      F1=1./(4.*DETA*DZETA)
      F2=-F1
      F1=1./(4.*DXI*DZETA)
      F2=-F1
      RETURN
      END

```

```

C
C
C
C
SUBROUTINE INIT(MSTART)
INITIALIZES SOLUTION ARRAYS.

```

```

COMMON/CONST/ ALPHA,7MACH,DPLIM,EPST,WE,WG,HSPAN,AI2,CSA,RX1,RX2,
* SNA,TDETA,TDXI,TDZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROUT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
COMMON/SOLV0/ DL1(49,16),DL2(49,16),DS1(49),DS2(49),DU1(49,16),
* DU2(49,16),DPMAX,UPMAXI,GAMMA(16),IMAXI,ITERI,
* JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,16,25),PLE1(16),

```

```

* PNEW(25),PNOSE,PSYM(49,25),PT1(16,25),PT2(16,25),
* PT3(16,25),PTE1(16),PTE2(16),PWAL(49,16),PWBU(49,16)
COMMON/SURF / DELL(49,16),DELS(49),DELU(49,16),DZDXL(49,16),
* DZDXU(49,16),DZDYL(49,16),DZUYU(49,16),HWBL(49,16),
* HWBU(49,16),IZL(16),IZU(16),JHOD(49),KHODL(49,16),
* KRODU(49,16),ZL(49,16),ZU(49,16)

```

C
C
C

```

IF (MSTART.NE.0) GO TO 105
DO 100 I=1,IM
DO 100 J=1,JM
DO 100 K=1,KM
100 P(I,J,K)=0.
105 DO 110 J=1,JM
DO 110 K=1,KM
PT1(J,K)=0.
PT2(J,K)=0.
110 PT3(J,K)=0.
DO 120 I=1,IM
DO 120 J=1,JM
UL1(I,J)=0.
DL2(I,J)=0.
PWBL(I,J)=0.
DU1(I,J)=0.
DU2(I,J)=0.
120 PWBU(I,J)=0.
DO 130 I=1,IM
DS1(I)=0.
130 DS2(I)=0.
DO 140 J=1,JM
PLE1(J)=0.
PTE1(J)=0.
140 PTE2(J)=0.
PNOSE=0.
DO 145 I=1,IM
DO 145 K=1,KM
145 PSYM(I,K)=0.
IF (MSTART.NE.0) GO TO 300
DO 150 J=1,JM
150 GAMMA(J)=0.

```

C

```

300 PI=4.*ATAN(1.)
ANGLE=ALPHA*PI/180.
CSA=COS(ANGLE)
SNA=SIN(ANGLE)
RX1=1./WE
RX2=1.-RX1
TOXI=2.*DXI
TDETA=2.*DETA
TDZETA=2.*DZETA
AI2=1./(ZMACH*ZMACH)
IF (MSTART.EQ.0) GO TO 900
DO 800 I=1,NOSE,IM
JEX=JHOD(I)-1
DO 750 J=1,JEX
KL=KRODL(I,J)+1
KU=KRODU(I,J)-1
DO 700 K=KL,KU
700 P(I,J,K)=J.
750 CONTINUE
800 CONTINUE
900 RETURN
END

```

C
C

SUBROUTINE SOLVE(ITER)

EXECUTES A COMPLETE RELAXATION SWEEP OVER THE COMPUTATION DOMAIN.

C
C
C

DIMENSION SLKW1(49,19),SL2(19),SS2(19),SU2(19)
DIMENSION SUKW1(49,19),SL3(19),SS3(19),SU3(19)

C

```
COMMON/CAPS / CAPG(49,19),CAPGT1(19,25),CAPGT2(49,19),CAPI(19),
CAPH(49,19),CAPHT1(19,25),CAPHT2(49,19),CAPIB(19),
* CAPGB(49,19),CAPIT(49,19)
COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
* SNA,TDETA,TDXI,TUZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
COMMON/COORD/ CHORD(19),DCDY(19),DTDY(19),DXLEDY(19),ETA(19),F(49)
* FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
* TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
* ZCAP(25),ZLE(19),UZLEDY(19),ALPHAT(19),DADY(19),GWT
COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
* DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERI,
* JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
* PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
* PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)
COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZDXL(49,19),
* DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
* HWBU(49,19),IZL(19),IZU(19),JB00(49),KB00L(49,19),
* KB00U(49,19),ZL(49,19),ZU(49,19)
```

C
C
C
C
C

UPDATE SURFACE CONDITION

```
UPMAX=0.0
NSUP=0
ILX=ILE-1
IMX=IM-1
INX=INOSE-1
ITP=ITE+1
JMX=JM-1
KMX=KM-1
DH=-DXI
DG=-GAMN
C1=2.*DH*UH/(UH+2.*DG)
C2=4.*DG/(DH+2.*DG)
C3=(DH-2.*DG)/(DH+2.*UG)
PW=P(INOSE-1,1,KW)
PWW=P(INOSE-2,1,KW)
PNOSE=C1*CHORD(1)*CSA/FN+C2*PW+C3*PWW
```

C

```
DG=-GAMLE
C1=2.*DH*UH/(UH+2.*DG)
C2=4.*UG/(DH+2.*DG)
C3=(DH-2.*DG)/(DH+2.*UG)
CF=CSA/FLE
DO 20 J=JROOT,JTIP
PW=P(ILE-1,J,KW)
PWW=P(ILE-2,J,KW)
20 PLE1(J)=C1*CHORD(J)*CF+C2*PW+C3*PWW
PLE1(JROOT-1)=2.*PLE1(JROOT)-PLE1(JROOT+1)
```

C

```
DO 30 MSURF=1,2
DO 30 ISET=INOSF,IM
JE=JB00(ISET)-1
DO 30 JSFT=1,JE
30 CALL SURFBC(ISET,JSFT,MSURF)
DO 40 ISET=INOSF,IM
JSET=JB00(ISET)
40 CALL SURFSC(ISET,JSET,0)
CALL SYMBC(0,0,0)
```

C

```
DH=DXI
DG=GAMTE
C1=2.*DH*UH/(UG*(DH+DG))
```

```

C2=-2.*(DH-DG)/DG
C3=(DH-UG)/(DH+DG)
DO 50 J=JROOT,JTIP
PTE=(2.-DG/DH)*PWHL(ITE,J)-(1.-DG/DH)*PWHL(ITE-1,J)
PE=P(ITE+1,J,KW)
PEE=P(ITE+2,J,KW)
PTE1(J)=C1*PTE+C2*PE+C3*PEE
50 PTE2(J)=3.*(PTE1(J)-PE1+PEE)
PTE1(JROOT-1)=2.*PTE1(JROOT)-PTE1(JROOT+1)

```

C
C
C

COMPUTE POTENTIAL: REGION UPSTREAM OF FUSELAGE NOSE

```

P(INOSE,1,KW)=PNOSE
IF (AI2.GT.1.) GO TO 103
DO 102 J=1,JM
DO 102 K=1,KM
PT3(J,K)=P(1,J,K)
102 PT2(J,K)=P(2,J,K)
ISTART=3
GO TO 105
103 DO 104 J=1,JM
DO 104 K=1,KM
104 PT2(J,K)=P(1,J,K)
ISTART=2
105 DO 150 ISET=ISTART,INX
DO 120 JSET=1,JMX
CALL TRICUE(2,KMX,ISET,JSET)
CALL INVERT(2,KMX)
CALL MAXI(2,KMX,ISET,JSET)
DO 115 K=2,KMX
PT1(JSET,K)=P(ISET,JSET,K)
115 P(ISET,JSET,K)=PNEW(K)
PT1(JSET,1)=P(ISET,JSET,1)
PT1(JSET,KM)=P(ISET,JSET,KM)
120 CONTINUE
DO 125 K=1,KM
125 PT1(JM,K)=P(ISET,JM,K)
DO 130 J=1,JM
DO 130 K=1,KM
PT3(J,K)=PT2(J,K)
130 PT2(J,K)=PT1(J,K)
150 CONTINUE

```

C
C
C

COMPUTE POTENTIAL: REGION DOWNSTREAM OF FUSELAGE NOSE

```

DO 200 J=1,JTIP
200 P(ILE,J,KW)=PLE1(J)
DO 210 J=1,JM
SU2(J)=PT2(J,KW)
SU3(J)=PT3(J,KW)
SL2(J)=PT2(J,KW)
SL3(J)=PT3(J,KW)
SS2(J)=PT2(J,KW)
210 SS3(J)=PT3(J,KW)
PSAVE=P(ILE,JROOT-1,KW)

```

C

```

DO 500 ISET=INOSF,IMX
ISETX=ISET-1
ISETP=ISET+1
JE2=JBOD(ISETX)-1
JE1=JBOD(ISET)-1
JF0=JBOD(ISETP)-1
IF (ISET.EQ.I LX) JF0=JE1
IF (ISET.EQ.ITP) JE2=JE1

```

C
C
C

LINE SEGMENTS BELOW WING/HOUY

```

DO 220 J=1,JM
PT2(J,KW)=SL2(J)
220 PT3(J,KW)=SL3(J)
IF (JE2.EQ.0) GO TO 230
DO 225 J=1,JE2

```

```

SUKW1(ISETX,J)=P(ISFTX,J,KW+1)
KH=KRODL(ISETA,J)
P(ISFTX,J,KH+1)=DL1(ISETX,J)
225 P(ISFTX,J,KH+2)=DL2(ISETX,J)
230 CONTINUE
DO 235 J=1,JE1
SUKW1(ISET,J)=P(ISET,J,KW+1)
KH=KRODL(ISET,J)
P(ISET,J,KH+1)=DL1(ISET,J)
235 P(ISET,J,KH+2)=DL2(ISET,J)
DO 240 J=1,JE0
SUKW1(ISETP,J)=P(ISETP,J,KW+1)
KH=KHODL(ISETP,J)
P(ISETP,J,KH+1)=DL1(ISETP,J)
240 P(ISETP,J,KH+2)=DL2(ISETP,J)
DO 250 JSET=1,JE1
KH=KRODL(ISET,JSET)
CALL TRICUE(2,KH,ISET,JSET)
CALL INVERT(2,KH)
CALL MAXI(2,KH,ISET,JSET)
DO 245 K=2,KH
PT1(JSET,K)=P(ISET,JSET,K)
245 P(ISET,JSET,K)=PNEW(K)
PT1(JSET,1)=P(ISET,JSET,1)
KB1=KB+1
DO 247 K=KB1,KW
247 PT1(JSET,K)=P(ISET,JSET,K)
CALL SURFBC(ISET,JSET,1)
P(ISET,JSET,KB+1)=DL1(ISET,JSET)
P(ISET,JSET,KB+2)=DL2(ISET,JSET)
250 CONTINUE
DO 255 J=1,JE1
SL3(J)=SL2(J)
255 SL2(J)=PT1(J,KW)

```

C
C
C

```

LINE SEGMENTS ABOVE WING/HODY
DO 260 J=1,JM
PT2(J,KW)=SU2(J)
260 PT3(J,KW)=SU3(J)
IF (JE2.EW.0) GO TO 270
DO 265 J=1,JE2
P(ISETX,J,KW+1)=SUKW1(ISETX,J)
SLKW1(ISETX,J)=P(ISETX,J,KW-1)
KH=KHODU(ISETX,J)
P(ISETX,J,KH-1)=DU1(ISETX,J)
265 P(ISETX,J,KH-2)=DU2(ISETX,J)
270 CONTINUE
DO 275 J=1,JE1
P(ISET,J,KW+1)=SUKW1(ISET,J)
SLKW1(ISET,J)=P(ISET,J,KW-1)
KH=KRODU(ISET,J)
P(ISET,J,KH-1)=DU1(ISET,J)
275 P(ISET,J,KH-2)=DU2(ISET,J)
DO 280 J=1,JE0
P(ISETP,J,KW+1)=SUKW1(ISETP,J)
SLKW1(ISETP,J)=P(ISETP,J,KW-1)
KH=KRODU(ISETP,J)
P(ISETP,J,KH-1)=DU1(ISETP,J)
280 P(ISETP,J,KH-2)=DU2(ISETP,J)
DO 290 JSET=1,JE1
KH=KRODU(ISET,JSET)
CALL TRICUE(KH,KMX,ISET,JSET)
CALL INVERT(KH,KMX)
CALL MAXI(KH,KMX,ISET,JSET)
DO 285 K=KB,KMX
PT1(JSET,K)=P(ISET,JSET,K)
285 P(ISET,JSET,K)=PNEW(K)
PT1(JSET,KM)=P(ISET,JSET,KM)
KH1=KH-1
DO 287 K=KW,KH1

```



```

287 PT1(JSET,K)=P(ISET,JSET,K)
CALL SURFBC(ISET,JSET,2)
P(ISET,JSET,KW-1)=DU1(ISET,JSET)
P(ISET,JSET,KW-2)=DU2(ISET,JSET)
290 CONTINUE
DO 295 J=1,JE1
SU3(J)=SU2(J)
295 SU2(J)=PT1(J,KW)
IF (JE2.EQ.0) GO TO 310
DO 305 J=1,JE2
305 P(ISETX,J,KW-1)=SLKW1(ISETX,J)
310 CONTINUE
DO 315 J=1,JE1
315 P(ISET,J,KW-1)=SLKW1(ISET,J)
DO 320 J=1,JEU
320 P(ISETP,J,KW-1)=SLKW1(ISETP,J)
C
C     LINES OUTBOARD OF WING/BODY SIDE EDGE
IF (ISET.EQ.I LX) P(ILE,JROOT-1,KW)=PSAVE
IF (ISET.EQ.ITP) GO TO 3325
DO 325 J=1,JM
PT2(J,KW)=SS2(J)
325 PT3(J,KW)=SS3(J)
GO TO 326
3325 DO 3330 J=JROOT,JTIP
PT2(J,KW)=PTE1(J)
PT3(J,KW)=PTE2(J)
P(ITE,J,KW)=PTE1(J)
3330 P(ITE-1,J,KW)=PTE2(J)
P(ITE,JROOT-1,KW)=PTE1(JROOT-1)
326 IF (JE2.EQ.0) GO TO 327
IF (ISET.EQ.ITP) GO TO 3327
P(ISETX,JE2,KW)=DS1(ISETX)
IF (JE2-1.LT.1) GO TO 327
P(ISETX,JE2-1,KW)=DS2(ISETX)
3327 CONTINUE
327 PT1(JE1,KW)=DS1(ISET)
IF (JE1.LT.1) GO TO 330
PT1(JE1-1,KW)=DS2(ISET)
330 CONTINUE
IF (ISET.EQ.I LX) GO TO 332
P(ISETP,JEU,KW)=DS1(ISETP)
IF (JEU-1.LT.1) GO TO 332
P(ISETP,JEU-1,KW)=DS2(ISETP)
332 CONTINUE
JSET=JBOD(ISET)
CALL SURFSC(ISET,JSET,0)
P(ISET,JE1,KW)=US1(ISET)
IF (JE1-1.LT.1) GO TO 335
P(ISET,JE1-1,KW)=DS2(ISET)
335 JE=JE1+1
DO 370 JSET=JE,JMX
CALL TRICUE(2,KMX,ISET,JSET)
CALL INVERT(2,KMX)
CALL MAXI(2,KMX,ISET,JSET)
DO 350 K=2,KMX
PT1(JSET,K)=P(ISET,JSET,K)
350 P(ISET,JSET,K)=PNEW(K)
PT1(JSET,1)=P(ISET,JSET,1)
PT1(JSET,KM)=P(ISET,JSET,KM)
370 CONTINUE
DO 375 K=1,KM
375 PT1(JM,K)=P(ISET,JM,K)
DO 400 J=1,JM
DO 400 K=1,KM
PT3(J,K)=PT2(J,K)
400 PT2(J,K)=PT1(J,K)
DO 410 J=1,JM
SS3(J)=PT3(J,KW)
410 SS2(J)=PT2(J,KW)

```

```

      DO 420 J=JE,JM
      SL2(J)=SS2(J)
420  SL2(J)=SS2(J)
500  CONTINUE
      WRITE (6,511) ITER,DPMAX,IMAXI,JMAXI,KMAXI,NSUP
511  FORMAT (1H,110,E15.4,3I5,110)
C
C
      UPDATE CIRCULATION AND FAR-FIELD CONDITION
      DO 600 J=JROOT,JTIP
      GNEW=(2.-DG/DH)*(PWHU(ITE,J)-PWBL(ITE,J))-(1.-DG/DH)*
      1 (PWRU(ITE-1,J)-PWL(ITE-1,J))
600  GAMMA(J)=WG*GNEW+(1.-WG)*GAMMA(J)
      GAMMA(JROOT-1)=2.*GAMMA(JROOT)-GAMMA(JROOT+1)
      CALL FARBC
      WRITE (6,611) GAMMA(JROOT),ITERI,DPMAXT,JMAXT,KMAXT
611  FORMAT (1H,55X,E15.4,110,E13.4,2I5)
      RETURN
      END

SUBROUTINE FARBC
      UPDATES FAR-FIELD BOUNDARY CONDITION IN THE TREFFTZ PLANE.
C
C
      DIMENSION SAVE(19)
      COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
      * SNA,TDelta,TDXI,TUZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
      * ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
      * PLCB2,PLCB3
      COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
      * DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERI,
      * JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
      * PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
      * PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)
      COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZDXL(49,19),
      * DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
      * HWBU(49,19),IZL(19),IZU(19),JBOD(49),KBODL(49,19),
      * KBODU(49,19),ZL(49,19),ZU(49,19)
C
C
      ITERI=0
      JMX=JM-1
      KMX=KM-1
      JE=JROOT(IM)-1
100  DPMAXT=0.0
      DO 110 J=1,JE
      SAVE(J)=P(IM,J,KW+1)
      KH=KBODL(IM,J)
      P(IM,J,KH+1)=DL1(IM,J)
110  P(IM,J,KH+2)=DL2(IM,J)
      DO 150 J=1,JE
      KH=KBODL(IM,J)
      JSET=J
      CALL TRIT(2,KH,0,JSET)
      CALL INVERT(2,KH)
      DO 125 K=2,KH
      DELP=PNEW(K)-P(IM,J,K)
      DP=ARS(DELP)
      IF (DP.LT.DPMAXT) GO TO 125
      DPMAXT=DP
      JMAXT=J
      KMAXT=K
      PT1(J,K)=P(IM,J,K)

```

```

125 P(IM,J,K)=PNEW(K)
    PT1(J,1)=P(IM,J,1)
    KB1=KB+1
    DO 130 K=KB1,KW
130 PT1(J,K)=P(IM,J,K)
150 CONTINUE
C
    DO 210 J=1,JE
    P(IM,J,KW+1)=SAVE(J)
    SAVE(J)=P(IM,J,KW-1)
    KB=KBODU(IM,J)
    P(IM,J,KB-1)=DU1(IM,J)
210 P(IM,J,KB-2)=DU2(IM,J)
    DO 250 J=1,JE
    KB=KBODU(IM,J)
    JSET=J
    CALL TRIT(KB,KMX,0,JSET)
    CALL INVERT(KB,KMX)
    DO 225 K=KB,KMX
    DELP=PNEW(K)-P(IM,J,K)
    DP=ABS(DELP)
    IF (DP.LT.DPMAXT) GO TO 225
    DPMAXT=DP
    JMAXT=J
    KMAXT=K
225 PT1(J,K)=P(IM,J,K)
    P(IM,J,K)=PNEW(K)
    PT1(J,KM)=P(IM,J,KM)
    KB1=KB-1
    DO 230 K=KW,KB1
230 PT1(J,K)=P(IM,J,K)
250 CONTINUE
C
    DO 310 J=1,JE
310 P(IM,J,KW-1)=SAVE(J)
    P(IM,JE,KW)=DS1(IM)
    IF (JE.NE.1) P(IM,JE-1,KW)=DS2(IM)
    PT1(JE,KW)=DS1(IM)
    IF (JE.NE.1) PT1(JE-1,KW)=DS2(IM)
    DO 350 J=JROOT,JMX
    JSET=J
    CALL TRIT(2,KMX,0,JSET)
    CALL INVERT(2,KMX)
    DO 325 K=2,KMX
    DELP=PNEW(K)-P(IM,J,K)
    DP=ABS(DELP)
    IF (DP.LT.DPMAXT) GO TO 325
    DPMAXT=DP
    JMAXT=J
    KMAXT=K
325 PT1(J,K)=P(IM,J,K)
    P(IM,J,K)=PNEW(K)
    PT1(J,1)=P(IM,J,1)
    PT1(J,KM)=P(IM,J,KM)
350 CONTINUE
C
    ITER=ITER+1
    IF (ITER.GE.200) RETURN
    IF (DPMAXT.GT.DPLIM) GO TO 100
    RETURN
    END

```

C
C
C
C
SUBROUTINE SURFBC(I,J,M)

CALCULATES INTERIOR IMAGE-POINT POTENTIAL VALUES REQUIRED TO SATISFY THE FLOW-TANGENCY BOUNDARY CONDITION ON LOWER (M.EQ.1) AND UPPER (M.EQ.2) SURFACES OF THE WING/BODY.

C

```

COMMON/CAPS / CAPG(49,19),CAPGT1(19,25),CAPGT2(49,19),CAPH(19),
* CAPH(49,19),CAPHT1(19,25),CAPHT2(49,19),CAPIB(19),
* CAPGB(49,19),CAPIT(49,19)
COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPST,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
* SNA,TDETA,TDXI,TUZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
COMMON/COORD/ CHORD(19),DCDY(19),DTDY(19),DXLEDY(19),ETA(19),F(49)
* ,FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
* TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
* ZCAP(25),ZLE(19),UZLEDY(19),ALPHAT(19),DADY(19),GWT
COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
* DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERT,
* JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
* PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
* PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)
COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZDXL(49,19),
* DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
* HWBU(49,19),IZL(19),IZU(19),JBOD(49),KBODL(49,19),
* KBODU(49,19),ZL(49,19),ZU(49,19)

```

C

```

DK=DETA
TC=TAU(J)*CHORD(J)
IF (M.EQ.2) GO TO 100
KP=-1
UL=-DZETA
DEL=-DELL(I,J)
IZ=IZL(J)
SLOPX=DZDXL(I,J)
SLOPY=DZDYL(I,J)
KS=KRDDL(I,J)
HWB=HWBL(I,J)
CAPGTH=(DL-DEL)*(CAPGT1(J,KS)-CAPGT1(J,KS+1))/DL+CAPGT1(J,KS+1)
CAPGTB=CAPGTB+CAPGT2(I,J)
GO TO 120
100 KP=+1
DL=DZETA
DEL=DELU(I,J)
IZ=IZU(J)
SLOPX=DZDXU(I,J)
SLOPY=DZDYU(I,J)
KS=KBODU(I,J)
HWB=HWBU(I,J)
CAPGTH=(DL-DEL)*(CAPGT1(J,KS)-CAPGT1(J,KS-1))/DL+CAPGT1(J,KS-1)
CAPGTB=CAPGTB+CAPGT2(I,J)
120 IF (I.EQ.1M) GO TO 210
IF (J.GE.JROOT) GO TO 140
DH=-DXI
IP=-1
GO TO 200
140 IF (I.GE.IZ) GO TO 160
DH=-DXI
IP=-1
GO TO 200
160 DH=+DXI
IP=+1
C
200 CAPK=(SLOPX/CHORD(J)+CAPG(I,J)*SLOPY)*F(I)
CAPL=SLOPY*G(J)/HSPAN
CAPM=HWB*(CAPGB(I,J)*SLOPX+CAPGTH*SLOPY-1./TC)
CAPP=CSA*SLOPX-SNA
GO TO 215
210 CAPK=0.0
CAPL=SLOPY*G(J)/HSPAN
CAPM=HWB*(CAPGTH*SLOPY-1./TC)
CAPP=-SNA
GO TO 220

```

```

C
215 A1=-1.5/DH
    A2=2.0/DH
    A3=-0.5/DH
    GO TO 225
220 A1=0.0
    A2=0.0
    A3=0.0
225 B1=-1.5/DK
    B2=2.0/DK
    B3=-0.5/DK
    C1=(DL+2.*DEL)/(DEL*(DL+DEL))
    C2=(DL+DEL)/(DEL*DL)
    C3=-DEL/(DL*(DL+DEL))
    D1=0.5*(DL+DEL)*(2.*DL+DEL)/(DL*DL)
    D2=-DEL*(2.*DL+DEL)/(DL*DL)
    D3=0.5*DEL*(DL+DEL)/(DL*DL)
    E1=2.*DL*DL/(DEL*(DL+DEL))
    E2=-2.*(DL-DEL)/DEL
    E3=(DL-DEL)/(DL+DEL)
C
    J5=J+1
    J6=J+2
    KS1=KS+KP
    KS2=KS1+KP
    P1=P(I,J,KS)
    P2=P(I,J,KS1)
    P5=D1*P(I,J5,KS)+D2*P(I,J5,KS1)+D3*P(I,J5,KS2)
    P6=D1*P(I,J6,KS)+D2*P(I,J6,KS1)+D3*P(I,J6,KS2)
    IF (J.EQ.JB1-1.OR.J.EQ.JB1) GO TO 181
    IF (J.EQ.JB2-1.OR.J.EQ.JB2) GO TO 1182
    IF (J.EQ.JB3-1.OR.J.EQ.JB3) GO TO 182
    GO TO 1185
181 JK=JB1
    PLC=PLC1
    GO TO 183
1182 JK=JB2
    PLC=PLC2
    GO TO 183
182 JK=JB3
    PLC=PLC3
183 CONTINUE
    PA=(2.*P(I,JK+1,KS)+(1.-PLC)*(P(I,JK-1,KS)
1 -2.*P(I,JK,KS)))/(1.+PLC)
    PB=(2.*P(I,JK+1,KS1)+(1.-PLC)*(P(I,JK-1,KS1)
1 -2.*P(I,JK,KS1)))/(1.+PLC)
    PC=(2.*P(I,JK+1,KS2)+(1.-PLC)*(P(I,JK-1,KS2)
1 -2.*P(I,JK,KS2)))/(1.+PLC)
    IF (J.EQ.JK) GO TO 184
    P6=D1*PA+D2*PB+D3*PC
    GO TO 1185
184 P5=D1*PA+D2*PB+D3*PC
    PAA=P(I,JK-1,KS)+3.*(PA-P(I,JK,KS))
    PBB=P(I,JK-1,KS1)+3.*(PB-P(I,JK,KS1))
    PCC=P(I,JK-1,KS2)+3.*(PC-P(I,JK,KS2))
    P6=D1*PAA+D2*PBB+D3*PCC
1185 IF (I.NE.IM) GO TO 230
    P3=0.0
    P4=0.0
    GO TO 240
230 I3=I+IP
    I4=I3+IP
    P3=D1*P(I3,J,KS)+D2*P(I3,J,KS1)+D3*P(I3,J,KS2)
    P4=D1*P(I4,J,KS)+D2*P(I4,J,KS1)+D3*P(I4,J,KS2)
240 DENOM=CAPK*A1+CAPL*H1+CAPM*C1
    T1=-CAPK*(A2*P3+A3*P4)
    T2=-CAPL*(B2*P5+B3*P6)
    T3=-CAPM*(C2*P1+C3*P2)-CAPP
    PSUHF=(T1+T2+T3)/DENOM
    P01=E1*PSUHF+E2*P1+E3*P2
    P02=3.*P01-J.*P1+P2
    IF (M.EQ.2) GO TO 250

```

```

DL1(I,J)=PD1
UL2(I,J)=PD2
PWFL(I,J)=PSURF
RETURN
250 DU1(I,J)=PD1
DU2(I,J)=PD2
PWBU(I,J)=PSURF
RETURN

```

C
C
C
C

PLANFORM-SIDE-EDGE IMAGE-POINT POTENTIALS

```

ENTRY SURFSC
IMX=IM-1
DK=DETA
DEL=DELS(I)
IF (I.GT.ITE) GO TO 300
DH=-DXI
IP=-1
GO TO 310
300 DH=+DXI
IP=+1
310 FP=F(I)
IF (I.LT.ILE.OR.I.GT.ITE) GP=GF
IF (I.GE.ILE.AND.I.LE.ITE) GP=GWT
GSET=((DK-DEL)*CAPG(I,J)+DEL*CAPG(I,J-1))/DK
GTA=CAPGT1(J,KW)+CAPGT2(I,J)
GTR=CAPGT1(J-1,KW)+CAPGT2(I,J-1)
GTSET=((DK-DEL)*GTA+DEL*GTR)/DK

```

C

```

A1=1.5/DH
A2=-2.0/DH
A3=0.5/DH
B1=-((DK+2.*DEL)/(DEL*(DK+DEL)))
B2=(DK+DEL)/(DEL*DK)
B3=-DEL/(DK*(DK+DEL))
E1=2.*DK*UK/(DEL*(DK+DEL))
E2=-2.*(DK-DEL)/DEL
E3=(DK-DEL)/(DK+DEL)

```

C

```

I1=I+IP
I2=I1+IP
IF (I.EQ.IM) I1=I
IF (I.EQ.IMX.OR.I.EQ.IM) I2=I1
J1=J+1
P1=P(I,J,KW)
P2=P(I,J1,KW)
P3=((DK+DEL)*P(I1,J,KW)-DEL*P(I1,J1,KW))/DK
P4=((DK+DEL)*P(I2,J,KW)-DEL*P(I2,J1,KW))/DK
P5=((DK+DEL)*P(I,J,KW-1)-DEL*P(I,J1,KW-1))/DK
P6=((DK+DEL)*P(I,J,KW+1)-DEL*P(I,J1,KW+1))/DK
GG=((DK+DEL)*GAMMA(J)-DEL*GAMMA(J1))/DK
IF (I.GT.ITE) P5=P5+GG

```

C

```

SC1=FP*GSET
SC2=GP/RSPAN
SC3=H(KW)*GTSET
IF (I.EQ.IM) SC1=0.
DENOM=SC1*A1+SC2*B2
T1=-SC1*(A2*P3+A3*P4)
T2=-SC2*(B2*P1+B3*P2)
T3=-SC3*(P6-P5)/(2.*DZETA)
PSURF=(T1+T2+T3)/DENOM
DS1(I)=E1*PSURF+E2*P1+E3*P2
DS2(I)=3.*DS1(I)-3.*P1+P2
RETURN

```

C
C
C
C

WING/BODY SYMMETRY-PLANE POTENTIALS

C

```

ENTRY SYMBC
KWX=KW-1
KWP=KW+1
KMX=KM-1
DO 500 II=2,IM
GSET=CAPG(II,1)
FC1=GSET*F(II)/DXI
FC2=G(1)/(RSPAN*DETA)
KL=KHODL(II,1)
KU=KHODU(II,1)
IF (II.EQ.IM) GO TO 470
DO 410 K=2,KWA
IF (KL.NE.0.AND.K.GT.KL) GO TO 410
CAPGT=CAPGT1(1,K)+CAPGT2(II,1)
FC3=CAPGT*H(K)/DZETA
IF (GSET.LE.0.) PSYM(II,K)=(FC1*PSYM(II-1,K)-FC2*P(II,1,K)
1 +FC3*PSYM(II,K-1))/(FC1-FC2+FC3)
IF (GSET.GT.0.) PSYM(II,K)=(-FC1*PSYM(II+1,K)-FC2*P(II,1,K)
1 +FC3*PSYM(II,K-1))/(-FC1-FC2+FC3)
410 CONTINUE
DO 420 K=KWP,KMX
KK=KMX-K+KWP
IF (KU.NE.0.AND.KK.LT.KU) GO TO 420
CAPGT=CAPGT1(1,KK)+CAPGT2(II,1)
FC3=CAPGT*H(KK)/DZETA
IF (GSET.LE.0.) PSYM(II,KK)=(FC1*PSYM(II-1,KK)-FC2*P(II,1,KK)
1 -FC3*PSYM(II,KK+1))/(FC1-FC2-FC3)
IF (GSET.GT.0.) PSYM(II,KK)=(-FC1*PSYM(II+1,KK)-FC2*P(II,1,KK)
1 -FC3*PSYM(II,KK+1))/(-FC1-FC2-FC3)
420 CONTINUE
IF (II.GE.INOSE) GO TO 500
CAPGT=CAPGT1(1,KW)+CAPGT2(II,1)
FC3=0.5*CAPGT*H(KW)/DZETA
PSYM(II,KW)=(FC1*PSYM(II-1,KW)-FC2*P(II,1,KW)-FC3*(PSYM(II,KW+1)
1 -PSYM(II,KW-1)))/(FC1-FC2)
GO TO 500
470 DO 480 K=2,KL
CAPGT=CAPGT1(1,K)+CAPGT2(II,1)
FC3=CAPGT*H(K)/DZETA
480 PSYM(IM,K)=(-FC2*P(IM,1,K)+FC3*PSYM(IM,K-1))/(-FC2+FC3)
DO 490 K=KU,KMX
KK=KMX-K+KU
CAPGT=CAPGT1(1,KK)+CAPGT2(II,1)
FC3=CAPGT*H(KK)/DZETA
490 PSYM(IM,KK)=(-FC2*P(IM,1,KK)-FC3*PSYM(IM,KK+1))/(-FC2-FC3)
500 CONTINUE
RETURN
END

```

SUBROUTINE TRICOE(K1,K2,I,J)

CALCULATES FINITE-DIFFERENCE COEFFICIENTS AT GRID POINTS ALONG THE SEGMENT K1 TO K2 OF LINE (I,J).

CCCCC

```

COMMON/ABCD / ACAP(25),BCAP(25),CCAP(25),DCAP(25)
COMMON/CAPS / CAPG(49,19),CAPGT1(19,25),CAPGT2(49,19),CAP1(19),
* CAPH(49,19),CAPHT1(19,25),CAPHT2(49,19),CAP1B(19),
* CAPGB(49,19),CAP1I(49,19)
COMMON/COEF / A1(49),A2(49),A3(49),A5(49),B1(19),B2(19),B3(19),
* B5(19),B7(19),C1(25),C2(25),C3(25),C5(25),C7(25),
* D1,D2,E1,E2,F1,F2

```

```

* COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,A12,CSA,RX1,RX2,
* SNA,TDETA,TDXI,TDZETA,DETA,DXI,UZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROUT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
* COMMON/COURD/ CHORD(19),DCDY(19),DTDY(19),DXLEDY(19),ETA(19),F(49)
* ,FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
* TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
* ZCAP(25),ZLE(19),UZLEDY(19),ALPHAT(19),DADY(19),GWT
* COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DUI(49,19),
* DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITEXT,
* JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
* PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
* PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)

```

C
C
C
C

CHECK P.D.E. TYPE

```

FP=F(I)
GP=G(J)
CJ=CHORD(J)
TC=TAU(J)*CJ
CBG=CAPG(I,J)
CU1=FP/CJ
CU2=CAPGB(I,J)
CV1=CBG*FP
CV2=GP/BSPAN
CW1=1./TC
CAP1=FP/(CJ*CJ)
CAP2=FP*CBG*CBG
CAP3=-2.*FP*CBG/CJ
CBP1=GP/(BSPAN*BSPAN)
CCP1=CU2*CU2
CCP2=1./(TC*TC)
CCP3=-2.*CU2
CCP4=-2./TC
CCP5=-2.*CU2/TC
CDP1=2.*FP*GP*CBG/BSPAN
CDP2=-2.*FP*GP/(CJ*BSPAN)
CEP1=2.*GP/BSPAN
CEP2=-2.*GP*CU2/BSPAN
CEP3=-2.*GP/(TC*BSPAN)
CFP1=2.*FP*CU2/CJ
CFP2=2.*FP*CBG
CFP3=-2.*FP/CJ
CFP4=-2.*FP*CU2*CBG
CFP5=-2.*FP*CBG/TC
CFP6=-2.*FP/(TC*CJ)
CJP1=2.*CAP1(J)*CJ
CJP2=-2.*CAPH(I,J)*CJ
CJP3=2.*CAPIT(I,J)*TC
CJP4=-2.*CAPIB(J)*TC
CJP5=-TC
CJP6=-CU2*TC

```

C

```

JK=JM
JKP=JK+1
IF (J.EQ.JB1.OR.J.EQ.JB1+1) GO TO 10
IF (J.EQ.JB2.OR.J.EQ.JB2+1) GO TO 20
IF (J.EQ.JB3.OR.J.EQ.JB3+1) GO TO 30
GO TO 90
10 JK=JB1
   PLC=PLCB1
   GO TO 80
20 JK=JB2
   PLC=PLCB2
   GO TO 80
30 JK=JB3
   PLC=PLCB3

```



```

80 JKP=JK+1
90 CONTINUE
DO 700 K=K1,K2
HP=H(K)
GTP=CAPGT1(J,K)+CAPGT2(I,J)
HTP=CAPHT1(J,K)+CAPHT2(I,J)
P1=P(I+1,J,K)
P2=P(I,J,K)
P3=PT2(J,K)
P4=P(I,J,K-1)
P5=P(I,J,K+1)
IF (J.EQ.1) P11=PSYM(I,K)
IF (J.NE.1) P11=PT1(J-1,K)
P16=P(I,J+1,K)
IF (J.EQ.JK) P16=(2.*P16+(1.-PLC)*(P11-2.*P2))/(1.+PLC)
IF (J.EQ.JKP) P11=(PLC*P16-2.*(PLC*P2-P11))/(2.-PLC)
UP=CSA+CU1*(P1-P3)/TDXI+CU2*HP*(P8-P5)/TDZETA
VP=CV1*(P1-P3)/TDXI+CV2*(P16-P11)/TDZETA+GTP*HP*(P8-P5)/TDZETA
WP=SNA+CW1*HP*(P8-P5)/TDZETA
IF (I.LE.ITE) GO TO 105
IF (K.NE.KW.AND.K.NE.KW+1) GO TO 105
UP=UP-CU2*HP*GAMMA(J)/TDZETA
VP=VP-GTP*HP*GAMMA(J)/TDZETA
WP=WP-CW1*HP*GAMMA(J)/TDZETA
105 UPSQ=UP*UP
VPSQ=VP*VP
WPSQ=WP*WP
QPSQ=UPSQ+VPSQ+WPSQ
TQ=AI2+0.20-1.20*QPSQ
APSQ=TQ*QPSQ
IF (TQ.LT.0.) GO TO 200

```

C
C
C
C

DEFINE POINTS OF ELLIPTIC (SUBSONIC) COMPUTATION MOLECULE AND
COMPUTE TRI-DIAGONAL COEFFICIENTS

```

P3N=P(I-1,J,K)
P4=P(I+1,J,K-1)
IF (I.EQ.ITE.AND.K1.EQ.KW+1) P4=P4+GAMMA(J)
P6N=P(I-1,J,K-1)
P7=P(I+1,J,K+1)
P9N=P(I-1,J,K+1)
P15=P(I+1,J+1,K)
P17N=P(I-1,J+1,K)
P18=P(I,J+1,K-1)
P19=P(I,J+1,K+1)
IF (J.EQ.1) GO TO 110
P10=P(I+1,J-1,K)
P11N=P(I,J-1,K)
P12N=P(I-1,J-1,K)
P13N=P(I,J-1,K-1)
P14N=P(I,J-1,K+1)
IF (J.EQ.JK) GO TO 112
IF (J.EQ.JKP) GO TO 116
GO TO 120
110 P10=PSYM(I+1,K)
P11N=PSYM(I,K)
P12N=PSYM(I-1,K)
P13N=PSYM(I,K-1)
P14N=PSYM(I,K+1)
GO TO 120
112 P15=(2.*P15+(1.-PLC)*(P10-2.*P1))/(1.+PLC)
P17N=(2.*P17N+(1.-PLC)*(P12N-2.*P3N))/(1.+PLC)
P18=(2.*P18+(1.-PLC)*(P13N-2.*P5))/(1.+PLC)
P19=(2.*P19+(1.-PLC)*(P14N-2.*P8))/(1.+PLC)
GO TO 120
116 P10=(PLC*P15-2.*(PLC*P1-P10))/(2.-PLC)
P11N=(PLC*P16-2.*(PLC*P2-P11N))/(2.-PLC)
P12N=(PLC*P17N-2.*(PLC*P3N-P12N))/(2.-PLC)
P13N=(PLC*P18-2.*(PLC*P5-P13N))/(2.-PLC)
P14N=(PLC*P19-2.*(PLC*P8-P14N))/(2.-PLC)
C
120 TU=APSQ-UPSQ
TV=APSQ-VPSQ

```

```

TW=APSQ-WPSQ
CAP=CAP1*TV+CAP2*TV+CAP3*UP*VP
CBP=CBP1*TV
CCP=(CCP1*TV+GTP*(TV*GTP+CCP3*UP*VP+CCP4*VP*WP)+CCP2*TW
1 +CCP5*UP*WP)*HP
CDP=CDP1*TV+CJP2*UP*VP
CEP=(CEP1*TV*GTP+CEP2*UP*VP+CEP3*VP*WP)*HP
CFP=(CFP1*TV+CFP2*TV*GTP+UP*VP*(CFP3*GTP+CFP4)
1 +WP*(CFP5*VP+CFP6*UP))*HP
CJP=(CJP1*UP*VP+CJP2*TV)*(UP-CSA+CJP6*(WP-SNA))
1 *(CJP3*UP*VP+CJP4*VP*WP+CJP5*(V*HTP)*(WP-SNA)

```

C

```

ACAP(K)=CCP*C3(K)
BCAP(K)=CAP*A2(I)*RX1+CBP*B2(J)+CCP*C2(K)
CCAP(K)=CCP*C1(K)
DCAP(K)=CJP-CAP*(A1(I)*P1+A2(I)*RX2*P2+A3(I)*P3N)-CBP*
1 (B1(J)*P16+B3(J)*P11N)-CDP*(D1*(P15+P12N)+D2*(P10+P17N))-
2 CEP*(E1*(P19+P13N)+E2*(P18+P14N))-CFP*(F1*(P7+P6N)+F2*(P4+P9N))
IF (K.EQ.K1) DCAP(K)=DCAP(K)-ACAP(K)*P5
IF (K.EQ.K2) DCAP(K)=DCAP(K)-CCAP(K)*P8
IF (I.LE.ITE.OR.J.LT.JROOT) GO TO 700
IF (K.EQ.KW) UCAP(K)=DCAP(K)+CCP*C1(K)*GAMMA(J)-CEP*E2*
1 (GAMMA(J+1)-GAMMA(J-1))
IF (K.EQ.KW+1) DCAP(K)=DCAP(K)-CCP*C3(K)*GAMMA(J)-CEP*E2*
1 (GAMMA(J+1)-GAMMA(J-1))
GO TO 700

```

C
C
C

DEFINE POINTS OF HYPERBOLIC (SUPERSONIC) COMPUTATION MOLECULE AND COMPUTE TRI-DIAGONAL COEFFICIENTS

```

200 NSUP=NSUP+1
P3N=P(I-1,J,K)
P4=P(I+1,J,K-1)
IF (I.EQ.ITE.AND.K1.EQ.KW+1) P4=P4+GAMMA(J)
P6N=P(I-1,J,K-1)
P7=P(I+1,J,K+1)
P9N=P(I-1,J,K+1)
P15=P(I+1,J+1,K)
P17N=P(I-1,J+1,K)
P18=P(I,J+1,K-1)
P19=P(I,J+1,K+1)
IF (J.EQ.1) GO TO 270
P10=P(I+1,J-1,K)
P11N=P(I,J-1,K)
P12N=P(I-1,J-1,K)
P13N=P(I,J-1,K-1)
P14N=P(I,J-1,K+1)
IF (J.EQ.JK) GO TO 272
IF (J.EQ.JKP) GO TO 276
GO TO 280
270 P10=PSYM(I+1,K)
P11N=PSYM(I,K)
P12N=PSYM(I-1,K)
P13N=PSYM(I,K-1)
P14N=PSYM(I,K+1)
GO TO 280
272 P15=(2.*P15+(1.-PLC)*(P10-2.*P11))/(1.+PLC)
P17N=(2.*P17N+(1.-PLC)*(P12N-2.*P3N))/(1.+PLC)
P18=(2.*P18+(1.-PLC)*(P13N-2.*P5))/(1.+PLC)
P19=(2.*P19+(1.-PLC)*(P14N-2.*P8))/(1.+PLC)
GO TO 280
276 P10=(PLC*P15-2.*(PLC*P1-P10))/(2.-PLC)
P11N=(PLC*P16-2.*(PLC*P2-P11N))/(2.-PLC)
P12N=(PLC*P17N-2.*(PLC*P3N-P12N))/(2.-PLC)
P13N=(PLC*P18-2.*(PLC*P5-P13N))/(2.-PLC)
P14N=(PLC*P19-2.*(PLC*P8-P14N))/(2.-PLC)
280 CONTINUE
P20=PT3(J,K)
IF (K.EQ.K1) P21=P5
IF (K.NE.K1) P21=P(I,J,K-2)
IF (K.EQ.K2) P22=P8
IF (K.NE.K2) P22=P(I,J,K+2)

```

```

IF (J.EQ.1) P23=2.*P11-P2
IF (J.EQ.2) P23=PSYM(I,K)
IF (J.GE.3) P23=PT1(J-2,K)

```

C

```

CPP=CAP1*UPSQ+CAP2*VPSQ-CAP3*UP*VP
CQP=CBP1*VPSQ
CRP=(CCP1*UPSQ+GTP*(VPSQ*GTP-CCP3*UP*VP-CCP4*VP*WP)+CCP2*WPSQ
  -CCP5*UP*WP)*HP
1 CSP=CDP1*VPSQ-CDP2*UP*VP
CTP=(CEP1*VPSQ*GTP-CEP2*UP*VP-CEP3*VP*WP)*HP
CVP=(CFP1*UPSQ+CFP2*VPSQ*GTP-UP*VP*(CFP3*GTP+CFP4)
  -WP*(CFP5*VP+CFP6*UP))*HP
1 CWP=(CJP1*UP*VP-CJP2*VPSQ)*(UP-CSA+CJP6*(WP-SNA))
  +(CJP3*UP*VP+CJP4*VP*WP-CJP5*VPSQ*HTP)*(WP-SNA)
1 UCP2=QPSQ*(CAP1+CAP2)-CPP
UCQ2=QPSQ*CBP1-CQP
UCR2=QPSQ*(CCP1+GTP*GTP+CCP2)*HP-CRP
UCS2=QPSQ*CDP1-CSP
UCT2=QPSQ*CEP1*GTP*HP-CTP
UCV2=QPSQ*(CFP1+CFP2*GTP)*HP-CVP
UCW2=QPSQ*(-CJP2*(UP-CSA+CJP6*(WP-SNA))-CJP5*HTP*(WP-SNA))-CWP
FEQ=FP*UP/(CJ*DXI)
CLP=FEQ*(UP/CJ*VP*CBG)
CMP=FEQ*VP/BSPAN
CNP=FEQ*(UP*CU2+GTP*VP*WP*CW1)

```

C

```

TQ=1.-QPSQ/APSQ
ACAPT=UCR2*C3(K)
BCAPT=-UCP2*A3(I)-UCQ2*B3(J)-UCR2*(C1(K)+C3(K))
1 +TQ*2.*(CPP*A3(I)+CQP*B3(J)+2.*CSP*O1)
2 -EPSI*(CLP*A3(I)*DXI+CMF*B3(J)*DETA)
CCAPT=UCR2*C1(K)
DCAPT=-UCP2*(A1(I)*(P1-P2)+A3(I)*P3N)
1 -UCQ2*(B1(J)*(P16-P2)+B3(J)*P11N)
2 -UCS2*(D1*(P15+P12N)+D2*(P10+P17N))
3 -UCT2*(E1*(P19+P13N)+E2*(P18+P14N))
4 -UCV2*(F1*(P7+P6N)+F2*(P4+P9N))-UCW2
DCAPT=DCAPT-TQ*
1 (CPP*(-A3(I)*(P2+P3N)-A5(I)*(P3N-P20))
2 +CQP*(-B3(J)*(P2+P11N)-B5(J)*(P11N-P23))
3 +4.*CSP*(O1*P12N+D2*(P3N+P11N))+CWP)
DCAPT=DCAPT+EPSI*
1 (CLP*A3(I)*DXI*(-P2-P3N+P3)
2 +CMF*B3(J)*DETA*(-P2-P11N+P11))
IF (WP.EQ.0.) GO TO 320
IF (WP.LT.0.) GO TO 330

```

C

```

ACAP(K)=ACAPT+TQ*(-CRP*(C3(K)+C5(K))+4.*(CTP*E2+CVP*F2))
1 +EPSI*CNP*C3(K)*DZETA
BCAP(K)=BCAPT+TQ*2.*(CRP*C3(K)+2.*(CTP*E1+CVP*F1))
1 -EPSI*CNP*C3(K)*DZETA
CCAP(K)=CCAPT
DCAP(K)=DCAPT-TQ*(CRP*(-C3(K)*P2+C5(K)*P21)
1 +4.*CTP*(E1*P13N+E2*P11N)
2 +4.*CVP*(F1*P6N+F2*P9N))
3 +EPSI*CNP*C3(K)*DZETA*(-P2+P5)
GO TO 400

```

C

```

320 ACAP(K)=ACAPT+TQ*(CRP*C3(K)+2.*(CTP*E2+CVP*F2))
1 +EPSI*CNP*HP/TOZETA
BCAP(K)=BCAPT-TQ*CRP*(C1(K)+C3(K))
CCAP(K)=CCAPT+TQ*(CRP*C1(K)+2.*(CTP*E1+CVP*F1))
1 -EPSI*CNP*HP/TOZETA
DCAP(K)=DCAPT-TQ*(2.*CTP*(E1*P13N+E2*P14N)
1 +2.*CVP*(F1*P6N+F2*P9N))
2 +EPSI*HP*(-P8+P5)/TOZETA
GO TO 400

```

C

```

330 ACAP(K)=ACAPT
BCAP(K)=BCAPT+TQ*2.*(CRP*C1(K)+2.*(CTP*E2+CVP*F2))
1 +EPSI*CNP*C1(K)*DZETA

```

```

CCAP(K)=CCAPT+TQ*(-CRP*(C1(K)+C7(K))+4.*(CTP*E1+CVP*F1))
1 -EPSI*CNP*C1(K)*DZETA
DCAP(K)=DCAPT-TQ*(CRP*(-C1(K)*P2+C7(K)*P22)
2 +4.*CTP*(E2*P14N+E1*P11N)
3 +4.*CVP*(F2*P9N+F1*P3N))
+EPSI*CNP*C1(K)*DZETA*(P2-P8)

C
400 CONTINUE
IF (K.EQ.K1) DCAP(K)=DCAP(K)-ACAP(K)*P5
IF (K.EQ.K2) DCAP(K)=DCAP(K)-CCAP(K)*P8
700 CONTINUE
RETURN

C
C
C
C
COMPUTE TRI-DIAGONAL COEFFICIENTS IN THE TREFFTZ PLANE

ENTRY TRIT
GP=G(J)
TC=TAU(J)*CHORD(J)
CV2=GP/BSPAN
CW1=1./TC

C
DO 900 K=K1,K2
HP=H(K)
GTP=CAPGT1(J,K)+CAPGT2(IM,J)
HTP=CAPHT1(J,K)+CAPHT2(IM,J)
P5=P(IM,J,K-1)
P8=P(IM,J,K+1)
IF (J.EQ.1) P11=PSYM(IM,K)
IF (J.NE.1) P11=PT1(J-1,K)
P16=P(IM,J+1,K)
VP=CV2*(P16-P11)/TDETA+GTP*HP*(P8-P5)/TDZETA
WP=SNA+CW1*HP*(P8-P5)/TDZETA
IF (K.NE.KW.AND.K.NE.KW+1) GO TO 750
VP=VP-GTP*HP*GAMMA(J)/TDZETA
WP=WP-CW1*HP*GAMMA(J)/TDZETA
750 CONTINUE
TV=A12-VP*VP
CBT=TV*GP/(BSPAN*BSPAN)
CCT=HP*((A12-WP*WP)/(TC*TC)+GTP*(TV*GTP-2.*VP*WP/TC))
CET=2.*GP*HP*(TV*GTP-VP*WP/TC)/BSPAN
CJT=TC*(WP-SNA)*(2.*VP*WP*CAPIB(J)-TV*HTP)

C
P2=P(IM,J,K)
P18=P(IM,J+1,K-1)
P19=P(IM,J+1,K+1)
IF (J.EQ.1) GO TO 800
P11N=P(IM,J-1,K)
P13N=P(IM,J-1,K-1)
P14N=P(IM,J-1,K+1)
GO TO 850
800 P11N=PSYM(IM,K)
P13N=PSYM(IM,K-1)
P14N=PSYM(IM,K+1)

C
850 ACAP(K)=CCT*C3(K)
BCAP(K)=CBT*B2(J)*RX1+CCT*C2(K)
CCAP(K)=CCT*C1(K)
DCAP(K)=CJT-CBT*(B1(J)*P16+B2(J)*RX2*P2+B3(J)*P11N)
1 -CET*(E1*(P19+P13N)+E2*(P18+P14N))
IF (K.EQ.K1) DCAP(K)=DCAP(K)-ACAP(K)*P5
IF (K.EQ.K2) DCAP(K)=DCAP(K)-CCAP(K)*P8
IF (J.LT.JROOT) GO TO 900
IF (K.EQ.KW) DCAP(K)=DCAP(K)+CCT*C1(K)*GAMMA(J)
1 -CET*(E1*(GAMMA(J+1)-GAMMA(J-1)))
IF (K.EQ.KW+1) DCAP(K)=DCAP(K)-CCT*C3(K)*GAMMA(J)
1 -CET*(E2*(GAMMA(J+1)-GAMMA(J-1)))
900 CONTINUE
RETURN
END

```

```

SUBROUTINE INVERT(K1,K2)
INVERTS TRI-DIAGONAL MATRIX (VARGA, MATRIX ITERATIVE ANAL., P.195)

DIMENSION GG(25),WW(25)
COMMON/ABCO / ACAP(25),BCAP(25),CCAP(25),DCAP(25)
COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
*             DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERT,
*             JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
*             PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
*             PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)

CALCULATE COEFFICIENTS OF REDUCED SYSTEM OF EQUATIONS
WW(K1)=CCAP(K1)/RCAP(K1)
GG(K1)=DCAP(K1)/BCAP(K1)
K1P=K1+1
DO 100 K=K1P,K2
KZ=K-1
DENOM=BCAP(K)-ACAP(K)*WW(KZ)
WW(K)=CCAP(K)/DENOM
100 GG(K)=(DCAP(K)-ACAP(K)*GG(KZ))/DENOM

INVERT REDUCED MATRIX BY HACK SUBSTITUTION
PNEW(K2)=GG(K2)
KD=K2-K1
DO 200 K=1,KD
KZ=K2-K
200 PNEW(KZ)=GG(KZ)-WW(KZ)*PNEW(KZ+1)
RETURN
END

```

```

SUBROUTINE MAXI(K1,K2,I,J)
DETERMINES LOCATION AND ABSOLUTE VALUE OF THE MAXIMUM CHANGE IN
POTENTIAL BETWEEN THE LAST TWO COMPUTED ITERATES.

COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
*             DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERT,
*             JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
*             PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
*             PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)

DO 100 K=K1,K2
DELP=PNEW(K)-P(I,J,K)
DP=ABS(DELP)
IF (DP.LT.DPMAX) GO TO 100
DPMAX=DP
IMAXI=I
JMAXI=J
KMAXI=K
100 CONTINUE
RETURN
END

```

CCCCC

SUBROUTINE ARRAYS(NPRINT)

OUTPUTS SOLUTION ARRAYS EVERY NPRINT ITERATIONS. IF NPRINT.EQ.0,
ONLY THE CIRCULATION IS WRITTEN.

```

COMMON/CONST/ ALPHA,ZMACH,DPLIM,EPSI,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
* SNA,TDETA,TDXI,TUZETA,DETA,OXI,UZETA,ILE,IM,INOSE,
* ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
* PLCB2,PLCB3
COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
* DU2(49,19),OPMAX,OPMAXT,GAMMA(19),IMAXI,ITEXT,
* JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
* PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
* PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)

```

CCC

```

WRITE (6,161)
WRITE (6,171)
WRITE (6,171)
WRITE (6,181)
WRITE (6,161)
WRITE (6,131) (GAMMA(J),J=1,JM)
IF (NPRINT.EQ.0) GO TO 225
WRITE (6,161)

```

C

```

WRITE (6,141)
DO 150 J=1,JM
WRITE (6,142) J
DO 145 I=1,IM
145 WRITE (6,146) I,(P(I,J,K),K=1,KM)
150 CONTINUE
WRITE (6,161)

```

C

```

WRITE (6,101)
DO 110 J=1,JM
WRITE (6,102) J
DO 105 I=1,IM
105 WRITE (6,106) I,PWBL(I,J),DL1(I,J),DL2(I,J),DU2(I,J),DU1(I,J),
PWRU(I,J)
110 CONTINUE
WRITE (6,161)

```

C

```

WRITE (6,121)
DO 125 I=1,IM
125 WRITE (6,126) I,DS1(I),DS2(I)
WRITE (6,161)

```

C

```

WRITE (6,191)
DO 220 I=1,IM
220 WRITE (6,146) I,(PSYM(I,K),K=1,KM)
225 WRITE (6,161)
WRITE (6,171)
WRITE (6,171)
WRITE (6,161)
RETURN

```

C

```

101 FORMAT (1H-///1X40HDUMMY-POINT POTENTIALS ON VERTICAL LINES/
1 1X25H(PLANE-HY-PLANE SPANWISE))
102 FORMAT (///1X10HPLANE J = I3//3X1HI,16X4HPWBL,12X3HDL1,12X3HDL2,
1 17X3HUU2,12X3HUU1,11X4HPWBU/)
106 FORMAT (I4,2(5X,3E15.4))
121 FORMAT (1H-///1X38HDUMMY-POINT POTENTIALS ALONG SIDE EDGE//
1 3X1HI,17X3HDS1,12X3HDS2/)
126 FORMAT (I4,5X,2E15.4)
131 FORMAT (1H-///1X34HCIRCULATION DISTRIBUTION, GAMMA(J)//
1 (4X,13E10.3)///)
141 FORMAT (1H-///1X26HP(I,J,K): POTENTIAL ARRAY/
1 1X43H(PLANE-RY-PLANE SPAN-ISE, I DOWN, K ACROSS))

```

```

142 FORMAT (///1X10HPLANE J = I3/)
146 FORMAT (I4/(4X,1)3E10.3))
161 FORMAT (//////)
171 FORMAT (50H *****,
1      50H *****,
2      36H *****)
181 FORMAT (1H-///1X15HSOLUTION ARRAYS)
191 FORMAT (1H-///1X46HPSYM(I,K): IMAGE POTENTIALS AT SYMMETRY PLANE/
1      1X18H(I DOWN, K ACROSS)/)

```

C

END

C
C
C
C

SUBROUTINE CPCOMP
COMPUTES THE WING/BODY SURFACE PRESSURE DISTRIBUTION.

C

DIMENSION CPL(49,19),CPU(49,19)
DIMENSION ZXL(49,19),ZXU(49,19)

C
C
C

```

COMMON/CAPS / CAPG(49,19),CAPGT1(19,25),CAPGT2(49,19),CAPI(19),
CAPH(49,19),CAPHT1(19,25),CAPHT2(49,19),CAPIB(19),
CAPGB(49,19),CAPIT(49,19)
COMMON/CONST/ ALPHA,ZMACH,DPLIM,E,S1,WE,WG,BSPAN,AI2,CSA,RX1,RX2,
SNA,TOETA,TOXI,TOZETA,DETA,DXI,DZETA,ILE,IM,INOSE,
ITAIL,ITE,JM,JROOT,JTIP,KM,KW,JB1,JB2,JB3,PLCB1,
PLCB2,PLCB3
COMMON/COORD/ CHORD(19),DCDY(19),DTDY(19),DXLEDY(19),ETA(19),F(49),
FLE,FN,FTE,G(19),GAMLE,GAMN,GAMF,GAMTE,GAMWT,H(25),
TAU(19),X(49,19),XCAP(49),XLE(19),Y(19),ZETA(25),GF,
ZCAP(25),ZLE(19),DZLEDY(19),ALPHAT(19),DADY(19),GWT
COMMON/SOLVO/ DL1(49,19),DL2(49,19),DS1(49),DS2(49),DU1(49,19),
DU2(49,19),DPMAX,DPMAXT,GAMMA(19),IMAXI,ITERT,
JMAXI,JMAXT,KMAXI,KMAXT,NSUP,P(49,19,25),PLE1(19),
PNEW(25),PNOSE,PSYM(49,25),PT1(19,25),PT2(19,25),
PT3(19,25),PTE1(19),PTE2(19),PWBL(49,19),PWBU(49,19)
COMMON/SURF / DELL(49,19),DELS(49),DELU(49,19),DZDXL(49,19),
DZDXU(49,19),DZDYL(49,19),DZDYU(49,19),HWBL(49,19),
HWBU(49,19),IZL(19),IZU(19),JBOD(49),KBODL(49,19),
KBODU(49,19),ZL(49,19),ZU(49,19)

```

C

```

CPSTAR=AI2*((1.+0.20/AI2)/1.20)**3.5-1.)/0.70
DO 100 I=1,IM
DO 100 J=1,JM
ZXU(I,J)=0.
ZXL(I,J)=0.
CPU(I,J)=0.0
100 CPL(I,J)=0.0

```

110

```

DK=DETA
IMX=IM-1
DO 200 MSURF=1,2
DO 200 I=1,INOSE,IMX
JE=JROOT(I)-1
DO 200 J=1,JE
IF (MSURF.EQ.2) GO TO 110
KP=-1
DL=-DZETA
DEL=-DELL(I,J)
IZ=IZL(J)
KS=KBODL(I,J)
HWB=HWBL(I,J)
CAPGTB=(DL-DEL)*(CAPGT1(J,KS)-CAPGT1(J,KS+1))/DL+CAPGT1(J,KS+1)
CAPGTR=CAPGTR+CAPGT2(I,J)
GO TO 120
110 KP=+1

```

```

DL=07ETA
DEL=DELU(I,J)
IZ=IZU(J)
KS=KRODU(I,J)
HXB=HWBU(I,J)
CAPGTB=(DL-DEL)*(CAPGT1(J,KS)-CAPGT1(J,KS-1))/DL+CAPGT1(J,KS-1)
CAPGTB=CAPGTB+CAPGT2(I,J)
120 IF (J.GE.JROOT) GO TO 140
DH=-DXI
IP=-1
GO TO 180
140 IF (I.GE.IZ) GO TO 160
DH=-DXI
IP=-1
GO TO 180
160 DH=+DXI
IP=+1
C
180 A1=-1.5/DH
A2=2.0/DH
A3=-0.5/DH
B1=-1.5/DK
B2=2.0/DK
B3=-0.5/DK
C1=- (DL+2.*DEL)/(DEL*(DL+DEL))
C2=(DL+DEL)/(DEL*DL)
C3=-DEL/(DL*(DL+DEL))
D1=0.5*(DL+DEL)*(2.*DL+DEL)/(DL*DL)
D2=-DEL*(2.*DL+DEL)/(DL*DL)
D3=0.5*DEL*(DL+DEL)/(DL*DL)
C
FP=F(I)
GP=G(J)
CJ=CHORD(J)
CRG=CAPG(I,J)
CUI=FP/CJ
CUZ=CAPGH(I,J)
CVI=CRG*FP
CVZ=GP/BSPAN
CW1=1./(TAU(J)*CJ)
C
I3=I+IP
I4=I3+IP
J5=J+1
J6=J+2
KS1=KS+KP
KS2=KS1+KP
IF (MSURF.EQ.1) PZ=PWBH(I,J)
IF (MSURF.EQ.2) PZ=PWBU(I,J)
P1=P(I,J,KS)
P2=P(I,J,KS1)
P3=D1*P(I3,J,KS)+D2*P(I3,J,KS1)+D3*P(I3,J,KS2)
P4=D1*P(I4,J,KS)+D2*P(I4,J,KS1)+D3*P(I4,J,KS2)
P5=D1*P(I,J5,KS)+D2*P(I,J5,KS1)+D3*P(I,J5,KS2)
P6=D1*P(I,J6,KS)+D2*P(I,J6,KS1)+D3*P(I,J6,KS2)
IF (J.EQ.JB1-1.0R,J.EQ.JB1) GO TO 181
IF (J.EQ.JB2-1.0R,J.EQ.JB2) GO TO 1182
IF (J.EQ.JB3-1.0R,J.EQ.JB3) GO TO 182
GO TO 1185
181 JK=JB1
PLC=PLC1
GO TO 183
1182 JK=JB2
PLC=PLC2
GO TO 183
182 JK=JB3
PLC=PLC3
183 CONTINUE
PA=(2.*P(I,JK+1,KS)+(1.-PLC)*(P(I,JK-1,KS)
  1 -2.*P(I,JK,KS)))/(1.+PLC)
PH=(2.*P(I,JK+1,KS1)+(1.-PLC)*(P(I,JK-1,KS1)

```


AD-A101 944

MCDONNELL DOUGLAS RESEARCH LABS ST LOUIS MO
THREE-DIMENSIONAL TRANSONIC FLOW ANALYSIS. (U)
JUN 80 G E CHMIELENSKI, F W SPAID

F/6 20/4

UNCLASSIFIED

MDC-00722

F44620-76-C-0096

AFOSR-TR-81-0572

NL

512
AT
01/1/81



END
DATE
FILMED
8 81
DTIC

```

1 -2.*P(I,JK,KS1)))/(1.+PLC)
PC=(2.*P(I,JK+1,KS2)+(1.-PLC)*(P(I,JK-1,KS2)
1 -2.*P(I,JK,KS2)))/(1.+PLC)
IF (J.EQ.JK) GO TO 184
P6=U1*PA+U2*PB+U3*PC
GO TO 1185
184 P5=U1*PA+U2*PB+U3*PC
PAA=P(I,JK-1,KS)+3.*(PA-P(I,JK,KS))
PBB=P(I,JK-1,KS1)+3.*(PB-P(I,JK,KS1))
PCC=P(I,JK-1,KS2)+3.*(PC-P(I,JK,KS2))
P6=U1*PAA+U2*PBB+U3*PCC
1185 DPDS=A1*PZ+A2*P3+A3*P4
DPOZ=C1*PZ+C2*P1+C3*P2
U=CSA+CU1*DPDS+CU2*HWH*DPOZ
V=CV1*DPDS+CV2*(R1*PZ+R2*P5+R3*P6)+CAPGTB*HWH*DPOZ
W=SNA+CW1*HWH*DPOZ
IF (J.EQ.1) V=0.0
QX=1.-U*U-V*V-W*W
CALC=A12*((1.+0.20*QX/A12)**3.5-1.)/0.70
IF (MSURF.EQ.2) GO TO 185
ZXL(I,J)=(W-V*DZDYU(I,J))/U
CPL(I,J)=CALC
GO TO 200
185 ZXU(I,J)=(W-V*DZDYU(I,J))/U
CPU(I,J)=CALC
200 CONTINUE
C
JROOTX=JROOT-1
WRITE (6,301) CPSTAR
DO 310 J=1,JROOTX
YY=2.*Y(J)/BSPAN
WRITE (6,302) J,YY
DO 310 I=INOSI,IMX
TEMP=XCAP(I)+0.5
DZDX1=DZDXU(I,J)
DZDX2=DZDXL(I,J)
WRITE (6,303) TEMP,CPU(I,J),ZXU(I,J),DZDX1,CPL(I,J),
1 ZXL(I,J),DZDX2
310 CONTINUE
DO 320 J=JROOT,JTIP
YY=2.*Y(J)/BSPAN
WRITE (6,302) J,YY
DO 320 I=ILE,ITE
TEMP=XCAP(I)+0.5
DZDX1=DZDXU(I,J)
DZDX2=DZDXL(I,J)
WRITE (6,303) TEMP,CPU(I,J),ZXU(I,J),DZDX1,CPL(I,J),
1 ZXL(I,J),DZDX2
320 CONTINUE
RETURN
C
301 FORMAT (1H-////////1X30HSURFACE PRESSURE DISTRIBUTIONS///
1 1X8HCPSTAR = E13.5///)
302 FORMAT (1H-,17HSPAN STATION J = I4,6X6H2Y/8 = F9.4//
1 3X9H(X-XLE)/C,12X3HCPU,8X7HSLOPE U,10X5HDZDXU,12X3HCPL,
2 8X7HSLOPE L,10X5HDZDXL/)
303 FORMAT (F12.4,2(F15.4,2F15.6))
C
END

```