NASA Contractor Report 145383

SLENDER BODY THEORY PROGRAMMED FOR BODIES WITH ARBITRARY CROSS SECTION

J. Werner and A. R. Krenkel

POLYTECHNIC INSTITUTE OF NEW YORK
Farmingdale, NY

(NASA-CR-145383) SLENDER BODY THEORY PROGRAMMED FOR BODIES WITH ARBITRARY CROSS SECTION (Polytechnic Inst. of New York.) 77 p HC A05/MF A01 CSCL 01A

N78-26102

Unclas
G3/02 23366

NASA NSG-1300
February 1978

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SLENDER BODY THEORY PROGRAMMED FOR BODIES WITH ARBITRARY CROSS SECTION

by

J. Werner and A. R. Krenkel

This research was sponsored by the National Aeronautics and Space Administration, Langley Research Center, Hampton, Va., under Grant NSG-1300

POLYTECHNIC INSTITUTE OF NEW YORK

AERODYNAMICS LABORATORIES

February 1978

POLY-M/AE Report No. 77-17

SLENDER BODY THEORY PROGRAMMED FOR BODIES WITH ARBITRARY CROSS SECTION†

by

J. Werner* and A. R. Krenkel*

Polytechnic Institute of New York
Aerodynamics Laboratories
Farmingdale, New York

ABSTRACT

A computer program has been developed for determining the subsonic pressure, force and moment coefficients for a fuselage-type body using slender body theory. The program is suitable for determining the angle of attack and sideslipping characteristics of such bodies in the linear range where viscous effects are not predominant. Procedures have been developed which are capable of treating cross sections with corners or regions of large curvature.

†This research was sponsored by the National Aeronautics and Space Administration, Langley Research Center, Hampton, Va., under Grant NSG-1300.

*Associate Professors, Dept. of Mechanical and Aerospace Engineering

TABLE OF CONTENTS

Table of Contents continued | Page

SUMMARY

A computer program has been developed to determine the subsonic pressure, force and moment coefficients based on slender body theory for bodies of arbitrary cross section. The program is based on the integral representation of the potential in which the flow in the crossflow plane is considered to be induced sources distributed about the cross sectional boundary. Analytical expressions are derived for φ and its derivatives and the integrals appearing in these are evaluated by dividing each cross sectional boundary into straight line segments approximating the integrands over these segments. Results for pressure force and moment coefficients have been obtained for circular cone and ogive bodies and compared with analytical determinations from slender body theory. Results are also obtained for a typical "slab-sided" fuselage.

In Part III modifications have been developed which extend the applicability of the program in Pa t II to crossections with corners or local regions of high curvature.

INTRODUCTION

Computerization of aerodynamic theory has progressed to a point where the flow field analysis of complete aircraft configurations by a single program is now an attainable goal. Programs designed for this purpose do in fact exist, but predictably they are extremely large and abound with subtleties often not evident to the user. Generally, each new application undergoes a "debugging" stage which may in itself constitute a major effort. Much of the complexity of these programs is attributable to the level of precision of the underlying theory. Although often impressive, this precision is not always required. In some stages of preliminary design, for instance, it would be more desirable to sacrifice precision for simplicity. One approach in this spirit is to replace the commonly employed exact superposition method which panels the entire aircraft surface, placing appropriate singularities at each panel, with linearized theories involving only solutions of a local two-dimensional potential equation. In the exact theories a determination of the singularity strengths required to satisfy boundary conditions leads to the necessity of inverting very large matrices. The quasi-two-dimensional nature of linearized theories on the other hand considerably reduces the size of the matrices encountered and consequently places far less demand upon computer capabilities.

It is the purpose here to develop programs based on slender body theory, utilizing two-dimensional singularities distributed along a cross sectional contour to solve for the required potential function in the cross flow plane. Such an approach is felt to be particularly adaptable to the formulation of the interaction problems encountered in the analysis of complete aircraft configurations.

SYMBOLS

A_1	Coefficient of doublet term in expansion of complex Potential W.
$C(x)$	Cross sectional boundary at station x .
C_L, C_Y	Lift and side force coefficients.
C_M, C_N	Pitch and Yaw moment coefficients about nose of body.
C_n	Cross sectional boundary at $x = x_n$.
C_p	Pressure coefficient $(p - p_o)/(\rho U^2/2)$
F_y, F_z	Horizontal and Vertical Force components for body of unit length.
$g(x)$	Function of x derived from outer solution to potential equation.
h	Radius of curvature of cross sectional boundary.
i	Index of points along cross sectional boundary C.
i, i+1	Segment of C from i to i+1.
iL	Total number of segments into which C is divided.
$l(i, n)$	Length of segment i, i+1 on C_n.
M	Mach number
M_y, M_z	Components of moments about nose for body of unit length.
$n(i, n)$	Inner unit normal to segment i, i+1.
N	Total number of stations x_n

p	Pressure
p_o	Free Stream Pressure
q^2	$v^2 + w^2$
$R(i, j, n)$	Displacement from pt $P_{i,n}$ to pt $P'_{j,n}$ on C_n.
$\overline{R}(i, j, n)$	Vector displacement from $P_{i,n}$ to $P'_{j,n}$.
s	Distance along C_n.
S	Cross sectional area.
$\overline{u}(i, n)$	Unit tangent to segment $i, i+1$.
U	Free Stream Velocity.
r	Normalized Radial Polar Coordinate.
v	Normalized y component of velocity in wind azes.
w	Normalized z component of velocity in wind axes.
W	Normalized Complex Potential Function.
x	Normalized Longitudinal Function.
y, z	Wind axes coordinates in transverse plane.
Z	$y + iz$
Z_g	Complex location of cross sectional centroid.
α	Angle of attack.

β	$\sqrt{1 - M^2}$
$\delta(\ )$	Differential corresponding to displacement normal to C_n.
$\delta(i, j, n)$	Angle subtended by $i, i+1$ at pt. j at station x_n. (see Fig. 3)
ϵ	Angular Polar coordinate.
σ	2 Dimensional source density.
κ	Value of θ at point $P_{i,n}$.
θ	Angle between tangent to C and y axis
η	Normal displacement from mid point of $i, i+1$ on C_n to $i, i+1$ on C_{n+1}.
φ	Perturbation potential.
φ_o	$\varphi + g(x)$
Φ	$U\varphi_o$
$\delta\nu_o/\delta x$	Body slope in body axis frame of reference.
$\delta\nu/\delta x$	Body slope in wind axis frame of reference.
ζ	Complex position on C in wind axes.
ζ_o	Complex position on C in body axes.
Ψ	Yaw angle.

PART I

THEORY AND DEVELOPMENT OF NUMERICAL PROCEDURES

A. Synopsis of Subsonic Slender Body Theory

According to slender body theory (ref. 1), the flow disturbance near a sufficiently regular 3-D body may be represented by a potential in the form:

$$\Phi(xyz) = U\varphi_o = U[\varphi(xyz) + g(x)] \tag{1}$$

$\varphi(xyz)$ is a solution of the 2-D Laplace equation in the y, z cross flow plane satisfying the following boundary conditions appropriate to wind axes*

$$\nabla\varphi = 0 \text{ at } \infty \tag{2a}$$

$$\frac{\partial\varphi}{\partial n} = -\frac{\partial\nu}{\partial x} \text{ on } C(x) \tag{2b}$$

C(x), n, and ν being defined in Fig. (1). A general solution for φ may be written as the real part of a complex potential function W(Z) with Z = y + iz.

$$\varphi = \text{Re}W = \text{Re}\left[A_o(x)\ln Z + \sum_n^\infty A_n(x)/Z^n\right] \tag{3}$$

A useful alternative representation of φ and W is obtainable with the aid of Green's theorem. (ref. 2)

$$\varphi = \text{Re}W = -2\text{Re}\oint_{c(x)} \sigma(\zeta)\ln(Z-\zeta)ds \tag{4}$$

where $\sigma(\zeta)$ is a "source" density for values of $\zeta = y_c + iz_c$, (y_c, z_c) being coordinates of a point on the contour c(x).

---

* Although wind axes have been adopted as a reference, the computations have been formulated in terms of input data obtained from a body axes frame of reference. This avoids the necessity of generating new input data for each change in body attitude.

The function $g(x)$ is obtained by matching Φ of Eq. (1) which is valid in the neighborhood of the body with an appropriate "outer" solution. $g(x)$ is then found to depend explicitly on the longitudinal variation of cross sectional areas $S(x)$, i.e.:

$$g(x) = \frac{1}{2\pi}\left[S'(x)\ln(\beta/2) - \frac{1}{2}\int_0^x S''(t)\ln(x-t)dt + \frac{1}{2}\int_x^1 S''(t)\ln(t-x)dt - \frac{S'(0)}{2}\ln x - \frac{S(1)}{2}\ln(1-x)\right] \tag{5}$$

$\beta = \sqrt{1-M^2}$

The pressure coefficient, to an approximation consistent with slender body theory is given by the expression:

$$C_p = \frac{p-p_o}{\rho U^2/2} = -2\frac{\partial\varphi_o}{\partial x} - \left(\frac{\partial\varphi}{\partial y}\right)^2 - \left(\frac{\partial\varphi}{\partial z}\right)^2 \tag{6}$$

The force and moment about the origin on the portion of the body between the nose and station x are represented by the coefficients:

$$\frac{F_y + iF_z}{\rho U^2} = 2\pi A_1(x) + \frac{d}{dx}\Big(S(x)\ Z_g(x)\Big) \tag{7}$$

$$\frac{M_y + iM_z}{\rho U^2} = i\left\{x\cdot\frac{F_y + iF_z}{\rho U^2} - 2\pi\int_0^x A_1(t)dt - S(x)Z_g(x)\right\} \tag{8}$$

where $Z_g(x) = y_g + iz_g$ represents the complex location of the cross sectional centroid at station x, and $A_1(x)$ is the coefficient of the $1/Z$ term of Eq. (3). In terms of these force and moment expressions the more commonly used aerodynamic coefficients are written:

$$C_L = 2\left(\frac{F_z}{\rho U^2}\right)\frac{L^2}{S_{ref}}$$

$$C_y = 2\left(\frac{F_y}{\rho U^2}\right)\frac{L^2}{S_{ref}}$$

$$C_M = 2\left(\frac{M_y}{\rho U^2}\right)\frac{L^3}{L_{ref}\,S_{ref}}$$

$$C_N = -\,2\left(\frac{M_z}{\rho U^2}\right)\frac{L^3}{L_{ref}\,S_{ref}}$$

where L = body length and L_{ref}, S_{ref} are convenient reference length and area respectively, usually, determined by the overall configuration to be analyzed. For this report L_{ref} has been chosen to be equal to L and $S_{ref} = L^2$.

The reduction of computations of these expressions to a numerical procedure shall be based on the integral representation of ϕ given in Eq. (4). The point of departure shall be the discretization of the cross sectional boundary into a large number of short linear segments over each of which the source density σ shall be assumed constant at a value to be determined by boundary conditions.

B. Summary of Equations, Computational Procedures and Sample Calculations

Derivations of the equations presented in this section are given in Appendix A.

Since analytical results for bodies of revolution are readily available computations have been carried out for the purpose of comparison in the cases of a circular cone;

$$r(x) = x \tan 10^{\circ} \qquad 0 < x < 1$$

and an "ogive" of circular cross section;

$$r(x) = x(1 - \frac{x}{2}) \tan 10^{\circ} \quad 0 < x < 1$$

both at angle of attack $\alpha = .1$ and at zero Mach no.

1. Processing of Surface Data

The original data consists of the cross sectional boundaries C_n at each x_n presented in body axes coordinates as shown in Fig. 2. Starting at a convenient station x_n curves S_i are constructed orthogonal to the C_n. The intersections of these curves with C_n define a set of points $P_{i,n}$. The boundary C_n may now be approximated by the straight line segments $\overline{i, i+1}$ between the points $P_{i,n}$ and $P_{i+1,n}$. The coordinates $(y_{i,n}, z_{i,n})$ of the points $P_{i,n}$ together with the corresponding x_n represent the basic input data which defines the surface geometry in the program. Denoting the number of segments in a cross section by iL and the number of stations x_n by N the computations of this report have been carried out for N=10 and iL=20.

From the points $P_{i,n}$ a set of intermediate points $P'_{i,n}$ between $P_{i,n}$ and $P_{i+1,n}$ on C_n are derived. It is assumed that the coordinates of $P'_{i,n}$ may be represented by a Taylor's series in terms of the distance from $P_{i,n}$, i.e.;

$$y'_{i,n} = y_{i,n} + \left(\frac{dy}{ds}\right)_i \Delta s + \left(\frac{d^2y}{ds^2}\right)_i \frac{(\Delta s)^2}{2} + \ldots$$

$$\Delta s = \overline{P'_{i,n} - P_{i,n}} \quad .$$

Reduction of this expression to one in terms of the discrete points $P_{i,n}$ results in the following form (with a corresponding form for $z'_{i,n}$)

$$y'_{i,n} = y_{i,n} + Dy_i \frac{l(i,n)}{2} + \frac{DDy_i}{2}\left(\frac{l(i,n)}{2}\right)^2 + \ldots$$

where Dy_i is obtained by first computing the divided difference $(y_{i+i,n} - y_{i,n})/l(i,n)$ and taking this to represent dy/ds at the intermediate point $P'_{i,n}$ a distance $l(i,n)/2$ from $P_{i,n}$. Linear interpolation of dy/ds between $P'_{i-1,n}$ and $P'_{i,n}$ yields approximately dy/ds at $P_{i,n}$ and this is denoted by Dy_i. DDy_i is the approximation to $(d^2y/ds^2)_i$ determined by operating on Dy_i in the same manner. In this report terms up to second order in $l(i,n)$ have been employed. (Results obtained with $P'_{i,n}$ defined as above have been compared with those for $P'_{i,n}$ defined simply as the mid point of the secant $i,i+1$ and it was found that the latter case required double the number iL of segments to obtain comparable accuracy).

2. Source density σ

σ is determined by requiring that φ of Eq. (4) satisfy the boundary condition Eq. (2b) at a point P' of each segment $i,i+1$ of the boundary. The result of this process is a set of simultaneous equations for the densities $\sigma(i,n)$ at each segment $i,i+1$ of C_n. There densities may be assigned to the pts. $P'_{i,n}$, located as prescribed in Sect. 1.

$$\left(\frac{\partial v}{\partial x}\right)_{j,n} = \sum_{i=1}^{iL} A(j,i)\ \sigma(i,n) \tag{9}$$

where, referring to Fig. 3 for $R(i, j, n)$ and $\delta(i, j, n)$

$$A(j, i) = 2\Big\{\sin\big[\theta(j, n) - \theta(i, n)\big]\ln\big[R(i+1, j, n)/R(i, j, n)\big]$$

$$+ \delta(i, j, n)\cos\big[\theta(j, n) - \theta(i, n)\big]\Big\} \quad .$$

The slope $(\partial\nu/\partial x)_{j,n}$ may be written in terms of $(\partial\nu_o/\partial x)_{j,n}$ referred to box body axes and the angles of attack α and sideslip β (see Fig. (4))

$$\left(\frac{\partial\nu}{\partial x}\right)_{j,n} = \left(\frac{\partial\nu_o}{\partial x}\right)_{j,n} + \alpha \cos \theta(j, n) - \Psi \sin \theta(j, n) \quad . \tag{10}$$

Computation of $(\partial\nu_o/\partial x)_{j,n}$ from the surface data is described in Appendix A.

Values of $\sigma(i, 10)$ obtained from Eq. (9) in the case of a circular cone at angle of attack are presented in Fig. (5). The analytical solution for σ in the case of bodies of revolution is:

$$\sigma = -\frac{1}{2\pi}\left[\frac{S'(x)/2\pi r}{2} + \alpha \cos \theta\right] \quad .$$

This result is also presented in Fig. (5) for comparison.

3. Potential φ

Once the source density $\sigma(i, n)$ is determined Eq. (4) yields an explicit representation of φ. Integrating over the segments i, i+1 of C_n:

$$\varphi(j, n) = 2 \sum_{i=1}^{iL} \sigma(i, n)\{\bar{R}(i+1, j, n)\cdot \bar{u}(i, n)\ln R(i+1, j, n)$$

$$- \bar{R}(i, j, n)\cdot\bar{u}(i, n)\ln R(i, j, n) - \bar{R}(i, j, n)\cdot \bar{n}(i, n)\delta(i, j, n) + l(i, n)\}$$

$$= 2 \sum_{i=1}^{iL} \sigma(i, n) \left\{\frac{\Delta\varphi(i, j, n)}{\sigma(i, n)}\right\} \tag{11}$$

Although $\varphi(j, n)$ is not of direct interest the auxiliary functions $\Delta\varphi(i, j, n)/\sigma(i, n)$ appear in the results for $\partial\varphi/\partial x$ and so must be computed.

4. <u>Axial Potential Derivative $\partial\varphi/\partial x$</u>

$\partial\varphi/\partial x$ is obtained by differentiation of the integral in Eq. (4) to first obtain an exact expression which is then approximated by evaluating the result over the segmented boundary. This is felt to be preferable to the procedure of differentiating the approximation to φ given in Eq. (11). However, some care must be exercised when differentiating since the path of integration C(x) of the integral in Eq. (4) is itself a function of x. The details of this process are supplied in Appendix A. The resulting expression for $\partial\varphi/\partial x$ is found to be:

$$\frac{\partial\varphi}{\partial x} = -2\mathrm{Re}\left\{\oint_{C(x)}\left[\left(\frac{\delta\sigma}{\delta x}\right)_o + \frac{d\sigma}{ds}(\alpha\sin\theta + \gamma\cos\theta) + \frac{\sigma}{h}\left(\frac{\delta\nu}{\delta x}\right)\right]\ln(Z-\zeta)ds + i\oint_{C(x)}\sigma\left(\frac{\delta\nu}{\partial x}\right)\frac{d\zeta}{Z-\zeta}\right\} \quad (12)$$

which after integration over the segmented boundary C_n yields:

$$\frac{\partial\varphi}{\partial x_{j,n}} = 2\sum_{1}^{iL}\left\{\left[\left(\frac{\delta\sigma}{\delta x}\right)_o + (\alpha\sin\theta + \beta\cos\theta)\frac{d\sigma}{ds} + \frac{\sigma}{h}\left(\frac{\delta\nu}{\delta x}\right)\right]_{i,n}\left\{\frac{\Delta\varphi(i,j,n)}{\sigma(i,n)}\right\} - \sigma(i,n)\left(\frac{\delta\nu}{\delta x}\right)_{i,n}\delta(i,j,n)\right\} \quad . \quad (13)$$

The radius of curvature h(i, n) and the derivatives $(\delta\sigma/\delta x)_o$, $d\sigma/ds$, $\delta\nu/\delta x$ are evaluated at the mid points of the segments i, i+1 by interpolation procedures described in Appendix A.

Calculations of $\partial\varphi/\partial x$ for the circular cone at angle of attack are presented in Fig. (6). For comparison, the analytical result for bodies of revolution is:

$$\frac{\partial\varphi}{\partial x} = \frac{S''(x)}{2\pi}\ln r - 3\alpha\frac{S'(x)}{2\pi r}\cos\theta - \alpha^2\frac{S(x)}{\pi r^2}\cos 2\theta \quad (14)$$

A plot of Eq. (14) for points on the cone surface is also provided in Fig. (6).

5. Velocity Components v, w and $q^2 = v^2 + w^2$

Differentiation of Eq. (4) with respect to Z yields the complex velocity function

$$v - iw = -2 \oint \frac{\sigma(\zeta)}{Z-\zeta} \, ds \tag{15}$$

which, upon integration over the segmented boundary yields:

$$v(j,n) - iw(j,n) = 2 \sum_i \sigma(i,n) \, e^{-i\theta(i,n)} \left[\ln \frac{R(i+1,j,n)}{R(i,j,n)} + i\delta(i,j,n) \right] \tag{16}$$

q^2 is most conveniently found by noting that it is the sum of the squares of the normal and tangential velocity components. Thus, upon introducing the boundary condition Eq. (2b):

$$q^2 = \left(\frac{\partial \nu}{\partial x}\right)^2 + (v \cos\theta + w \sin\theta)^2 \tag{17}$$

on the segment $j, j+1$ this becomes

$$q^2(j,n) = \left(\frac{\partial \nu}{\partial x}\right)^2_{j,n} + \left\{ 2 \sum_i \sigma(i,n) \left[\cos\Big(\theta(j,n) - \theta(i,n)\Big) \ln \frac{R(i+1,j,n)}{R(i,j,n)} - \delta(i,j,n) \sin\Big(\theta(j,n) - \theta(i,n)\Big) \right] \right\}^2 \tag{18}$$

6. Pressure Coefficient C_p and $g'(x)$

C_p depends upon q^2 and $\partial\nu/\partial x$ as determined above and the derivative $g'(x)$. Differentiation of $g(x)$ must be carried out with due concern for the nature of the improper integrals appearing in Eq. (5). The result of the differentiation process as given in Appendix A, Sect. (5) is:

$$g'(x_n) = \frac{1}{4\pi} \left\{ S''(x) \ln\left(\frac{1-M^2}{4}\right) + I_n(x_n) - J_n(x_n) - \frac{S'(0)}{x_n} + \frac{S'(1)}{1-x_n} - S''(0) \ln x_n - S''(1) \ln(1-x_n) \right\} \tag{19}$$

where

$$I_n = \int_{x_n}^{1} \ln(x_n - t)\, S'''(t)\,dt = \sum_{m=n}^{N-1} \left(S''_{m+1} - S''_m\right) \ln(x'_m - x_n)$$

$$J_n = \int_{0}^{x_n} \ln(x_n - t)\, S'''(t)\,dt = \sum_{m=0}^{n-1} \left(S''_{m+1} - S''_m\right) \ln(x_n - x'_m)$$

$$x'_m = (x_{m+1} + x_m)/2 \ .$$

To compute the second derivatives of the cross sectional area required for $g'(x)$ the first derivatives at x'_m are found by finite differences between x_m and x_{m+1}. Second derivatives $S''(x''_m)$ at $x''(m) = (x'_{m+1} + x'_m)/2$ are then found by finite differences between S' at x'_m and x'_{m+1}. Finally $S''(x_m)$ is determined by linear interpolation of $S''(x''_m)$ between x''_m and x''_{m+1}.

Because of the possible singularity at $x=0$ the results are sensitive to the value of $S''(0)$. Rather than compute this second derivative from discrete data it is assumed that the nose of the body may be specified analytically and that an analytically derived value is available for $S''(0)$.

The pressure coefficient

$$C_p = \frac{p - p_o}{\rho U^2/2} = -2\left(\frac{\partial\varphi}{\partial x} + g'(x)\right) - q^2 \tag{20}$$

may now be computed. The computational precision may be evaluated by comparison with the analytical results for a conical body of revolution. In this special case we obtain for points on the surface of the body

$$g'(x) = -\frac{S''(x)}{2\pi} \ln\left(2\sqrt{x(1-x)}\right) + \frac{S'(1)}{1-x} \cdot \frac{1}{4\pi} \tag{21}$$

and

$$q^2 = \left(\frac{S'(x)}{2\pi r}\right)^2 + 2\alpha \frac{S'(x)}{2\pi r} \cos\theta + \alpha^2 \tag{22}$$

with $\partial\varphi/\partial x$ as given by Eq. (14).

Computed values of C_p for the conical body are presented in Fig. (7) together with the analytical results obtained from Eq. (20) and Eqs. (14, 21, 22).

7. Force and Moment Coefficients

From Eqs. (7, 8) for the force and moment coefficients it is seen that a determination of the "doublet" strength $A_1(x)$ is required. This term represents the coefficient of $1/Z$ in the expansion of the complex potential $W(Z)$ about the origin (see Eq. (3)). $A_1(x)$ as derived in Appendix A, Sect. (3) is given by:

$$A_1(x_n) = A_{10}(x_n) + (\Psi + i\alpha)\, x_n \frac{S'(x_n)}{2\pi} \tag{23}$$

where

$$A_{10}(x_n) = 2 \sum_i \sigma(i, n)\Gamma(i, n)(y'_{i,n} + i\, z'_{i,n}) \quad .$$

To obtain force and moment coefficients $A_1(x)$ is substituted into Eqs. (7) and (8) which may now be written in a more convenient form by introducing the centroid location Z_g in terms of body axes coordinates.

$$Z_g = Z_{go} - (i\alpha + \Psi)\, x \ . \tag{24}$$

The resulting force and moment equations are:

$$\frac{F_y + i\, F_z}{\rho U^2} = 2\pi\, A_{10}(x) - (\Psi + i\alpha)\, S + (Z_{go}\, S)' \tag{25}$$

$$\frac{M_y + i\, M_z}{\rho U^3} = i \left\{ x \frac{F_y + iF_z}{\rho U^2} - \int_0^x [2\pi\, A_{10}(x) - (\Psi + i\alpha)S]dx - Z_{go}S \right\} . \tag{26}$$

Numerical evaluation of the integral in the expression for the moment coefficient is carried out by the trapezoidal rule using values of $A_{10}(x_n)$ and $S(x_n)$ obtained at each of the stations x_n. Computation of $Z_{go}(x)\, S(x)$

is described in Appendix A. The derivative of $Z_{go}S$ at x_n is obtained by first computing the divided difference between stations x_n, and x_{n+1}, then letting this represent $(Z_{go}S)'$ at x_n'. The derivative at x_n is determined by linear interpolation of $[Z_{go}S'(x_n')]$ between x'_n and x'_{n+1}.

Analytical results in the case of bodies of revolution at angle of attack α are particularly simple:

$$\frac{F_z}{\rho U^2} = \alpha S(x) \tag{27}$$

$$\frac{M_y}{\rho U^2} = -\alpha\Big(x\, S(x) - V(x)\Big) \tag{28}$$

where V(x) is the volume of the body up to the station x.

Computational results for the cone and ogive bodies of revolution at $\alpha = \cdot 1$ are presented in Figs. 8 and 9, together with plots of Eqs. (27) and (28) for comparison. There results are presented in terms of the coefficients $C_L(x)$, and $C_M(x)$ defined at the end of Section A.

C. Application to Typical Fuselage

A typical "slab-sided" fuselage together with details of the geometry, is shown in Fig. 10. Cross-sections have been made of straight lines and circular arcs while the profile is composed of straight lines and parabolic arcs. Stations x_n have been taken closer together toward the rear of the body to promote a more accurate determination of total force and moment. Stations are situated farther apart over the center section since there is no change in cross-section for $1/3 < x < 2/3$.

Processing of the surface data in accordance with paragraph 1 of section B is shown in Fig. 11.

Results of the computation of pressure coefficient, force coefficient and moment coefficient are given in Figs. 12 and 13.

APPENDIX A

DERIVATIONS

1. Source Strength σ

Computation of $\sigma(i, n)$ over the segment $i, i+1$ proceeds by applying the boundary condition Eq. (2b) at each segment of C_n. If $\nabla\varphi = \overline{q} = \overline{j}v + \overline{k}w$ represents the velocity vector, the corresponding complex velocity in the crossflow plane is obtained by differentiation of W in Eq. (4) with respect to Z:

$$v - iw = -2\oint \frac{\sigma(\zeta)ds}{Z-\zeta} \tag{A1}$$

The contribution by the sources located on segment $i, i+1$ to the velocity at $P'_{j,n}$ is first evaluated. Noting that $i, i+1$ makes an angle $\theta(i, n)$ with respect to the horizontal axis, we have

$$d\zeta = ds\ e^{i\theta(i,n)}$$

and the contribution to the integral in Eq. (A1) may be written:

$$\Delta[v(j,n) - iw(j,n)] = -2\sigma(i,n)e^{-i\theta(i,n)} \int_{\zeta_{i,n}}^{\zeta_{i+1,n}} \frac{d\zeta}{Z_{j,n} - \zeta} \ . \tag{A2}$$

After integration of the last term and summation over all contributing segments, the result may be written:

$$v(j,n) - iw(j,n) = 2\sum_i \sigma(i,n)\ e^{-i\theta(i,n)}\left[\ln \frac{R(i+1,j,n)}{R(i,j,n)} + i\delta(i,j,n)\right] \tag{A3}$$

in which, referring to Fig. 3, the quantities $R(i, j, n)$ and $\delta(i, j, n)$ are defined by the relationships:

$$R(i,j,n)\ e^{i\psi(i,j,n)} = Z'_{j,n} - \zeta_{i,n}$$

$$\delta(i,j,n) = \psi'(i,j,n) - \psi(i,j,n) \ .$$

To insure uniqueness of the complex velocity, care must be exercised in assigning values to the angles $\psi(i,j,n)$ and $\psi'(i,j,n)$. Referring to Fig. 3, these are measured counter-clockwise from the positive y axis so that when facing from $P_{i,n}$ to $P_{i+1,n}$, a point $P'_{j,n}$ just to the left of $i, i+1$ shall define an angle $\psi(i,j,n) = \theta(i,n)$. As $P'_{j,n}$ traverses a path around $P_{i,n}$ to a point just to the right of $i, i+1$, $\psi(i,j,n)$ increases from $\theta(i,n)$ to $\theta(i,n) + 2\pi$. The same holds true for $\psi'(i,j,n)$ as $P'_{j,n}$ traverses a path aroung $P_{i+1,n}$. In consequence of these definitions $\delta(i,j,n)$ becomes $-\pi$ when approaching $i, i+1$ from the right and π when approaching from the left. This discontinuity reflects that exhibited by the stream function upon traversing any closed path which encloses a distribution of finite sources.

From the boundary condition Eq. (2b), we have:

$$\left(\frac{\partial\nu}{\partial x}\right)_{j,n} = v(j,n)\sin\theta(j,n) - w(j,n)\cos\theta(j,n)\ . \qquad \text{(A4)}$$

After substitution of v and w from Eq. (A3), this last expression becomes

$$\left(\frac{\partial\nu}{\partial x}\right)_{j,n} = \sum_i a(j,i)\sigma(i,n) \qquad \text{(A5)}$$

where

$$a(j,i) = 2\left\{\sin\Big(\theta(j,n) - \theta(i,n)\Big)\ln\frac{R(i+1,j,n)}{R(i,j,n)} + \delta(i,j,n)\cos\Big(\theta(j,n) - \theta(i,n)\Big)\right\}\ .$$

In addition, we see from Fig. 4 that the slope $\partial\nu/\partial x$ may be expressed in terms of the body slope $\partial\nu_o/\partial x$ referred to body axes:

$$\left(\frac{\partial\nu}{\partial x}\right)_{j,n} = \left(\frac{\partial\nu_o}{\partial x}\right)_{j,n} + \alpha\cos\theta(j,n) - \Psi\sin\theta(j,n) \qquad \text{(A6)}$$

thus eliminating the necessity of constructing a new set of projections similar to Fig. 2 for each set of α and Ψ. Satisfying Eq. (A5) at each of the points $P'_{j,n}$ on a given cross-sectional boundary yields a set of equations for $\sigma(i, n)$.

2. <u>Determination of ϖ, $\partial\varpi/\partial x$</u>

A knowledge of $\sigma(i, n)$ allows the numerical integration of Eq. (4) for ϖ in a manner similar to that for the complex velocity above:

$$\varpi(j, n) = -2\mathrm{Re}\sum_i e^{-i\theta(i, n)}\sigma(i, n)\int_{\zeta_{i,n}}^{\zeta_{i+1,n}} \ln(Z_{j,n} - \zeta)d\zeta \tag{A7}$$

After integration, $\varpi(j, n)$ may be written concisely in the nomenclature of Fig. 3:

$$\begin{aligned}\varpi(j, n) &= 2\sum_i \sigma(i, n)\Big\{\bar{R}(i+1, j, n)\cdot\vec{u}(i, n)\ln R(i+1, j, n) \\ &\quad - \bar{R}(i, j, n)\cdot\vec{u}(i, n)\ln R(i, j, n) - \bar{R}(i, j, n)\cdot\bar{n}(i, n)\,\delta(i, j, n) \\ &\quad + l(i, n)\Big\} = 2\sum_i \sigma(i, n)\left\{\frac{\Delta\varpi(i, j, n)}{\sigma(i, n)}\right\}\end{aligned} \tag{A8}$$

in which use has been made of the geometric relationship:

$$\bar{R}(i, j, n)\cdot\bar{n}(i, n) = \bar{R}(i+1, j, n)\cdot\bar{n}(i, n)\ .$$

The derivation of $\partial\varpi/\partial x$ must take into account the fact that the path of integration in Eq. (4) is a function of x. Referring to Fig. 1, we shall distinguish between increments of a dependent variable taken along C(x) and denoted by d( ) and increments taken normal to C and denoted by δ( ). Differentiation of Eq. (4) then yields

$$\begin{aligned}\frac{\partial\varpi}{\partial x} &= -2\,\mathrm{Re}\left\{\oint \frac{\delta\sigma}{\delta x}\ln(Z - \zeta)ds - \oint \frac{\sigma(\zeta)}{Z-\zeta}\,\frac{\delta\zeta}{\delta x}\,ds\right. \\ &\quad \left. + \oint \sigma(\zeta)\ln(Z - \zeta)\,\frac{\delta(ds)}{\delta x}\right\}\end{aligned} \tag{A9}$$

From Fig. 1 it becomes evident that

$$\delta(ds) = \delta\nu\, d\theta = \delta\nu \frac{ds}{h(\zeta)} \tag{A10}$$

where $h(\zeta)$ is the radius of curvature of $C(x)$ at ζ. In addition, we have from Fig. 1,

$$\frac{\delta\zeta}{\delta x} = \frac{\delta\nu}{\delta x} e^{i(\theta - \pi/2)} \tag{A11}$$

To evaluate $\delta\sigma/\delta x$ we note, referring to Fig. 4,

$$\frac{\delta\sigma}{\delta x} = \lim_{\delta x \to 0} \frac{\sigma'' - \sigma(i,n)}{\delta x} \tag{A12}$$

where σ'' denotes the value of σ at the point P''. The relative displacement between $P_{i,n}$ and P'' is shown in Fig. 4, as it would appear in "wind" axes. However, the computation of σ has been carried out in a body axis frame of reference. To make use of the results of that computation we note that σ'' in the wind axis frame corresponds to σ' in the body axis frame. From Fig. 4 then, we have

$$\sigma'' = \sigma' = \sigma(i,n+1) + \frac{d\sigma}{ds}(\alpha \sin\theta + \Psi \cos\theta)\delta x \tag{A13}$$

which, after substitution into Eq. (A12), leads to the required expression,

$$\frac{\delta\sigma}{\delta x} = \left(\frac{\delta\sigma}{\delta x}\right)_o + \frac{d\sigma}{ds}(\alpha \sin\theta + \Psi \cos\theta) \tag{A14}$$

where $\left(\frac{\delta\sigma}{\delta x}\right)_o$ is the derivative evaluated in the body axis frame. Finally, introducing Eqs. (A10), (A11), and (A14) into Eq. (A9),

$$\frac{\partial\phi}{\partial x} = -2\,\mathrm{Re}\left\{ \oint \left[\left(\frac{\delta\sigma}{\delta x}\right)_o + \frac{d\sigma}{ds}(\alpha \sin\theta + \Psi \cos\theta) + \frac{\sigma}{h}\frac{\delta\nu}{\delta x} \right] \ln(Z-\zeta)\, ds \right.$$

$$\left. + i \oint \left[\sigma \frac{\delta\nu}{\delta x} \right] \frac{d\zeta}{Z-\zeta} \right\} \tag{A15}$$

Again, assuming that quantities in the brackets of the integrands are constant over $i, i+1$, the integrations proceed in a straightforward manner:

$$\left(\frac{\partial \varphi}{\delta x}\right)_{j,n} = 2\sum_{i}\left\{\left[\left(\frac{\delta\sigma}{\delta x}\right)_{o} + \frac{d\sigma}{ds}(\alpha \sin\theta + \Psi\cos\theta)\,\frac{\omega}{ds} + \frac{\sigma}{h}\,\frac{\delta\nu}{\delta x}\right]_{i,n}\left\{\frac{\Delta\varphi(i,j,n)}{\sigma(i,n)}\right\}\right.$$

$$\left. - \sigma(i,n)\left(\frac{\delta\nu}{\delta x}\right)_{i,n}\varphi(i,j,n)\right\} \qquad (A16)$$

in which we note that $(\delta\nu/\delta x) \equiv \partial\nu/\partial x$ as defined in Eq. (A6).

Equations defining $(d\sigma/ds)$, $(\delta\sigma/\delta x)_o$, $\delta\nu_o/\delta x$ and $1/h$ at the point $P'_{i,n}$ are provided in Sections C-1 and C-3 in Part II of this report. A description of the computational process is given here:

a) $d\sigma/ds$ - σ at $P_{i,n}$ is first obtained by interpolation between the computed values of $\sigma(i,n)$ at $P'_{i,n}$. $d\sigma/ds$ at $P'_{i,n}$ is then set equal to the divided difference between these interpolated values of σ. (see Section C-3 of Part II).

b) $(\delta\sigma/\delta x)_o$ - the derivative at the mid-point x'_n of the interval x_n, x_{n+1} is set equal to the divided difference between $\sigma(i,n)$ and $\sigma(i,n+1)$. Linear interpolation between these derivatives then yields $(\delta\sigma/\delta x)_o$ at x_n. (see Fig. 14 and Section C-3 of Part II).

c) $\delta\nu_o/\delta x$ - Referring to Fig. 15, the displacement δr is determined by interpolation between $\delta\zeta_{i,n}$ and $\delta\zeta_{i+1,n}$. $\delta r/(x_{n+1} - x_n)$ then represents $\delta\nu_o/\delta x$ at x'_n. Interpolation between the stations x'_n then yields $\delta\nu_o/\delta x$ at x_n (see Section C-1 of Part II).

d) $1/h$ - θ at $P_{i,n}$ is determined by interpolation between values of $\theta(i,n)$ at $P'_{i,n}$. The curvature $1/h$ at $P'_{i,n}$ is then set equal to the divided difference between θ at $P_{i+1,n}$ and θ at $P_{i,n}$. (see Section C-3 of Part II).

3. "Doublet Strength" $A_1(x)$

$A_1(x)$ is the coefficient of the $1/Z$ term in the expansion of the complex potential $W(Z)$ about the origin (see Eq. (3)). If the integral representation of W from Eq. (4) is expanded we find:

$$W(Z) = \left(-2\oint \sigma(\zeta)ds\right)\ln Z + \left(2\oint \zeta\, \sigma(\zeta)ds\right)\frac{1}{Z}$$

$$+ \left(2\oint \zeta^2\, \sigma(\zeta)ds\right)\frac{1}{Z^2} + \ldots \quad , \tag{A17}$$

Thus, we have for the coefficient of the $1/Z$ term:

$$A_1(x) = 2\oint \zeta\, \sigma(\zeta)ds$$

Introducing body axes coordinates

$$\zeta = \zeta_o - (i\alpha + \beta)x$$

we have

$$A_1(x) = 2\oint \zeta_o\, \sigma(\zeta_o)ds - 2(i\alpha + \beta)\, x \oint \sigma(\zeta_o)ds$$

The last integral on the right hand side is recognized as the coefficient of the "source" term in the above expansion of $W(Z)$. According to slender body theory Ref. (1), this is related to the rate of change of cross-sectional area:

$$2\oint \sigma(\zeta_o)ds = -\frac{S'(x)}{2\pi}$$

our final expression for the "doublet" term is therefore

$$A_1(x) = 2\oint \zeta_o\, \sigma(\zeta_o)ds + (i\alpha + \beta)x\,\frac{S'(x)}{2\pi}$$

Integrating over the segmented boundary C_n.

$$\int \zeta_o\, \sigma(\zeta_o)ds = \sum_i \sigma(i,n)\int_i^{i+1} \zeta_o\, ds$$

the last integral may be interpreted as the moment of the arc i, i+1 about the origin and may be approximated by $(y'_{i,n} + iz'_{i,n})\, l(i,n)$ so that

$$A_1(x) = 2 \sum \sigma(i,n)\, l(i,n)\, (y'_{i,n} + i z'_{i,n}) \qquad \text{(A18)}$$

4. Cross-sectional Properties

Computation of $S(x_n)$, $Z_{go} S(x_n)$ and their derivatives is accomplished with the aid of Stokes' theorem in the complex plane. Thus,

$$S(x) = \frac{1}{2i} \oint \bar{\zeta}_o \, d\zeta_o \qquad \text{(A19)}$$

$$Z_{go} S(x) = \frac{1}{2i} \oint \zeta_o \bar{\zeta}_o \, d\zeta_o \qquad \text{(A20)}$$

which expressions, after integration around C_n, yield

$$S(x_n) = \frac{1}{2} \sum (y'_{i,n} dz_{i,n} - z'_{i,n} dy_{i,n}) \qquad \text{(A21)}$$

and

$$Z_{go} S(x) = \frac{1}{2} \sum_i r^2_{i,n} (dz_{i,n} - i\, dy_{i,n}) \qquad \text{(A22)}$$

where

$$r^2_{i,n} = (y'_{i,n})^2 + (z'_{i,n})^2$$

$$dz_{i,n} = z_{i+1,n} - z_{i,n}$$

$$dy_{i,n} = y_{i+1,n} - y_{i,n} \quad .$$

5. $g'(x)$

The derivative of $g(x)$ appears in the expression for the local pressure coefficient, Eq. (6). To avoid the occurrence of singular integrals, differentiation is accomplished by first integrating by parts the integrals appearing in Eq. (5) for $g(x)$ and then differentiating the resulting expressions

$$\int_0^x S''(t) \ln (x-t) dt = - S''(0)(x - x \ln x) - \int_0^x S'''(t)[(x-t) - (x-t)\ln (x-t)] dt$$

then

$$\frac{\partial}{\partial x}\int_0^x S''(t)\ln(x-t)dt = S''(0)\ln x + \int_0^x S'''(t)\ln(x-t)dt$$

and similarly

$$\frac{\partial}{\partial x}\int_x^1 S''(t)\ln(t-x)dt = -S''(1)\ln(1-x) + \int_x^1 S'''(t)\ln(t-x)dt$$

Thus, differentiation of Eq. (5) for g(x) yields:

$$g'(x) = \frac{1}{2\pi}\Big\{S''(x)\ln(\tfrac{\beta}{2}) + \frac{1}{2}\int_x^1 S'''(t)\ln(t-x)dt$$

$$-\frac{1}{2}\int_0^x S'''(t)\ln(x-t)dt - \frac{S'(0)}{2}\cdot\frac{1}{x}$$

$$+\frac{S'(1)}{2}\cdot\frac{1}{1-x} - \frac{S''(0)}{2}\ln x - \frac{S''(1)}{2}\ln(1-x)\Big\}$$

Expressing the integrals as Stieltjes integrals facilitates their computation.

$$I_n = \int_{x_n}^1 \ln(t-x_n)d\,S''(t) = \sum_{m=n}^{N-1}(S''_{m+1} - S''_m)\ln(x'_n - x_n)$$

and

$$J_n = \int_0^{x_n} \ln(x_n-t)d\,S''(t) = \sum_{m=0}^{n-1}(S''_{m+1} - S''_m)\ln(x_n - x'_m)$$

where $x'_m = (x_m + x_{m+1})/2$

we thus have

$$g'(x_n) = \frac{1}{4\pi}\Big\{S''(x_n)\ln\Big(\frac{1-M^2}{4}\Big) + I_n - J_n - \frac{S'(0)}{x_n} + \frac{S'(1)}{1-x_n}$$

$$- S''(0)\ln x_n - S''(1)\ln(1-x_n)\Big\} \qquad \text{(A26)}$$

The occurence of singularities in g(x) and g'(x) at x=0, 1 signifies the failure of slender body theory in these regions unless S is sufficiently well behaved there i.e., first and second derivatives equal to 0. For pointed bodies $S'(0) = 0$ and the occurrence of $S'(1) = 0$ is common.

REFERENCES

1. Ward, G. N.: "Linearized Theory of Steady High Speed Flow." Cambridge University Press, 1955.
2. Hess, J. F. and Smith, A. M. O.: "Calculation of Potential Flow About Arbitrary Bodies." Prog. Aero. Sci., Pergamon Press, 1966.

PART II
FORTRAN PROGRAM

A. Input

1. Comments

The body axes coordinates $y_{i,n}$, $z_{i,n}$ at x_n may be read from cards or computed by a code supplied by the user; the indices IX and IR are set equal to 0 or 1 depending upon the choice made. After the source strength σ is computed the program computes φ, $\partial\varphi/\partial x$, C_p at the locations $P'_{i,n}$ on the surface. The capability of computing these quantities at arbitrarily specified points on or off the body has also been included to facilitate induced flow studies. Thus φ, $\partial\varphi/\partial x$, C_p are computed at $P'_{i,n}$ or at locations supplied by the user as additional input, depending upon whether the index IYPP is set equal to 0 or 1

2. List of Fortran Symbols for Input Data

ALP	Angle of attack α, positive for nose up attitude relative to wind axes.
BET	Angle of yaw Ψ, positive for clockwise rotation about z-axis.
ACH	Free Stream Mach No.
SPPO	$S''(0)$ Second derivative of cross-section area evaluated at the nose. It is assumed that this is available from analytical considerations regarding the special geometry of the nose section.
SREF	Dimensional reference area.
ENG	Dimensional body length.
REFL	Dimensional reference length.

IYPP — =0 if coordinates of $P'_{i,n}$ are computed by program,
=1 if $P'_{i,n}$ are to be read from input cards.

IL — Number of segments into which a cross-sectional boundary is divided .

NL — Number of longitudinal stations at which cross-sections are taken.

IR — =1 if $y_{i,n}$, $z_{i,n}$ are to be read from input cards.
=0 if these cards are to be computed by a code inserted after statement 111.

IX — =1 if x_n are to be read from input cards.
=0 if these stations are to be computed by a code inserted after statement 113.

ISYMLR = 0 — if contour does not have lateral symmetry
= 1 — if contour has lateral symmetry

ISYMUD = 0 — if contour does not have vertical symmetry
= 1 — if contour has vertical symmetry

ISR — if ≠ 0 SREF will be defined = S(ISR)

X(N) — Dimensional longitudinal coordinates x_n.

Y(I, N) — Dimensional coordinate $y_{i,n}$

Z(I, N) — Dimensional coordinate $z_{i,n}$

YPP(I) — Dimensional coordinate of collocation pt. $y'_{i,n}$.

ZPP(I) — Dimensional coordinate of collocation pt. $z'_{i,n}$.

3. Preparation of Input Cards

Card #	Format	Variable
1	5E15.8	ALP BET ACH SPPO SREF
2	5E15.8	ENG REFL

3	1015	IYPP IL NL IR IX ISYMLAR ISYMUD ISR

The following cards are prepared in the order presented, when the indices IX, IR, IYPP are as specified

If IX=1	10F8.0	X(1) X(2) . . X(NL)
If IR=1 If ISMLR = 1, ISYMUD = 0, or 1 I=1 placed in 4th quadrant I = IL placed in 3rd quadrant If ISYMLR = 0 ISYMUD = 1 I = 1 placed in 1st quadrant I = IL placed in 4th quadrant If ISYMLR = ISYMUD = 0 no restriction on placement of I=1	10F8.0	Y(1,1) Y(1,2) . . Y(1,NL) Z(1,1) Z(1,2) Z(1,NL) Y(2,1) Y(2,1) . . Y(2,NL) Z(2,1) . . Z(2,NL) . . Z(IL,NL)
If IYPP = 1	5E15.8	YPP(1) ZPP(1) YPP(2) ZPP(2) . . YPP(IL) ZPP(IL)

If IR=0 A code to compute $y_{i,n}$, $z_{i,n}$ must be inserted after statement 111.

If IX=0 A code to compute x_n must be inserted after statement 113.

B. <u>Output</u>

1. <u>Input parameters</u>

The first row of output presents the pertinent input parameters ALPHA, BETA, MACH NO., SPP(0), REF AREA, BODY LENGTH, REF LENGTH.

2. $\underline{\sigma, \varphi, y', z'}$

$\sigma(j, n)$ and $\varphi(j, n)$ at the location $y'_{j,n}$ $z'_{j,n}$ are presented as follows for $1 \leq n \leq N$

n

SIGMA

$\sigma(1, n)$ - - - - - - - - - - - - - $\sigma(7, n)$
$\sigma(8, n)$ - - - $\sigma(IL, n)$

PHI

$\varphi(1, n)$ - - - - - - - - - - - - - $\varphi(7, n)$
$\varphi(8, n)$ - - - $\varphi(IL, n)$

Y PRIME
Z PRIME

y'_{in} - - - - - - - - - - - - - $y'_{7,n}$
z'_{in} - - - - - - - - - - - - - $z'_{7,n}$

$y'_{8,n}$ - - - $y'_{IL,n}$
$z'_{8,n}$ - - - $z'_{IL,n}$

3. $\partial\varphi/\partial x$

$(\partial\varphi/\partial x)_{j,n}$ at the points $P'_{j,n}$ are presented as follows:

D PHI/D X

$(\partial\varphi/\partial x)_{1,1}$ - - - - - - - - - - - - - $(\partial\varphi/\partial x)_{7,1}$
$(\partial\varphi/\partial x)_{8,1}$ - - - - $(\partial\varphi/\partial x)_{IL,1}$

- -

- -

$(\partial\varphi/\partial x)_{1,NL}$ - - - - - - - - - - - - $(\partial\varphi/\partial x)_{7,NL}$
$(\partial\varphi/\partial x)_{8,NL}$ - - - $(\partial\varphi/\partial x)_{IL,NL}$

4. $AR_{10}(x_n)$, $AI_{10}(x_n)$

Real and imaginary parts of the "doublet strength" $A_{10}(x_n)$ are presented as follows:

AI AND AR

$AI_{10}(x_1)$, $AR_{10}(x_1)$, $AI_{10}(x_2)$, $AR_{10}(x_2)$ - - - $AI_{10}(x_4)$
$AR_{10}(x_4)$ - - - - - - - - $AI_{10}(x_N)$, $AR_{10}(x_N)$

5. Force and Moment coefficients, $g'(x_n)$, Pressure Coefficient

Pressure coefficient C_p at $P'_{j,n}$ is computed for $1 \leq n \leq N-1$. Force and moment coefficients are presented as follows:

N = n, CY = $C_y(x_n)$, CL = $C_L(x_n)$, CN = $C_N(x_n)$
CM = $C_M(x_n)$, GP = $g'(x_n)$

$C_p(1,n)$ - - - - - - - - - - - - - - $C_p(7,n)$

$C_p(8,n)$ - - - - - - $C_p(IL,n)$

ORIGINAL PAGE IS
OF POOR QUALITY

C. Summary of Programmed Equations

These equations are presented in order of use. The Fortran svmbol at the left represents the quantity at the left hand side of each equation.

1) Computation of $\sigma(i, n)$

Y(ILP, N)	$y_{iL+1,n} = y_{i,n}$	
Z(ILP, N)	$z_{iL+1,n} = z_{i,n}$	
Y(ILP, N)	$y_{iL+1} = y_{2,n}$	
Y(IL2, N)	$y_{iL+2} = y_{2,n}$	
Y(IL3, N)	$y_{iL+3} = y_{3,n}$	
Y(IL4, N)	$y_{iL+3} = y_{4,n}$	
Fl(ILP, N)	$l(iL+1) = l(1,n)$	
Fl(IL2, N)	$l(iL+2) = l(2,n)$	
Fl(IL3, N)	$l(iL+3) = l(3,n)$	
DPY(I)	$D'y_i = (y_{i+1,n} - y_{i,n})/l(i,n)$	$1 \le i \le iL + 3$
DY(I)	$Dy_i = \dfrac{D'y_{i-1}l(i,n) + D'y_i l(i-1,n)}{l(i,n) + l(i-1,n)}$	$2 \le i \le iL + 3$
DPY(I)	$D''y_i = (Dy_{i+1} - Dy_i)/l(i,n)$	$2 \le i \le iL + 2$
YP(I)	$DDy_i = \dfrac{D''y_{i-1}l(i,n) + D''y_i l(i-1,n)}{l(i,n) + l(i-1,n)}$	$3 \le i \le iL + 2$
YP(I)	$y_i' = y_{i,n} + Dy_i \dfrac{l(i,n)}{2} + \dfrac{DDy_i}{2}\left(\dfrac{l(i,n)}{2}\right)^2$	$3 \le i \le iL + 2$
YP(1)	$y_1' = y_{iL+1}'$	
YP(2)	$y_2' = y_{iL+2}'$	

The above operations from Y(ILP, N) to YP(2) are repeated for Z(ILP, N) to ZP(2) to obtain z_i'.

R(I, J)	$R(i,j,n) = \left[(y'_{j,n} - y_{i,n})^2 + (z'_{j,n} - z_{i,n})^2\right]^{\frac{1}{2}}$
FL(I, N)	$l(i,n) = \left[(y_{i+1,n} - y_{i,n}) + (z_{i+1,n} - z_{i,n})^2\right]^{\frac{1}{2}}$
ST(I)	$\sin\theta(i,n) = (z_{i+1,n} - z_{i,n})/l(i,n)$
CT(I)	$\cos\theta(i,n) = (y_{i+1,n} - y_{i,n})/l(i,n)$

For the computation of angles it is assumed that a computer will obey the following rules:

$$0 < \sin^{-1}\sin\theta < \pi/2 \quad , \quad \sin\theta\ (+)$$
$$-\pi/2 < \sin^{-1}\sin\theta < 0 \quad , \quad \sin\theta\ (-)$$
$$0 < \cos^{-1}\cos\theta < \pi/2 \quad , \quad \cos\theta\ (+)$$
$$\pi/2 < \cos^{-1}\cos\theta < \pi \quad , \quad \cos\theta\ (-)\ .$$

T(I, N)

$$\theta(i,n) = \begin{cases} \sin^{-1}\sin\theta(i,n) & \sin\theta(i,n) \geq 0,\ \cos\theta(i,n) \geq \\ \pi - \sin^{-1}\sin\theta(i,n) & \sin\theta(i,n) \geq 0,\ \cos\theta(i,n) < 0 \\ -\sin^{-1}\sin\theta(i,n) & \sin\theta(i,n) < 0,\ \cos\theta(i,n) < 0 \\ 2\pi - \sin^{-1}\sin\theta(i,n) & \sin\theta(i,n) < 0,\ \cos\theta(i,n) \geq 0 \end{cases}$$

AS	$\sin\nu(i,j,n) = (z'_{j,n} - z_{i,n})/R(i,j,n)$
ZZ	$\Delta z = z'_{j,n} - z_{i,n}$
YY	$\Delta y = y'_{j,n} - y_{i,n}$

G

$$\gamma(i,j,n) = \begin{cases} \sin^{-1}\sin\gamma(i,j,n) & \Delta z \geq 0,\ \Delta y \geq 0 \\ \pi - \sin^{-1}\sin\gamma(i,j,n) & \Delta z \geq 0,\ \Delta y < 0 \\ \pi - \sin^{-1}\sin\gamma(i,j,n & \Delta z < 0,\ \Delta y < 0 \\ 2\pi + \sin^{-1}\sin\gamma(i,j,n) & \Delta z < 0,\ \Delta y \geq 0 \end{cases}$$

P(J, I) $$\psi(i,j,n) = \begin{cases} \gamma(i,j,n) & , \quad \gamma(i,j,n) > \theta(i,n) \\ \gamma(i,j,n) + 2 & , \quad \gamma(i,j,n) \le \theta(i,n) \end{cases}$$

PHS $$\psi'(i,j,n) = \begin{cases} \gamma(i+1,j,n) & , \quad \gamma(i+1,j,n) > \theta(i,n) \\ \psi(i,j,n) & , \quad \gamma(i+1,j,n) = \theta(i,n) \\ \gamma(i+1,j,n) + 2\pi & , \quad \gamma(i+1,j,n) < \theta(i,n) \end{cases}$$

D(J, I, N) $\delta(i,j,n) = \psi'(i,j\ n) - \psi(i,j,n)\ , \quad i \neq j$

D(J, I, N) $\delta(j,j,n) = -\pi$

The following redefinitions of $\theta(i,n)$ assure continuity of $\theta(i,n)$ when passing directly between first and fourth quadrants:

$$\theta(iL+1,n) = \theta(1,n)$$

$$\Delta\theta = \theta(i+1,n) - \theta(i,n)$$

$$\theta(i+1,n) = \begin{cases} \theta(i+1,n) + 2\pi & , \quad \Delta\theta < -(\pi + 10^{-5}) \\ \theta(i+1,n) & , \quad -\pi < \Delta\theta < \pi \\ \theta(i+1,n) - 2\pi & , \quad \Delta\theta > \pi + 10^{-5} \end{cases}$$

FL(IL+1, N) $l(iL+1,n) = l(i,n)$

BE(I, N) $$\kappa(i+1,n) = \theta(i,n) + \frac{[\theta(i+1,n) - \theta(i,n)]\, l(i,n)}{l(i+1,n) + l(i,n)}\ , \quad 1 \le i < iL$$

BE(1, N) $\kappa(1,n) = \kappa(iL+1,n) - 2\pi$

DR(I) $$\delta\nu_o(i,n) = (y_{i,n+1} - y_{i,n}) \sin \kappa(i,n)$$
$$- (z_{i,n+1} - z_{i,n}) \cos \kappa(i,n) \qquad 1 \le n \le n-1$$

DNX(I) $$\left(\frac{\delta\tau}{\delta x}\right) = \frac{[\delta\nu_o(i,n) + \delta\nu_o(i+1,n)]/2}{x_{n+1} - x_n} \qquad 2 \le n \le N-1$$

DN(I, N) $\left(\frac{\partial \nu_o}{\partial x}\right)_{i,n} = \left(\frac{\delta \eta}{\delta x}\right)_{i,n-1} + \frac{[(\delta\eta/\delta x)_{i,n} - (\delta\eta/\delta x)_{i,n-1}](x_n - x_{n-1})}{x_{n+1} - x_{n-1}}$

$2 \leq n \leq N-1$

DN(I, N) $\left(\frac{\partial \nu_o}{\partial x}\right)_{i,N} = \left(\frac{\delta \eta}{\delta x}\right)_{i,N-1}$

DN(I, 1) $\left(\frac{\partial \nu_o}{\partial x}\right)_{i,1} = \left(\frac{\delta \eta}{\delta x}\right)_{i,1}$

DN(I, N) $\left(\frac{\partial \nu}{\partial x}\right)_{i,n} = \left(\frac{\partial \nu_o}{\partial x}\right)_{i,n} + Z \cos\theta(i,n) - Y \sin\theta(i,n)$

STT $\sin[\theta(j,n) - \theta(i,n)] = \sin\theta(j,n)\cos\theta(i,n) - \sin\theta(i,n)\cos\theta(j,n)$

CTT $\cos[\theta(i,n) - \theta(i,n)] = \cos\theta(j,n)\cos\theta(i,n) + \sin\theta(j,n)\sin\theta(i,n)$

AJI( ) $a(i,j,n) = 2\left\{\sin[\theta(j,n) - \theta(i,n)]\ln\frac{R(i+1,j,n)}{R(i,j,n)}\right.$

$\left. + \cos[\theta(j,n) - \theta(i,n)]\,\delta(i,j,n)\right\}$

SIG(I, N) $\sigma(i,n) = |a(j,i,n)|^{-1}\left\{\left(\frac{\partial \nu}{\partial x}\right)_j\right\}$

2) Computation of $\varphi(j,n)$

RT $\vec{R}(i,j,n)\cdot\vec{u}(i,n) = (y'_{j,n} - y_{i,n})\cos\theta(i,n) + (z'_{j,n} - z_{i,n})\sin\theta(i,n)$

Ru $\vec{R}(i+1,j,n)\cdot\vec{u}(i,n) = (y'_{j,n} - y_{i+1,n})\cos\theta(i,n)$

$+ (z'_{j,n} - z_{i+1,n})\sin\theta(i,n)$

RN $\vec{R}(i,j,n)\cdot\vec{n}(i,n) = -(y'_{j,n} - y_{i,n})\sin\theta(i,n)$

$+ (z'_{j,n} - z_{i,n})\cos\theta(i,n)$

DT(J, I, N) $\left\{\frac{\delta\varphi(i,j)}{\sigma(i)}\right\} = \vec{R}(i+1,j,n)\cdot\vec{u}(i,n)\ln R(i+1,j,n)$

$- \vec{R}(i,j,n)\cdot\vec{u}(i,n)\ln R(i,j,n)$

$- \vec{R}(i,j,n)\cdot\vec{n}(i,n)\,\delta(i,j,n) + 1(i,n)$

PH(J) $\varphi(j,n) = 2\sum_i \sigma(i,n)\left\{\frac{\Delta\varphi(i,j,n)}{\sigma(i,n)}\right\}$

3) Computation of $(\partial\varphi/\partial x)_{j,n}$

SIG(1LP, N) $\sigma(iL+1,n) = \sigma(1,n)$

SII $\Sigma(i+1,n) = \frac{[\sigma(i+1,n) - \sigma(i,n)]\, l(i,n)}{l(i+1,n) + l(i,n)} + \sigma(i,n)$

SS $\left(\frac{d\sigma}{ds}\right)_{i,n} = \frac{\Sigma(i+1,n) - \Sigma(i,n)}{l(i,n)}$

XIN(I) $\left(\frac{\Delta\sigma}{\Delta x}\right)_{i,n} = \frac{\sigma(i,n+1) - \sigma(i,n)}{x_{n+1} - x_n}$ $\quad 2 \le n \le N-1$

DSX $\left(\frac{\delta\sigma}{\delta x}\right)_{oi,n} = \left(\frac{\Delta\sigma}{\Delta x}\right)_{i,n-1} + \left[\left(\frac{\Delta\sigma}{\Delta x}\right)_{i,n} - \left(\frac{\Delta\sigma}{\Delta x}\right)_{i,n-1}\right]\frac{x_n - x_{n-1}}{x_{n+1} - x_{n-1}}$ $\quad 2 \le n \le N-1$

DSX $\left(\frac{\delta\sigma}{\delta x}\right)_{oi,n} = \left(\frac{\Delta\sigma}{\Delta x}\right)_{i,n-1}$

RD $1/h(i,n) = \frac{\varkappa(i+1,n) - \varkappa(i,n)}{l(i,n)}$

TX(IN) $$\left(\frac{\partial\varphi}{\partial x}\right)_{i,n} = 2\sum_i^{iL}\left\{\left[\left(\frac{\delta\sigma}{\delta x}\right)_{oi,n} + \left(\alpha\sin\theta(i,n) + \gamma\cos\theta(i,n)\right)\left(\frac{d\sigma}{ds}\right)_{i,n} + \frac{\sigma(i,n)}{h(i,n)}\left(\frac{\delta\nu}{\delta x}\right)_{i,n}\right]\left\{\frac{\Delta\varphi(i,j,n)}{\sigma(i,n)}\right\} - \sigma(i,n)\left(\frac{\delta\nu}{\delta x}\right)_{i,n}\delta(i,j,n)\right\}$$

4) Computation of $q^{6}(j,n)$

Q2(J, N) $$(j,n) = \left(\frac{\partial\nu}{\delta x}\right)^2_{j,n} + \left\{2\sum_i \sigma(i,n)\left[\cos\left(\theta(j,n) - \theta(i,n)\right) \cdot \ln\frac{R(i+1,j,n)}{R(i,j,n)} - \delta(i,j,n)\sin\left(\theta(j,n) - \theta(i,n)\right)\right]\right\}^2$$

5) <u>Computation of Cross-sectional Properties</u>

YZP $\qquad r^2_{i,n} = (y'_{i,n})^2 + (z'_{i,n})^2$

ZZ $\qquad dz_{i,n} = z_{i+1,n} - z_{i,n}$

YY $\qquad dy_{i,n} = y_{i+1,n} - y_{i,n}$

S(N) $\qquad S(x_n) = \sum_i (y'_{i,n}\, dz_{i,n} - z'_{i,n}\, dy_{i,n})/2$

SYG(N) $\qquad y_{go} S(x_n) = \frac{1}{2}\sum_i r^2_{i,n}\, dz_{i,n}$

SZA(N) $\qquad z_{go} S(x_n) = -\frac{1}{2}\sum_i r^2_{i,n}\, dy_{i,n}$

DSYG $\qquad DYS_n = \dfrac{y_{go}S(x_{n+1}) - y_{go}\,S(x_n)}{x_{n+1} - x_n}, \qquad 1 \le n \le N-1$

SYP $\qquad \left(y_g S(x_n)\right)' = DYS_{n-1} + [DYS_n - DYS_{n-1}]\dfrac{(x_n - x_{n-1})}{x_{n+1} - x_{n-1}}$

$2 \le n \le N-1$

SYP $\qquad \left(y_g S(x_1)\right)' = 2\ DYS_1 - \left(y_g S(2)\right)'$

SYP $\qquad \left(y_g S(x_N)\right)' = 2\ DYS_{N-1} - \left(y_g S(N-1)\right)'$

repeat for $\left(z_g S(x_n)\right)$

SPXP $\qquad S'(x'_m) = \dfrac{S(x_{m+1}) - S(x_m)}{x_{m+1} - x_m}, \qquad 1 \le m \le N-1$

XP(J) $\qquad x'_m = (x_{m+1} + x_m)/2, \qquad 1 \le m \le N-1$

SPPXPP(J) $\qquad S''(x''_1) = \dfrac{S'(x'_1) - S(x_1)/x_1}{x'_1 - x_1/2}$

SPPXPP $\qquad S''(x''_m) = \dfrac{S'(x'_m) - S'(x'_{m-1})}{x'_m - x'_{m-1}}, \qquad 2 < m < N-1$

XPP(J) $\qquad x''_m = (x'_{m-1} + x'_m) ,\qquad 2 < m < N-1$

XPP(1) $\qquad x''_1 = (x_1 + x_2/2)/2$

SPPX(J) $\qquad S''_m = S''(x''_m) + [S''(x''_{m+1}) - S''(x''_m)] \dfrac{(x_m - x''_m)}{x''_{m+1} - x''_m}$

$$1 \le m < N-2$$

SPPX(NM) $\qquad S''_{N-1} = S''(x''_{N-1}) + [S''(x''_{N-1}) - S''(x''_{N-2})] \dfrac{(x_{N-1} - x''_{N-1})}{x''_{N-1} - x''_{N-2}}$

SPPX(NL) $\qquad S''_N = S''_{N-1} + [S''_{N-1} - S''_{N-2}] \dfrac{(x_N - x_{N-1})}{x_{N-1} - x_{N-2}}$

SPX $\qquad S'(1) = S'(x'_{N-1}) + [S'(x'_{N-1}) - S'(x'_{N-2})] \dfrac{(x_N - x'_{N-1})}{x'_{N-1} - x'_{N-2}}$

6) Computation of $g'(x)$, C_p, ($S'(0)$ assumed $= 0$)

RIN $\qquad I_n = \sum_{m=n}^{N-1} (S''_{m+1} - S''_m) \ln(x'_m - x_n)$

RJN $\qquad J_n = \sum_{m=0}^{n-1} (S''_{m+1} - S''_m) \ln(x_n - x'_m)$

GP $\qquad g'(x_n) = \dfrac{1}{4\pi} \Big\{ S''(x_n) \ln \dfrac{(1-M^2)}{4} + I_n - J_n + \dfrac{S'(1)}{1 - x_n}$

$$- S''(0) \ln x_n - S''(1) \ln (1 - x_n) \Big\}$$

$$1 \le n \le N-1$$

W1(J) $\qquad C_p(j,n) = -2 \left(\dfrac{\partial\varphi}{\partial x}\right)_{i,n} - q^2(j,n) - 2g'(x_n)$

7) Computation of Force and Moment Coefficients

AR(N) $$AR_{10}(x_n) = 2\sum_i \sigma(i,n)\, l(i,n)\, y'_{i,n}$$

AI(N) $$AI_{10}(x_n) = 2\sum_i \sigma(i,n)\, l(i,n)\, z'_{i,n}$$

W3 $$F_y/\rho U^2 = 2\pi AR_{10}(x_n) - \gamma\, S(x_n) + (y_{go}\, S(x_n))'$$

W2 $$F_z/\rho U^2 = 2\pi AI_{10}(x_n) - \alpha\, S(x_n) + \left(z_{go}\, S(x_n)\right)'$$

W2 $$C_L = 2(F_z/\rho U^2)(L^2/S_{ref})$$

W3 $$C_y = 2(F_y/\rho U^2)(L^2/S_{ref})$$

SUM
$$\int_o^x [2\pi AR_{10}(x) - \gamma\, S(x)]dx = 2\pi AR_{10}(x_1)\, x_1/2$$
$$+ 2\pi \sum_{m=1}^{n-1} \left\{ \left[AR_{10}(x_{m+1}) - \gamma\, S(x_{m+1})/2\pi\right] + \left[AR_{10}(x_m) - \gamma\, S(x_m)/2\pi\right] \right\} \frac{(x_{m+1} - x_n)}{2}$$

SUM1
$$\int_o^x [2\pi AI_{10}(x) - \alpha\, S(x)]dx = \left[2\pi AI_{10}(x_1)\, x_1/2 - x_1 \alpha S(x_1)/2\right]$$
$$+ 2\pi \sum_{m=1}^{n-1} \left\{ \left[AI_{10}(x_{m+1}) - \alpha\, S(x_{m+1})/2\pi\right] + \left[AI_{10}(x_m) - \alpha\, S(x_m)/2\pi\right] \right\} \frac{(x_{m+1} - x_m)}{2}$$

W5 $$M_y/\rho U^2 = -x(F_z/\rho U^2) + \int_0^x [2\pi A I_{10}(x) - \alpha S(x)]dx + z_{go} S(x_n)$$

W4 $$M_z/\rho U^2 = x(F_y/\rho U^2 - \int_0^x [2\pi A R_{10}(x) - \beta S(x)]dx - y_{go} S(x_n)$$

W5 $$C_M = 2(M_y/\rho U^2)(L^3/L_{ref} S_{ref})$$

W4 $$C_N = -2(M_z/\rho U^2)(L^3/L_{ref} S_{ref})$$

D. Program Listing

```
0001          DIMENSION Y(30,16),Z(30,16),DN(30,16),FL(30,16),YP(30),ZP(30),
             1PS(30,30),RS(30,30),DS(30,30),AJT(900),SI(30),YPP(30),
             2ZPP(30),R(30,30),D(30,30,16),P(30,30),DT(30,30,16),ST(30),X(16),
             3PH(30),T(30,16),SIG(30,16),GSUM(30,30),CT(30),
             4TX(30,16),DW(30),AA(16),AI(16),DAX(30),HE(30,16),S(16),
             5SYG(16),SZG(16),DZ(30,16)
0002          DIMENSION XP(16),XPP(16),SPPXPP(16),SPPX(16)
0003          DIMENSION L(30),H(30),DNS(30),XI(30),XIN(30)
0004          DIMENSION DPY(30),DY(30),DPZ(30),DZ(30)
0005          DIMENSION OUT(1)
0006          EQUIVALENCE(AJT,OUT)
0007          EQUIVALENCE (P,PS),(R,RS),(DS,D(1,1,16)),(GSUM,DT(1,1,16)),
             1(DY,CT),(DZ,ST),(DW,SI),(W,SIG(1,16)),(PH,H),L),(DPY,GSUM(1,1)),
             2(SPPX,HE(1,1)),(SPPXPP,HE(1,2)),(XP,HE(1,3)),(XPP,HE(1,4)),
             3(DNS,DZ(1,16)),(DPZ,GSUM(1,2)),(DNX,XIN)
      C ***     IF IYPP=0 THEN YPP WILL BE SET = TO YP ZPP=ZP R=RS D=DS AND P=P
      C ***     IF IYPP=1 THEN YPP AND ZPP MUST BE INPUT
      C ***  IF IW=0 THEN CODE TO COMPUTE Y,Z MUST BE INSERTED AFTER
      C ***  STATEMENT 111
      C ***  IF IW=1 THEN Y,Z MUST BE INPUT
      C ***  IF IX=0 THEN CODE TO COMPUTE X MUST BE INSERTED AFTER
      C *** STATEMENT 113
      C ***  IF IX=1 THEN X MUST BE INPUT
      C *** IF ISYMLR=0 THERE IS NO LEFT TO RIGHT SYMMETRY
      C *** IF ISYMLR=1 THERE IS LEFT TO RIGHT SYMMETRY
      C *** IFISYMUD=0 THERE IS NO UP TO DOWN SYMMETRY
      C *** IF ISYMUD=1 THERE IS UP TO DOWN SYMMETRY
      C *** IF ISR=0 SREF= INPUT VALUE
      C *** IF ISR NOT=0 SREF WILL BE REDEFINED =S(ISR)
0008          READ(5,501) ALPHA,BETA,ACH,SPP0,SREF,FNG,REFL
0009          WRITE(6,599) ALPHA,BETA,ACH,SPP0,SREF,FNG,REFL
0010      599 FORMAT(7H ALPHA=,F6.3,5X,5HBETA=,F6.3,5X,9HMACH NO.=,F6.2,/
             17HSPP(0)=,F10.4,5X,9HREF AREA=,F6.3,5X,12HBODY LENGTH=,F6.3,5X,
             211HREF LENGTH=,F6.3,//)
0011          READ(5,500) IYPP,IL,NL,IW,IX,ISYMLR,ISYMUD,ISR
0012          PI=4.*ATAN(1.0)
0013          PI2=2.*PI
0014          PIP=PI+1.E-5
0015          ILP=IL+1
0016          NLM=NL-1
0017          PI4=4.*PI
0018          IF(ISYMLR.EQ.0.AND.ISYMUD.EQ.0) ILL=IL
0019          IF(ISYMLR.EQ.0.AND.ISYMUD.EQ.1) ILL=IL/2
0020          IF(ISYMLR.EQ.1.AND.ISYMUD.EQ.0) ILL=IL/2
0021          IF(ISYMLR.EQ.1.AND.ISYMUD.EQ.1) ILL=IL/4
0022          IF(IX.EQ.0) GO TO 113
0023          READ(5,505) (X(N),N=1,NL)
0024          GO TO 114
0025      113 CONTINUE
      C ***  IF X WAS NOT INPUT THEN CODE TO COMPUTE MUST BE
      C *** INSERTED HERE
0026      114 IF(IW.EQ.0) GO TO 111
0027          DO 105 N=1,NL
0028      105 READ(5,505)(Y(I,N),I=1,ILL)
0029          DO 106 N=1,NL
0030      106 READ(5,505)(Z(I,N),I=1,ILL)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  505 FORMAT(10F8.0)
      GO TO 112
  111 CONTINUE
C *** IF Y AND Z WERE NOT INPUT THEN CODE TO COMPUTE MUST BE
C *** INSERTED HERE
  112 DO 597 N=1,NL
      X(N)=X(N)/ENG
      DO 597 I=1,ILL
      Y(I,N)=Y(I,N)/ENG
  597 Z(I,N)=Z(I,N)/ENG
      IF(ILL.EQ.IL) GO TO 461
      IF(ILL.NE.IL/2) GO TO 455
      IB=ILL+1
      ILL=IL/2
      DO 450 N=1,NL
      DO 450 I=IB,ILL
      ID=ILL+1-I
      Y(I,N)=Y(ID,N)
  450 Z(I,N)=-Z(ID,N)
  455 IB=ILL+1
      FY=-1.
      FZ=1.
      IF(ISYMLR.EQ.1) GO TO 465
      FY=1.
      FZ=-1.
  465 DO 460 N=1,NL
      DO 460 I=IB,IL
      ID=IL+1-I
      Y(I,N)=Y(ID,N)*FY
  460 Z(I,N)=Z(ID,N) *FZ
  461 CONTINUE
    1 DO 125 N=1,NL
      AR(N)=0.0
      AI(N)=0.0
      S(N)=0.0
      SZS(N)=0.0
      SYS(N)=0.0
      IF(ITYPQ.EQ.1) READ(5,501) (YPP(I),ZPP(I),I=1,IL)
      Y(ILP,N)=Y(1,N)
      Z(ILP,N)=Z(1,N)
      IF(N.NE.1) GO TO 3
      N1=2
      N2=1
      GO TO 5
    3 IF(N.NE.NL) GO TO 4
      N1=N
      N2=N-1
      GO TO 5
    4 N1=N+1
      N2=N-1
    5 DO 11 I=1,IL
      YY=Y(I+1,N)-Y(I,N)
      ZZ=Z(I+1,N)-Z(I,N)
   11 EL(I,N)=SQRT(YY*YY+ZZ*ZZ)
      IL2=IL+2
      IL3=IL+3
```

```
      IL4=IL+4
      Y(IL2,N)=Y(2,N)
       Y(IL3,N)=Y(3,N)
       Y(IL4,N)=Y(4,N)
      Z(IL2,N)=Z(2,N)
       Z(IL3,N)=Z(3,N)
       Z(IL4,N)=Z(4,N)
      FL(ILP,N)=FL(1,N)
       FL(IL2,N)=FL(2,N)
       FL(IL3,N)=FL(3,N)
      IM1=1
       IM2=2
       IM3=3
       IF(ILL.EQ.IL) GO TO 480
       IL3=IL+1
      IL2=IL
       IM1=ILL-2
       IM2=ILL-1
       IM3=ILL
  480 DO 8 I=IM1,IL3
       DPZ(I)=(Z(I+1,N)-Z(I,N))/FL(I,N)
    8 DPY(I)=(Y(I+1,N)-Y(I,N))/FL(I,N)
       DO 9 I=IM2,IL3
      DZ(I)=(DPZ(I-1)*FL(I,N)+DPZ(I)*FL(I-1,N))/(FL(I,N)+FL(I-1,N))
    9 DY(I)=(DPY(I-1)*FL(I,N)+DPY(I)*FL(I-1,N))/(FL(I,N)+FL(I-1,N))
      DO 13 I=IM2,IL2
       DPZ(I)=(DZ(I+1)-DZ(I))/FL(I,N)
   13 DPY(I)=(DY(I+1)-DY(I))/FL(I,N)
       DO 18 I=IM3,IL2
      ZP(I)=(DPZ(I-1)*FL(I,N)+DPZ(I)*FL(I-1,N))/(FL(I,N)+FL(I-1,N))
   16 YP(I)=(DPY(I-1)*FL(I,N)+DPY(I)*FL(I-1,N))/(FL(I,N)+FL(I-1,N))
      ZP(I)=Z(I,N)+DZ(I)*.5*FL(I,N)+.5*ZP  (I)*(.5*FL(I,N))**2
   18 YP(I)=Y(I,N)+DY(I)*.5*FL(I,N)+.5*YP  (I)*(.5*FL(I,N))**2
      IF(ILL.EQ.IL) GO TO 492
       ILLM=ILL-1
       IF(ISYMLR.EQ.1) GO TO 486
      ZP(ILL)=0.0
       ZP(IL)=0.0
       DO 484 I=1,ILLM
       ZP(I)=-ZP(IL-I)
  484 YP(I)=YP(IL-I)
       GO TO 494
  486 YP(ILL)=0.0
       YP(IL)=0.0
       DO 488 I=1,ILLM
       YP(I)=-YP(IL-I)
  488 ZP(I)=ZP(IL-I)
       GO TO 494
  492 CONTINUE
      YP(1)=YP(ILP)
       YP(2)=YP(IL2)
       ZP(1)=ZP(ILP)
       ZP(2)=ZP(IL2)
  494 CONTINUE
      DO 10 I=1,IL
       YY=Y(I+1,N)-Y(I,N)
```

```
      ZZ=Z(I+1,N)-Z(I,N)
      ST(I)=ZZ/FL(I,N)
       CT(I)=YY/FL(I,N)
      S(N)=YP(I)*ZZ-ZP(I)*YY+S(N)
       YZP=YP(I)**2+ZP(I)**2
      SYG(N)=SYG(N)+YZP*ZZ
       SZG(N)=SZG(N)+YZP*YY
      AS=ARSIN(ST(I))
      IF(CT(I).GE.0.0) GO TO 7
      T(I,N)=PI-AS
      GO TO 10
7     T(I,N)=AS
       IF(ST(I).LT.0.0) T(I,N)=PI2+AS
10    CONTINUE
       YP(ILP)=YP(1)
       ZP(ILP)=ZP(1)
       T(ILP,N)=T(1,N)
      DO 6 I=1,IL
       IP=I+1
      TPT=T(IP,N)-T(I,N)
       IF(TPT.GT.PIP) T(IP,N)=T(IP,N)-PI2
      IF(TPT.LT.(-PIP)) T(IP,N)=T(IP,N)+PI2
6     HE(IP,N)=T(I,N)+(T(IP,N)-T(I,N))*(FL(I,N)/(FL(IP,N)+FL(I,N)))
      HE(1,N)=HE(ILP,N)-PI2
       DO 2 I=1,IL
2     DR(I)=(Y(I,N))-Y(I,N))*SIN(HE(I,N))-(Z(I,N)-Z(I,N))*COS(HE(I,N))
      S(N)=.5*S(N)
       SYG(N)=.5*SYG(N)
       SZG(N)=-.5*SZG(N)
      DR(ILP)=DR(1)
       XX=1./(X(N1)-X(N2))
      IF(N.NE.1) XXP=X(N)-X(N-1)
       IF(N.NE.NL) XXM=1./(X(N+1)-X(N))
      DO 20 I=1,IL
       DV=.5*(DR(I)+DR(I+1))
      IF(N.EQ.NL) GO TO 12
       DNAS=DNX(I)
       DNX(I)=DV*XXM
      IF(N.EQ.1) GO TO 12
       DN(I,N)=DNAS+(DNX(I)-DNAS)*XX*XXP
       GO TO 14
12    DN(I,N)=DNX(I)
14    DN(I,N)=DN(I,N)+ALP*CT(I)-BET*ST(I)
      DNS(I)=DN(I,N)*.5
      DO 20 J=1,IL
       YY=YP(J)-Y(I,N)
       ZZ=ZP(J)-Z(I,N)
      RS(J,I)=SQRT(YY*YY+ZZ*ZZ)
       AS=ARSIN(ZZ/RS(J,I))
      IF(YY.GE.0.0) GO TO 15
       G=PI-AS
       GO TO 17
15    G=AS
       IF(ZZ.LT.0.0) G=PI2+AS
17    RS(J,I)=G
       IF(G.LT.T(I,N))RS(J,I)=G+PI2
```

```
20    DS(J,I)=G
       KS=-IL
      DO 30 J=1,IL
       DS(J,ILP)=DS(J,1)
       RS(J,ILP)=RS(J,1)
      KS=KS+1
       K=KS
      DO 30 I=1,IL
       STT=ST(J)*CT(I)-CT(J)*ST(I)
       K=K+IL
      G=DS(J,I+1)
       IF(G.GT.T(I,N)) PHS=G
       IF(G.EQ.T(I,N))PHS=PS(J,I+1)
      IF(G.LT.T(I,N)) PHS=G+PI2
      CTT=CT(J)*CT(I)+ST(J)*ST(I)
       DS(J,I)=PHS-PS(J,I)
      IF(J.EQ.I) DS(J,I)=-PI
      RRL=ALOG(RS(J,I+1)/RS(J,I))
       GSUM(J,I)=CTT*RRL-DS(J,I)*STT
30    AJI(K)=RRL*STT+DS(J,I)*CTT
      CALL MINV(AJI,IL,DD,L,M)
      CALL GMPRD(AJI,DN,SI,IL,IL,1)
      WRITE(6,705) N
      WRITE(6,700)
      WRITE(6,503) (SI(I),I=1,IL)
   50 DO 57 J=1,IL
       Q2(J,N)=0.0
       SIG(J,N)=SI(J)
      DO 55 I=1,IL
55    Q2(J,N)=Q2(J,N)+SI(I)*GSUM(J,I)
57    Q2(J,N)=DN(J,N)**2+(2.*Q2(J,N))**2
      IF(IYPP.EQ.1) GO TO 70
       DO 60 I=1,IL
       YPP(I)=YP(I)
      ZPP(I)=ZP(I)
       DO 60 J=1,IL
   60 P(J,I,N)=DS(J,I)
70    YPP(ILP)=YPP(1)
       ZPP(ILP)=ZPP(1)
       IF(IYPP.EQ.0) GO TO 100
      DO 80 I=1,IL
       DO 80 J=1,IL
       ZZ=ZPP(J)-Z(I,N)
      YY=YPP(J)-Y(I,N)
       R(J,I)=SQRT(YY*YY+ZZ*ZZ)
       AS=ARSIN(ZZ/R(J,I))
      IF(YY.GE.0.0) GO TO 75
       G=PI-AS
       GO TO 77
75    G=AS
       IF(ZZ.LT.0.0) G=PI2+AS
77    R(J,I)=G
       IF(G.LE.T(I,N)) R(J,I)=G+PI2
80    G(J,I,N)=G
       DO 90 J=1,IL
       D(J,ILP,N)=D(J,1,N)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 90 I=1,IL
      G=D(J,I+1,N)
      IF(G.GT.T(I,N)) PHS=G
      IF(G.EQ.T(I,N)) PHS=P(J,I+1)
      IF(G.LT.T(I,N)) PHS=G+P12
90    D(J,I,N)=PHS-P(J,I)
100   DO 120 J=1,IL
      R(J,IL+1)=R(J,1)
      PH(J)=0.0
      SF=SI(J)*FL(J,N)
      AR(N)=AR(N)+YP(J)*SF
      AI(N)=AI(N)+ZP(J)*SF
      TX(J,N)=0.0
      DO 110 I=1,IL
      IP=I+1
      YY=YPP(J)-Y(I,N)
      ZZ=ZPP(J)-Z(I,N)
      UT=YY*CT(I)+ZZ*ST(I)
      RU=(YPP(J)-Y(IP,N))*CT(I)+(ZPP(J)-Z(IP,N))*ST(I)
      RN=-YY*ST(I)+ZZ*CT(I)
      DT(J,I,N)=RU*ALOG(R(J,IP))-RT*ALOG(R(J,I))-RN*D(J,I,N)+FL(I,N)
110   PH(J)=PH(J)+SIG(I,N)*DT(J,I,N)
120   PH(J)=2.*PH(J)
      WRITE(6,710)
      WRITE(6,503) (PH(J),J=1,IL)
      WRITE(6,725)
      I=0
      DO 122 II=1,IL
      I=I+1
      OUT(I)=YP(II)*ENG
      I=I+1
  122 OUT(I)=ZP(II)*ENG
      ILT=2*IL-1
      L1=1
      L9=13
  123 L2=L1+1
      L8=L9+1
      WRITE(6,503) (OUT(I),I=L1,L9,2)
      WRITE(6,503) (OUT(I),I=L2,L8,2)
      WRITE(6,730)
      L1=L1+14
      L9=L9+14
      IF(L1.GT.ILT) GO TO 124
      IF(L9.LT.ILT) GO TO 123
      L9=ILT
      GO TO 123
  124 CONTINUE
  725 FORMAT(1X,[illegible])
  730 FORMAT(/)
      AR(N)=2.*AR(N)
      AI(N)=2.*AI(N)
      SIG(IL+1,N)=SIG(1,N)
125   CONTINUE
      IF(ISW.NE.0) SREF=S(ISO)*ENG**2
      DO 1[illegible] N=1,NI
      NP=N+1
```

```
      NM=N-1
      IF(N.EQ.1) GO TO 181
      IF(N.EQ.NL) GO TO 182
      XX=1./(X(NP)-X(NM))
  181 XXP=1./(X(NP)-X(N))
      IF(N.EQ.1) GO TO 183
  182 XXM=X(N)-X(NM)
  183 SII=SIG(IL,N)+(SIG(1,N)-SIG(IL,N))*FL(IL,N)/(FL(IL,N)+FL(1,N))
      DO 190 I=1,IL
      IP=I+1
      FFF=FL(I,N)/(FL(IP,N)+FL(I,N))
      SIIP=SIG(I,N)+(SIG(IP,N)-SIG(I,N))*FFF
      IF(N.EQ.NL) GO TO 187
      XINO=XIN(I)
      TMT=T(I,NP)-T(I,N)
      IF(TMT.GT.PI) TMT=TMT-PI2
      IF(TMT.LT.(-PI)) TMT=TMT+PI2
      XIN(I)=(SIG(I,NP)-SIG(I,N))*XXM
      IF(N.NE.1) GO TO 186
      DSX=XIN(I)-((SIG(I,3)-SIG(I,2))/(X(3)-X(2))-XIN(I))/(X(3)-X(1))
     1/XXM
      GO TO 188
  186 CONTINUE
      DSX=XINO+(XIN(I)-XINO)*XX*XXP
      GO TO 188
  187 XIN(I)=(SIG(I,NM)-SIG(I,N-2))/(X(NM)-X(N-2))
      DSX=2.*XIN(I)-XINO-(XIN(I)-XINO)/(X(N)-X(N-2))*(X(NM)-X(N-2))
  188 WO=(OE(IP,N)-OE(I,N))/FL(I,N)
      SS=(SIIP-SII)/FL(I,N)
      SII=SIIP
      ONE=(DSX+SS*(ALP*ST(I)+BET*CT(I))+SIG(I,N)*RD*DN(I,N))*2.
      TWO=2.*SIG(I,N)*DN(I,N)
      DO 190 J=1,IL
  190 TX(J,N)=TX(J,N)+ONE*DT(J,I,N)-TWO*D(J,I,N)
      WRITE(6,400) (S(N),SYG(N),SZG(N),N=1,NL)
      WRITE(6,715)
      DO 299 N=1,NL
  299 WRITE(6,510) (TX(J,N),J=1,IL)
  510 FORMAT(7F16.8)
      WRITE(6,720)
      WRITE(6,503)(AI(N),AR(N),N=1,NL)
      XPP(1)=.5*(X(1)  5*X(2))
      NM=NL-1
      SPXP=0.
      NM2=NM-1
      DO 403 J=1,NM
      JP=J+1
      JM=J-1
      XX=X(JP)+X(J)
      XP(J)=.5*XX
      IF(J.NE.1) XPP(J)=(XP(JM)+XP(J))*.5
      SPXPS=SPXP
      SPXP=(S(JP)-S(J))/(X(JP)-X(J))
      IF(J.NE.1) GO TO 400
      SPPXPP(1)=(SPXP-S(1)/X(1))/(XP(1)-.5*X(1))
      GO TO 403
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  300 SPPAPP(J)=(SPAP-SPAPS)/(AP(J)-XP(JM))
  303 CONTINUE
      DO 305 J=1,NM2
      JP=J+1
  305 SPPX(J)=SPPAPP(J)+(SPPAPP(JP)-SPPAPP(J))/(XPP(JP)-XPP(J))*(X(J)-
     1XPP(J))
      SPPX(NM)=SPPAPP(NM)+(SPPAPP(NM)-SPPAPP(NM2))/(XPP(NM)-XPP(NM2))*
     1(X(NM)-XPP(NM))
      SPPX(NL)=SPPX(NM)+(SPPX(NM)-SPPX(NM2))/(X(NM)-
     2X(NM2))*(X(NL)-X(NM))
      SPX=SPXP+(SPAP-SPAPS)/(XP(NM)-XP(NM2))*(X(NL)-XP(NM))
      DSYG=0.0
      DSZG=0.0
      DO 400 N=1,NL
      IF(N.NE.NL) GO TO 310
      SYP=2.*DSYG-SYP
      SZP=2.*DSZG-SZP
      GO TO 330
  310 NP=N+1
      NM=N-1
      XX=X(NP)-X(N)
      SSYG=DSYG
      SSZG=DSZG
      DSYG=(SYG(NP)-SYG(N))/XX
      DSZG=(SZG(NP)-SZG(N))/XX
      IF(N.NE.1) GO TO 320
      XXM=X(3)-X(2)
      XXP=XX/(X(3)-X(1))
      SYP=DSYG-((SYG(3)-SYG(2))/XXM-DSYG)*XXP
      SZP=DSZG-((SZG(3)-SZG(2))/XXM-DSZG)*XXP
      GO TO 330
  320 XXP=(X(N)-X(NM))/(X(NP)-X(NM))
      SYP=SSYG+(DSYG-SSYG)*XXP
      SZP=SSZG+(DSZG-SSZG)*XXP
  330 WAN=(AP(1)-BETAS(1)/P(2)*X(1))*.5
      XX=ALOG(.5*SQRT(1.-XCH**2))
      WJM=(SPPX(1)-SPPO)*ALOG(X(N)-.5*X(1))
      WIN=0.0
      DO 337 J=1,NLM
      JP=J+1
      IF(J.GE.N) GO TO 335
      WJN=WJN+(SPPX(JP)-SPPX(J))*ALOG(X(N)-XP(J))
      GO TO 337
  335 WIN=WIN+(SPPX(JP)-SPPX(J))*ALOG(XP(J)-X(N))
  337 CONTINUE
      IF(N.NE.NL) WN=(2.*(SPPX(N)*XX+.5*(WIN-.5*WJN)-SPPX(NL)*
     1ALOG(1.-X(N))-SPPO*ALOG(X(N))+S-S/(1.-X(N))))/P1+
      WAT=(AT(1)-BETAS(1)/P(2)*X(1))*.5
      IF(N.EQ.1) GO TO 450
      NL=N-1
      SUM=0.0
      SUM1=0.0
      DO 460 NS=1,ML
      NP=NS+1
      XX=.5*(X(NP)-X(NS))
      SUM=SUM+(AN(NP)-BETAS(NP)/P(2)+AN(NS)-BETAS(NS)/P(2))*XX
```

```
 380 SUM1=SUM1+(A1(M)*ALP**(M-1)/PI2*A1(M)*ALP**(M-1)/PI2)*S
     WAR=WAR+SUM
      WAI=WAI+SUM1
 390 W2=PI2*AI(N)*S/P-ALP*S*WR
      W3=PI2*AR(N)+SYR-WFI*S(N)
     W4=W3*R(N)+SYG(N)-PI2*WAR
     F1=2.*ENG*W2/SWEF
     F2=F1*ENG/WEFL
     W3=F1*W3
     W4=-F2*W4
     W5=F2*(W2*R(N)-S G(N)*PI2*WAI)
     W2=F1*W2
     WRITE(6,740)N,W3,W2,W4,W5,GP
     IF(N.EQ.NL) STOP
      DO 380 J=1,IL
 380 Q1(J)=2.*(-TR(J,N)*GP+.5*Q2(J,N))
     WRITE(6,750)
 400 WRITE(6,503) (Q1(J),J=1,IL)
 500 FORMAT(10I5)
 501 FORMAT(5E15.8)
 503 FORMAT(7E16.8)
 700 FORMAT(1X,5HSIGMA)
 705 FORMAT(1X,I2)
 710 FORMAT(1X,3HPHI)
 715 FORMAT(1X,5HD PHI/D X)
 720 FORMAT(1X,6HAI AND AR)
 740 FORMAT(/,3H N=,I2,5X,3HCY=,E15.7,5X,3HCL=,E15.7,5X,3HCM=,E15.7,5X,
    13HCX=,E15.7,5X,3HGP=,E15.7)
 750 FORMAT(5X,2HCP)
 600 FORMAT(/,5X,3HSY=,E15.8,5X,5HYG=,E15.8,5X,5HYGS=,E15.8)
     END
```

```
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      D=1.0
       NK=-N
       DO 80 K=1,N
       NK=NK+N
       L(K)=K
       M(K)=K
       KK=NK+K
      BIGA=A(KK)
       DO 20 J=K,N
       IZ=N*(J-1)
       DO 20 I=K,N
       IJ=IZ+I
      IF(ABS(BIGA)-ABS(A(IJ)).GT.0.)GOTO20
       BIGA=A(IJ)
       L(K)=I
       M(K)=J
   20 CONTINUE
       J=L(K)
       IF(J-K.LE.0)GO TO 35
       KI=K-N
      DO 30 I=1,N
       KI=KI+N
       HOLD=-A(KI)
       JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI)=HOLD
   35 I=M(K)
       IF(I-K.LE.0) GO TO 45
       JP=N*(I-1)
       DO 40 J=1,N
      JK=NK+J
       JI=JP+J
       HOLD=-A(JK)
       A(JK)=A(JI)
   40 A(JI)=HOLD
   45 IF(BIGA.NE.0.0) GO TO 48
       D=0.0
       RETURN
   48 DO 55 I=1,N
       IF(I-K.EQ.0) GO TO 55
       IK=NK+I
       A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
       DO 65 I=1,N
       IK=NK+I
       HOLD=A(IK)
       IJ=I-N
      DO 65 J=1,N
       IJ=IJ+N
       IF(I-K.EQ.0)GO TO 65
      IF(J-K.EQ.0)GO TO 65
      KJ=IJ-I+K
       A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
```

```
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K.EQ.0) GO TO 75
      A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
      A(KK)=1.0/BIGA
   80 CONTINUE
      K=N
  100 K=K-1
      IF(K.LE.0) RETURN
      I=L(K)
      IF(I-K.LE.0) GO TO 120
      JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI)=HOLD
  120 J=M(K)
      IF(J-K.LE.0) GO TO 100
      KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI)=HOLD
      GO TO 100
      END
```

PART III

MODIFICATIONS FOR CROSSECTIONS WITH CORNERS

A. Discussion

Parts I and II describe a program to compute force coefficients and pressure distributions over arbitrarily but smoothly shaped crossections in the absence of corners. Although solutions based on slender body theory are invalid over regions of high surface curvature they are still capable of yielding good results away from such irregularities provided additonal care is exercised in the computation of geometric surface properties as a corner is approached. Analytically, a corner represents an arbitrary break in the structure of local surface properties. Any scheme of specifying corner properties by a finite number of discrete parameters must involve implicit assumptions regarding the behavior of such corners between points at which data is given. For this reason it is desireable to have a procedure which allows the user some discretion regarding these assumptions without requiring an excessive amount of data to define surfaces. In the following procedure this discretion is excersized in the choice of the distribution of orthogonal lines S_i introduced in Fig. 2.

In a finite computational scheme the difficulties inherent at a corner first become manifest when the local radius of curvature on C_n becomes small compared to the distance in the y, z plane to the neighboring crossection C_{n+1}. Such points are illustrated in Fig. 16 at (i, n) = (15,5), (15, 6), (15, 7). For practical computations such points are equivalent to the sharp corners of (4, 2), (4, 3) and must be treated in the same way. In contrast to the procedure of Part II which "rounds off" regions of higher curvature it is more appropriate now to adapt the opposite procedure namely: a region of finite but large curvature is to be replaced by a sharp corner. If this approximation should prove too crude it would then be necessary to include an additional contour between C_n and C_{n+1} so that the distance between contours is less than the local radius of curvature, a procedure which is equivalent to supplying more detailed data to fill in the objectionable gaps.

B. Definitions

1. Stringers S_i

The lines orthogonal to the crossectional contours C(n) and for which i = constant shall be called stringers. These are the family of lines S_i first illustrated in Fig. 2.

2. Corner lines i = IC(K)

These are lines passing thru corner points. They are to be considered as part of the family of stringers S_i. As such they are continued over the entire length of the body even though previous or subsequent crossections do not have corners. An example of one such line is shown for i = 4 in Fig. 16. Corner lines are distinguished by the index IC(K) = i signifying that the index of the K^{th} corner, counting counter clockwise, is i. Thus in Fig. 16 IC(2) = 4. (For programming convenience it is expedient to designate the first stringer i = 1 as a corner line ie (C(1) = 1 even though there may be no corners along this line.)

3. Submerged lines IBP(K, n), IBM(K, n)

A stringer S_i from the contour C_n may intersect a corner line before it intersects the next contour C_{n+1} as illustrated in Fig. 16 at (10, 6), (16, 4), (14, 5) and (17, 6). Subsequently such stringers are considered to follow the corner line and are regarded as submerged. At the K^{th} corner on C_n the highest submerged line index is denoted by IBP(K, n) and the lowest by IBM(K, n). Thus from Fig. 16 we find IB1(5, 7) = 17, IBM(5, 7) = 14. A corner line may also be counted as a submerged line ie: IBM(15, 5) = 15, IBM(15, 4) = 15. We note then, that every intersection of a corner line IC(K) with a contour C_n has associated with it the indices IBM(K, n), IBP(K, n). For purposes of illustration a complete table of IBP(K, n) is provided in Fig. 16. Finally, we note that in the absence of any corners along i = 1 we net IBM(1, n) = IL + 1. In Fig. 16 this means that IBM(1, n) = 19.

C. Modifications to the computational procedure

1. Collocation Points

Points P'(i, n) at which $\partial v/\partial x$, σ etc. are to be evaluated were previously found by smooth interpolation between P(i-2, n) and P(i+2, n). To avoid the requiring of an excessive number of data locations P(i, n) between corners this has been modified so that P'(i, n) is read directly from supplied data or by simple interpolation between neighboring locations P(i, n), P(i+1, n). In many practical applications the contour curvature between two corners is small and the later procedure should be adequate.

2. Computation of $\partial v/\partial x$

Values of $\delta v/\delta x$ are to be found at P(i, n), P(i+1, n) and interpolated to obtain a value of P'(i, n) between i and i + 1. (This represents a minor but necessary change from the procedure of Part II which determines δv at P'(i, n-1) & P'(i, n) and interpolales the associated derivatives along the x direction). The increments δv are taken along the stringers and as long as these do not intersect the corner lines the determination of $\delta v/\delta x$ at the data points P(i, n) is carried out as though no corners were present.

When a stringer S_i intersects a corner line the local corner geometry is assumed as shown in Fig. 17 which represents an enlargement of the local configuration as it appears in Fig. 16 at P(4, 2) and P(4, 3). While δv as indicated in Fig. 17 may be calculated directly from the data presented in the plots of C_n and S_i, the value of δx must be inferred from the assumption that the corner line shown in Fig. 17 is closely approximated by a straight line. Thus with $\delta\nu_1$, $\delta\nu_2$, as indicated in Fig. 17:

$$\delta x = \left[x(n+1) - x(n)\right] \frac{\delta\nu_1}{\delta\nu_1 + \delta\nu_2}$$

This is to be compared with the calculation away from a corner where $\delta(x)$ is simply $x(n+1) - x(n)$.

To devise a program which is applicable to all possible instances of corner geometry it is necessary to have tests which indicate when a stringer emerges from corner as between P(4, 2) and P(4, 3) in Fig. 16, and when it converges toward a corner to become subsequently submerges as is the case between P(11, 6) and P(11, 7). Such a test is readily constructed with the aid of the indices IBP and IBM. Thus for example:

IBP(K, n+1) < IBP(K, n) — at least one stringer has emerged between P(IC(K), n) & P(IC(K), n+1)

and

IBP(K, n) - IBP(K, n+1) = no. of emerged stringers.

In this manner IBP and IBM provide complete information regarding the emergence or convergence of stringers on either side of a corner line. This information together with implied geometry of Fig. 17 enables the computation of $\delta\nu/\delta x$ at the center of contour segments which are adjacent to corner lines.

3. Curvature

The fact that curvature is divergent near corner-like points leads to errors in the computation of $\partial\varphi/\partial x$ when using the program of Part II. This program in effect rounds off corners whereas as pointed out in the discussion above a more appropriate procedure is to treat regions of high curvature as sharp corners ie as though high curvature regions were concentrated at a corner point. With this procedure the curvature of segments adjacent to a corner is small and may be obtained by extrapolation from a neighboring segment. Thus, referring to Fig. 16 we would have:

$$h(3,3) = h(2,3)$$
$$h(4,3) = h(5,3)$$

4. Computation of $\delta\sigma/\delta x$

In Part II $\delta\sigma/\delta x$ was approximated by central divided differences involving $\sigma(i,n-1)$, $\sigma(i,n)$, $\sigma(i,n+1)$. This procedure breaks down at a corner. The rules to be followed near corners will now be that $(\delta\sigma/\delta x)_{i,n}$ will be computed by:

Forward divided difference when S_i and/or S_{i+1} emerge from a corner.

Backward divided differences when S_i and/or S_{i+1} converge to a corner.

Central divided differences away from corner.

As an illustration corresponding to Fig. 16

$$(\partial\sigma/\partial x)_{4,3} = \frac{\sigma(4,4) - \sigma(4,3)}{x(4) - x(3)}$$

$$(\partial\sigma/\partial x)_{10,6} = \frac{\sigma(10,5) - \sigma(10,6)}{x(6) - x(5)}$$

In the event of a stringer emerging just behind a segment and again converging just ahead of it $\partial\sigma/\partial x$ shall be assumed to be zero.

5. $d\sigma/ds$

To compute $d\sigma/ds$ we just find σ at all the data points $P(i,n)$ (except at a corner pt.) by interpolation between neighboring collocation points $P'(i-1,n)$, $P'(i,n)$. At corner points $d\sigma/ds$ is then found by forward differences leaving a corner along C_n and by backward differences when approaching a corner.

6. Matrix Inversion and Summation

For the matrix inversion process encountered in the evaluation of $\sigma(i, n)$ it is convenient to reorder the indices so that the actual finite segments of a contour C_n are indexed consecutively. This involves shipping over submerged segments in the counting process. Such a reordering may be accomplished through the introduction of a new index IR(m) for which the m are consecutive indices and:

$$IR(m) = i$$

for values of i corresponding to unsubmerged segments. Thus we would have, for example

$$\sum_{m=1}^{mL} \sigma\Big(IR(m), n\Big)\, a\Big(IR(m), j\, n\Big) = \sum \sigma(i, n)\ a(i, j, n)$$

where the latter summation is taken only over those values of i corresponding to segments which are not submerged.

The remaining computational procedures from Part II are not affected by the presence of corners and do not require modification.

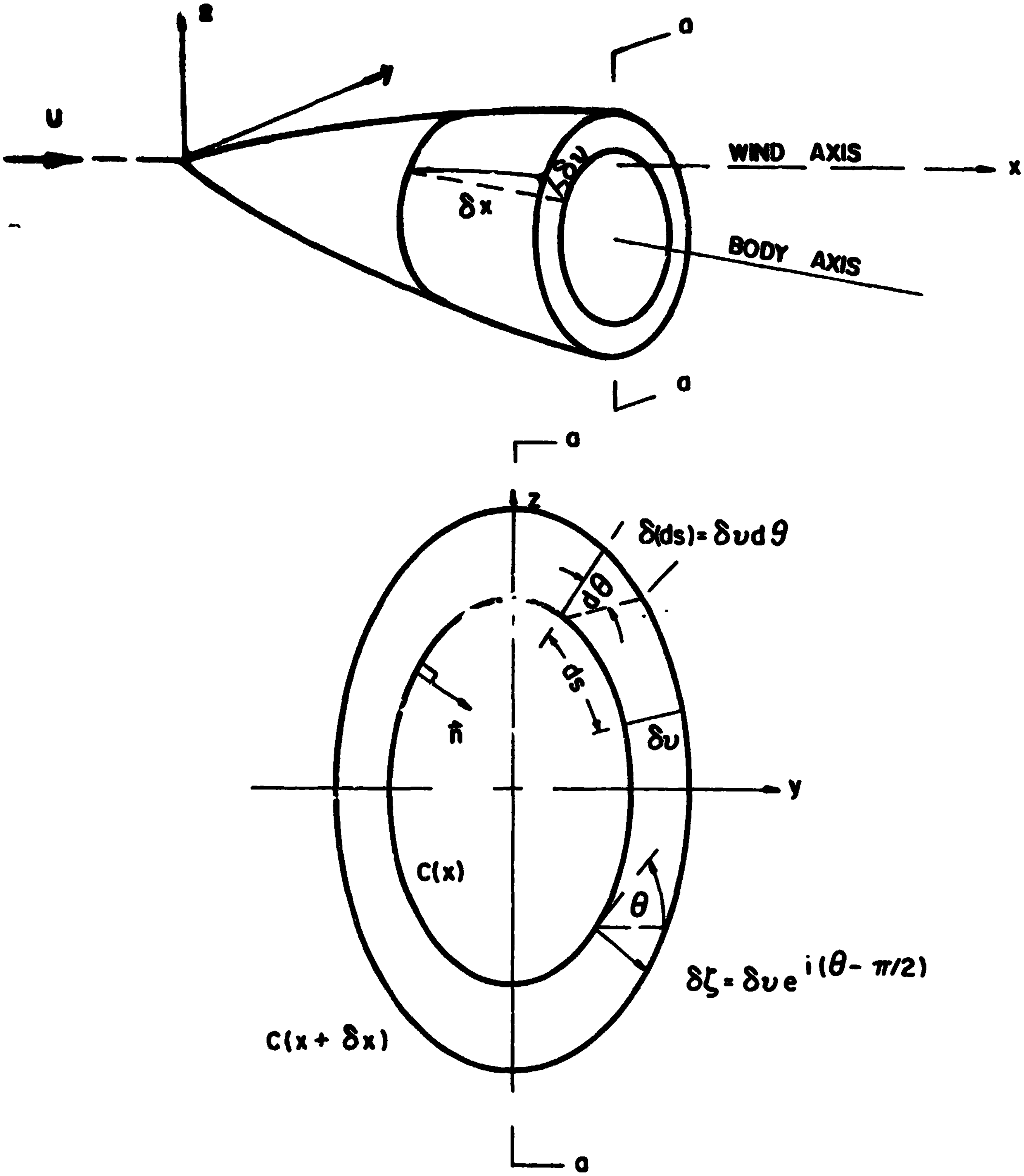


FIG. 1 BODY SLOPE AND CROSSECTIONAL VARIABLES

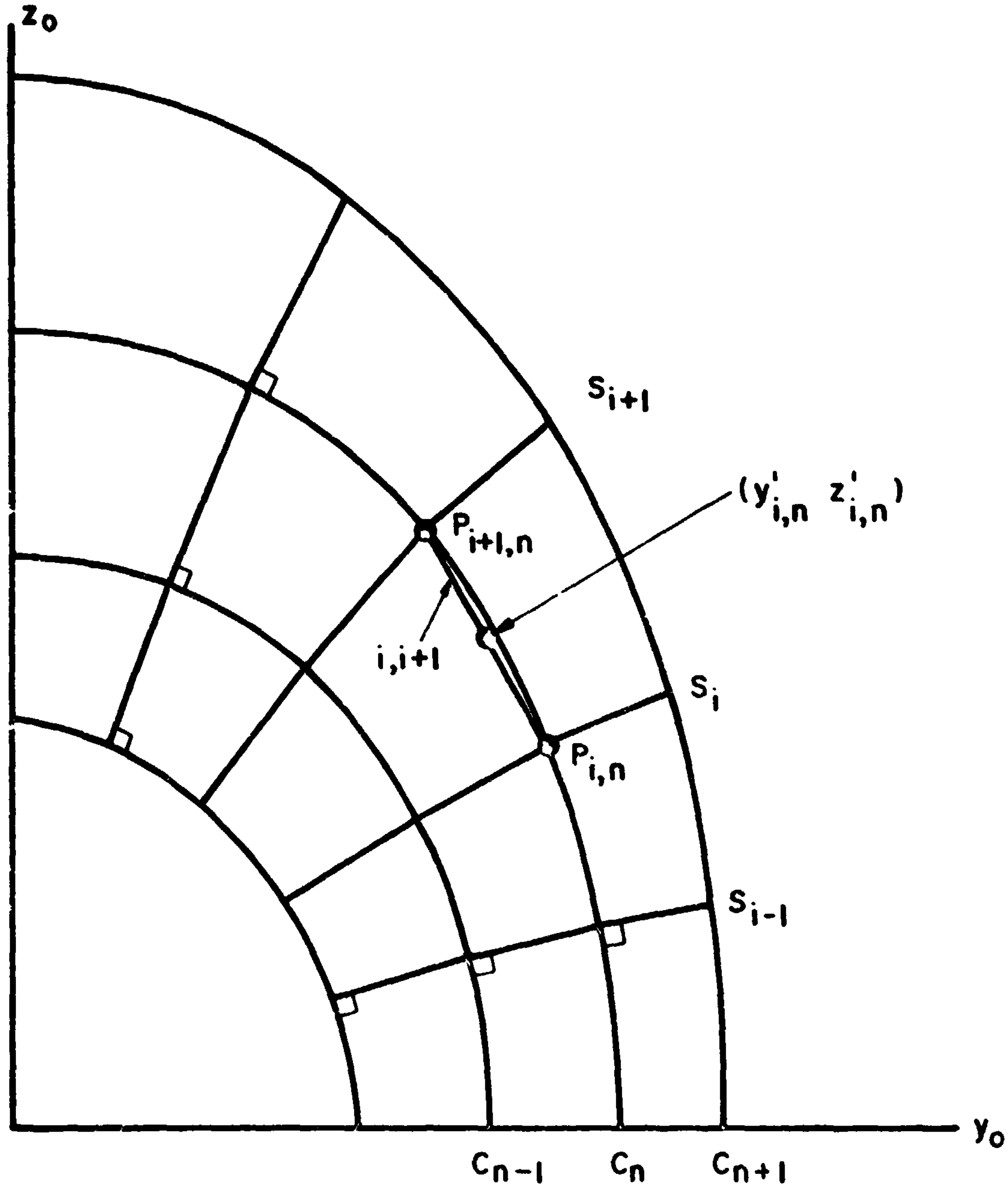


FIG. 2 CROSSECTION BOUNDARY SEGMENTING SCHEME IN BODY AXES COORDINATES (z_0, y_0)

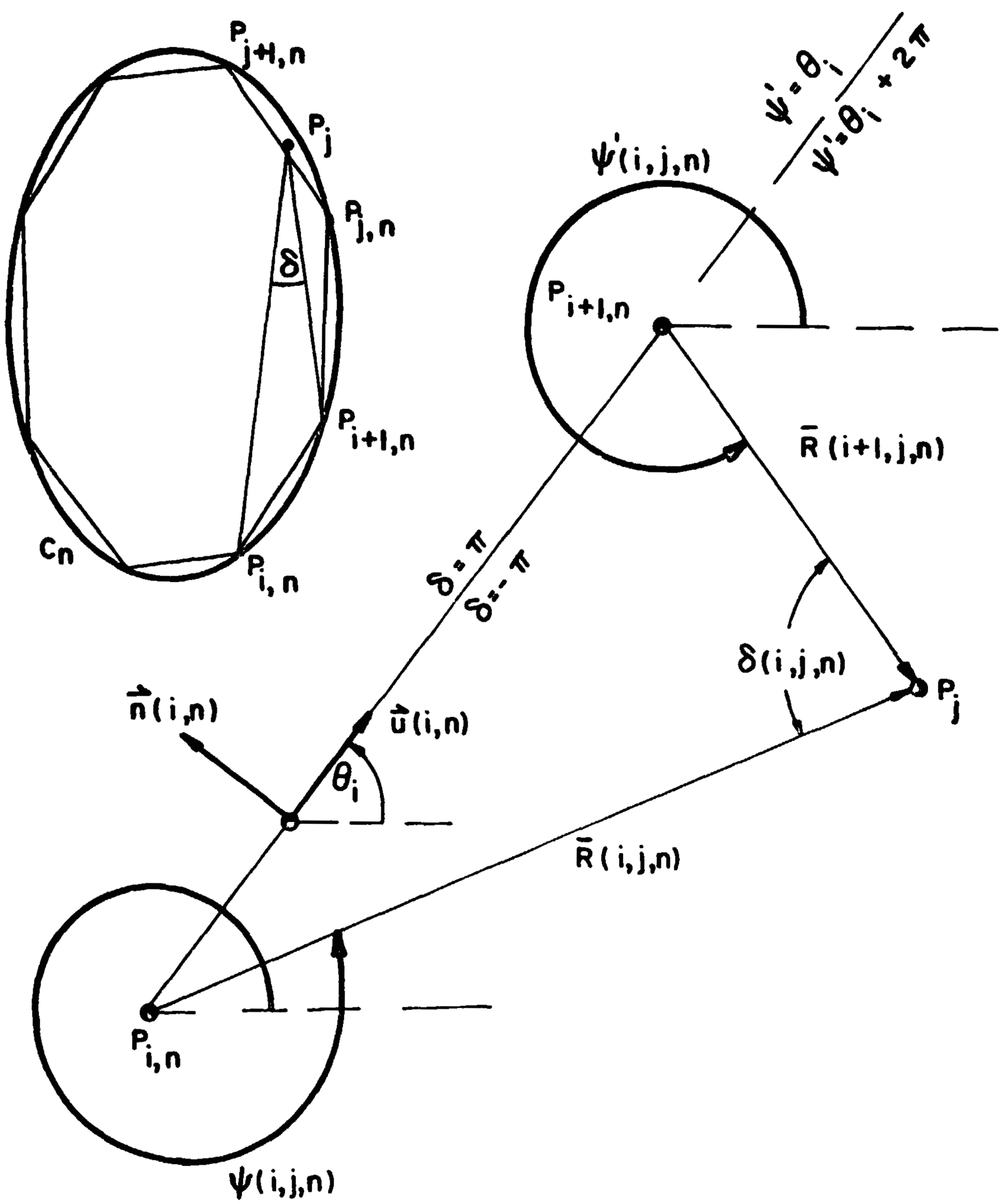


FIG. 3 DETAILS OF VARIABLES PERTAINING TO SEGMENT i, i+1 OF BOUNDARY C_n

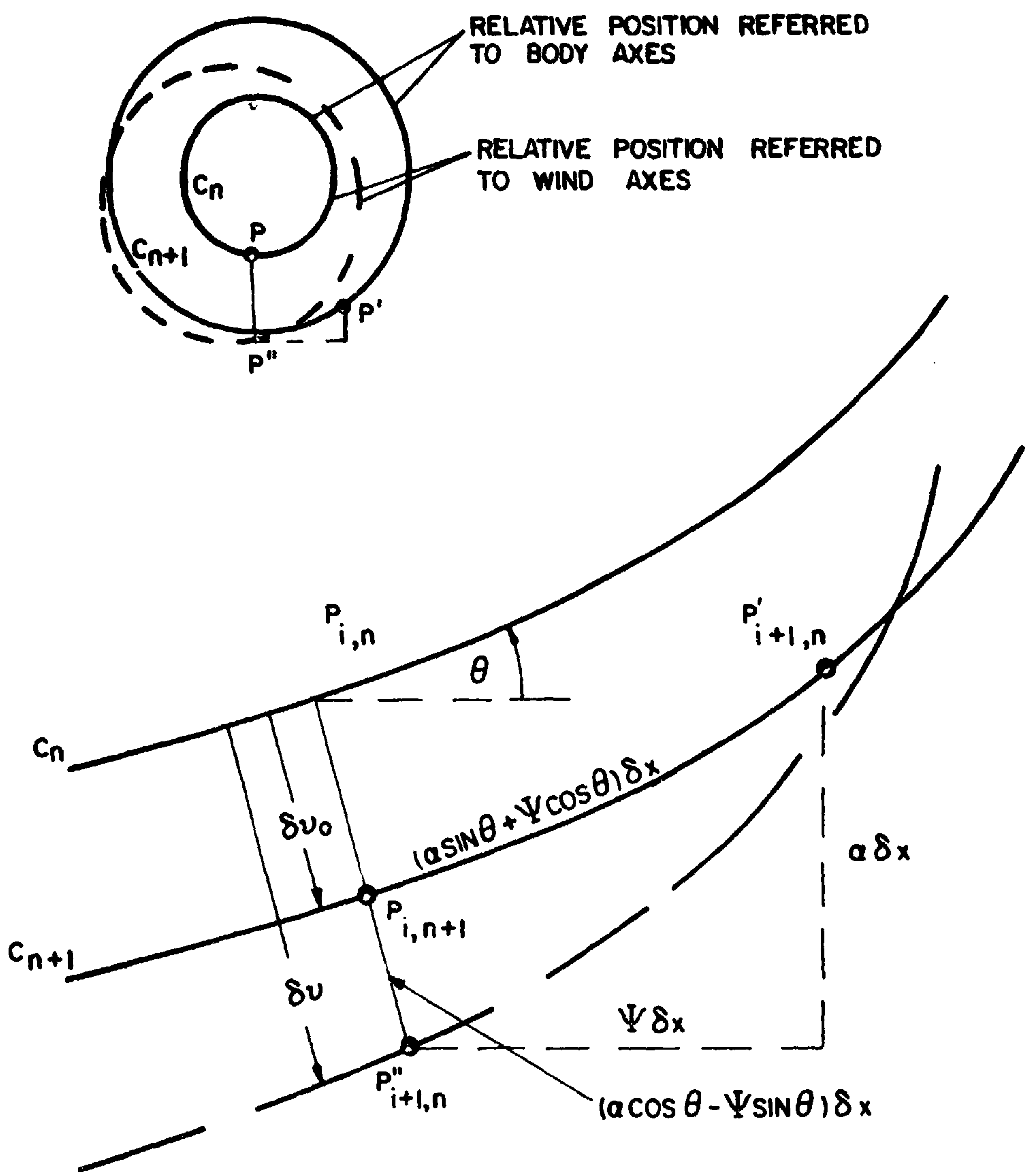


FIG. 4 RELATIVE POSITIONS OF C_n AND C_{n+1} IN BODY AXIS AND WIND AXIS REFERENCE FRAMES

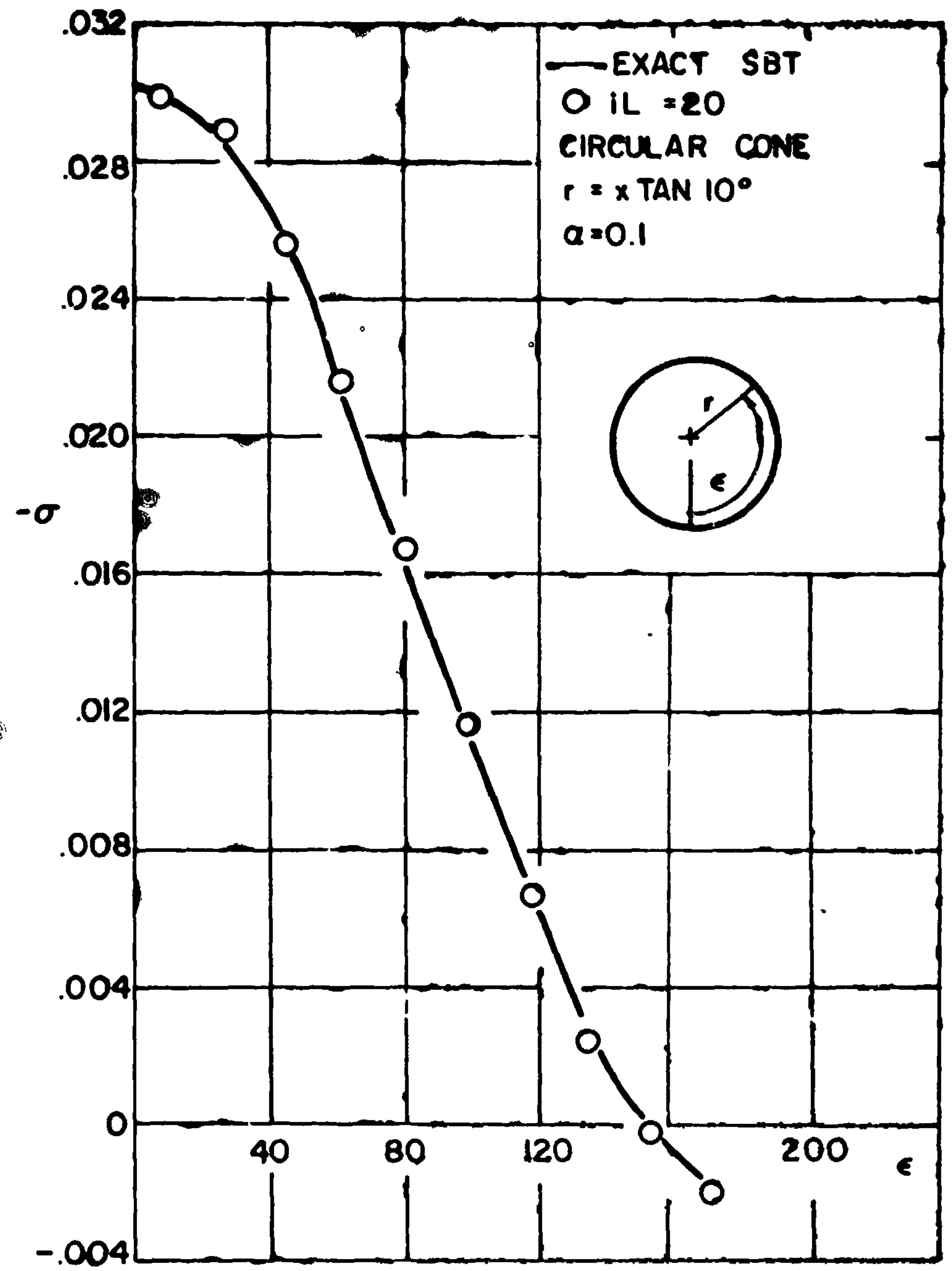


FIG.5 SOURCE STRENGTH σ ON CIRCULAR CONE AT ANGLE OF ATTACK, α=0.1

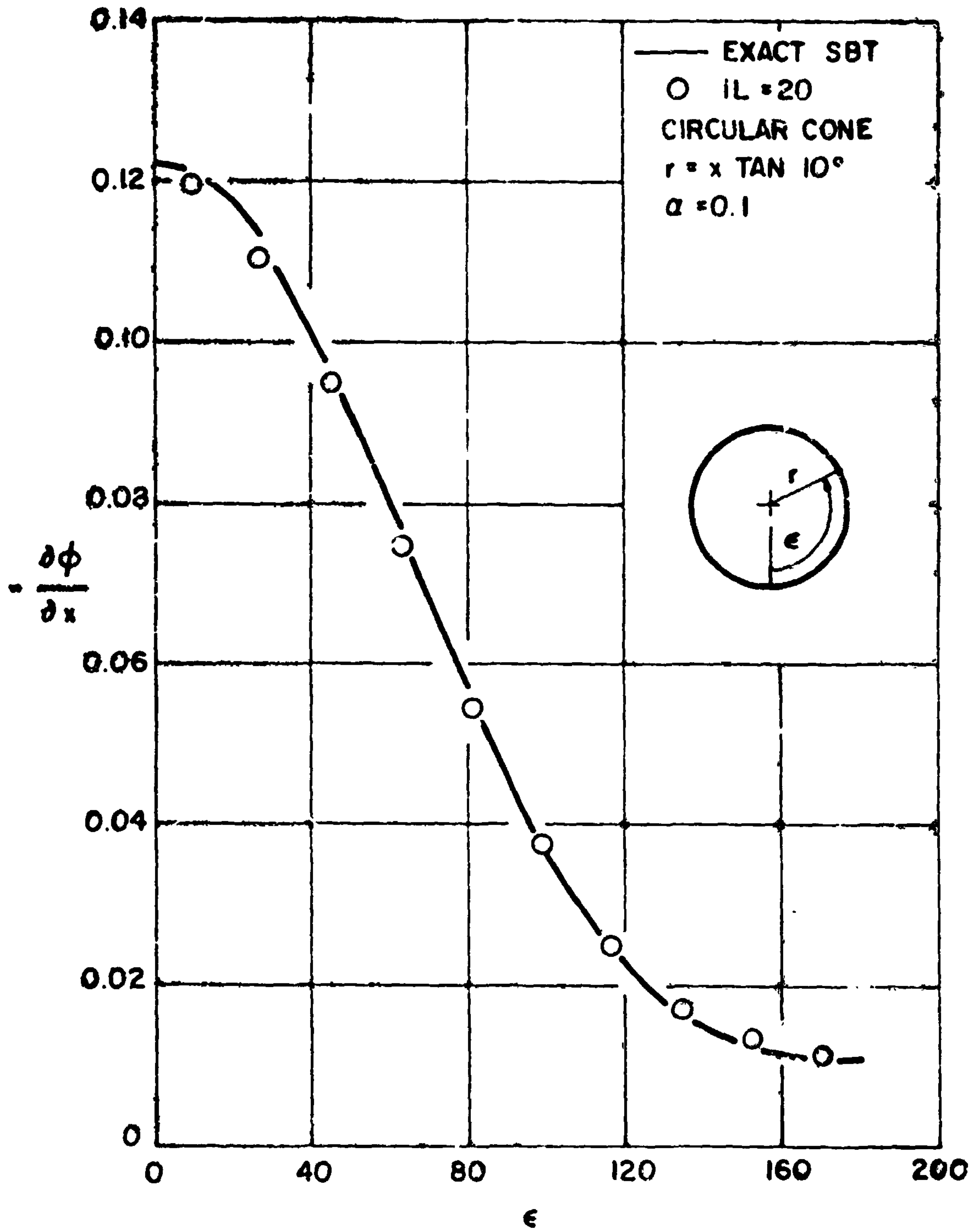


FIG. 6 $\partial\phi/\partial x$ AT X = 1 ON CIRCULAR CONE AT ANGLE OF ATTACK, α = 0.1

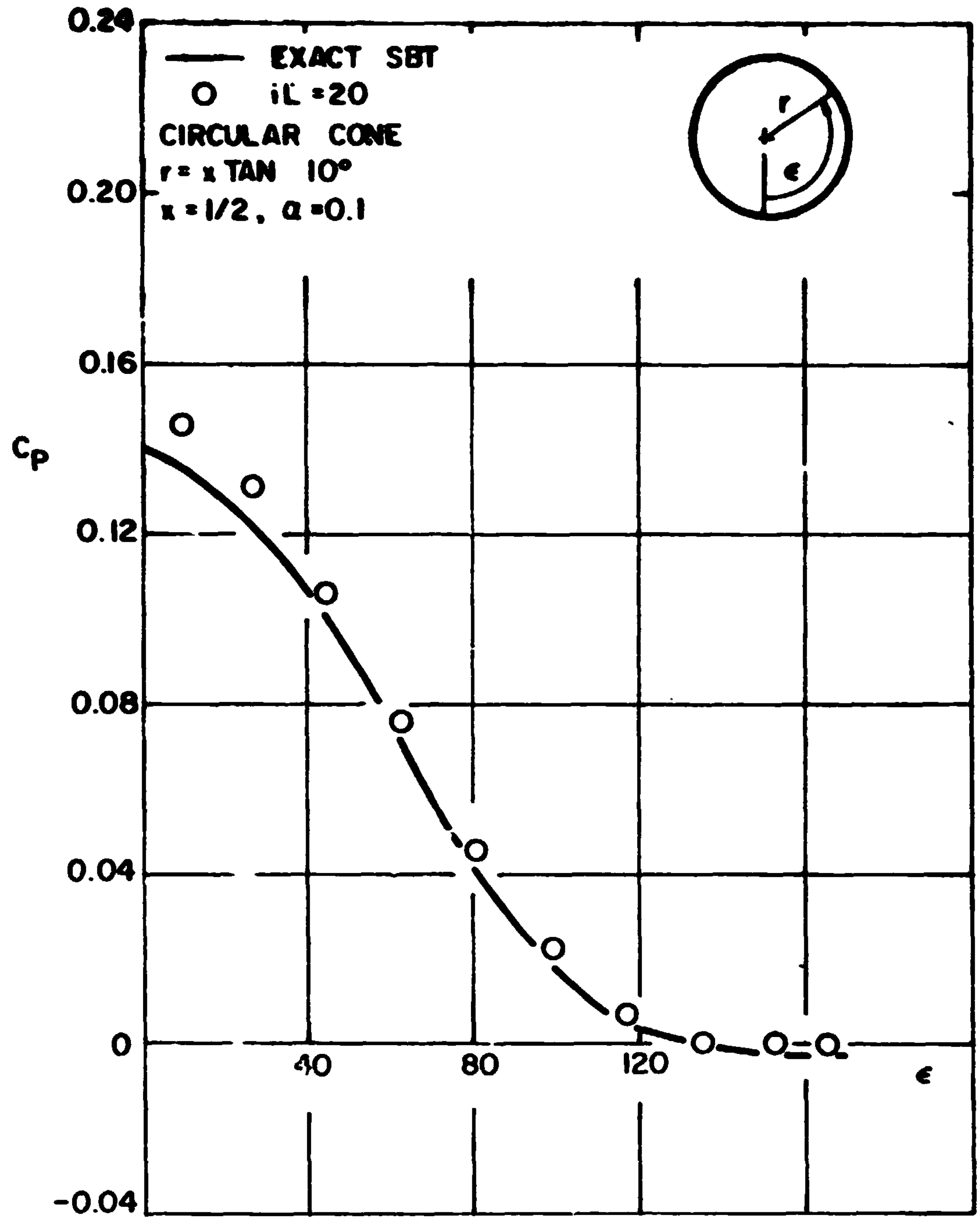


FIG. 7 PRESSURE COEFFICIENT AT x = 1/2 ON CIRCULAR CONE AT ANGLE OF ATTACK, α = 0.1 AND MACH NO. = 0

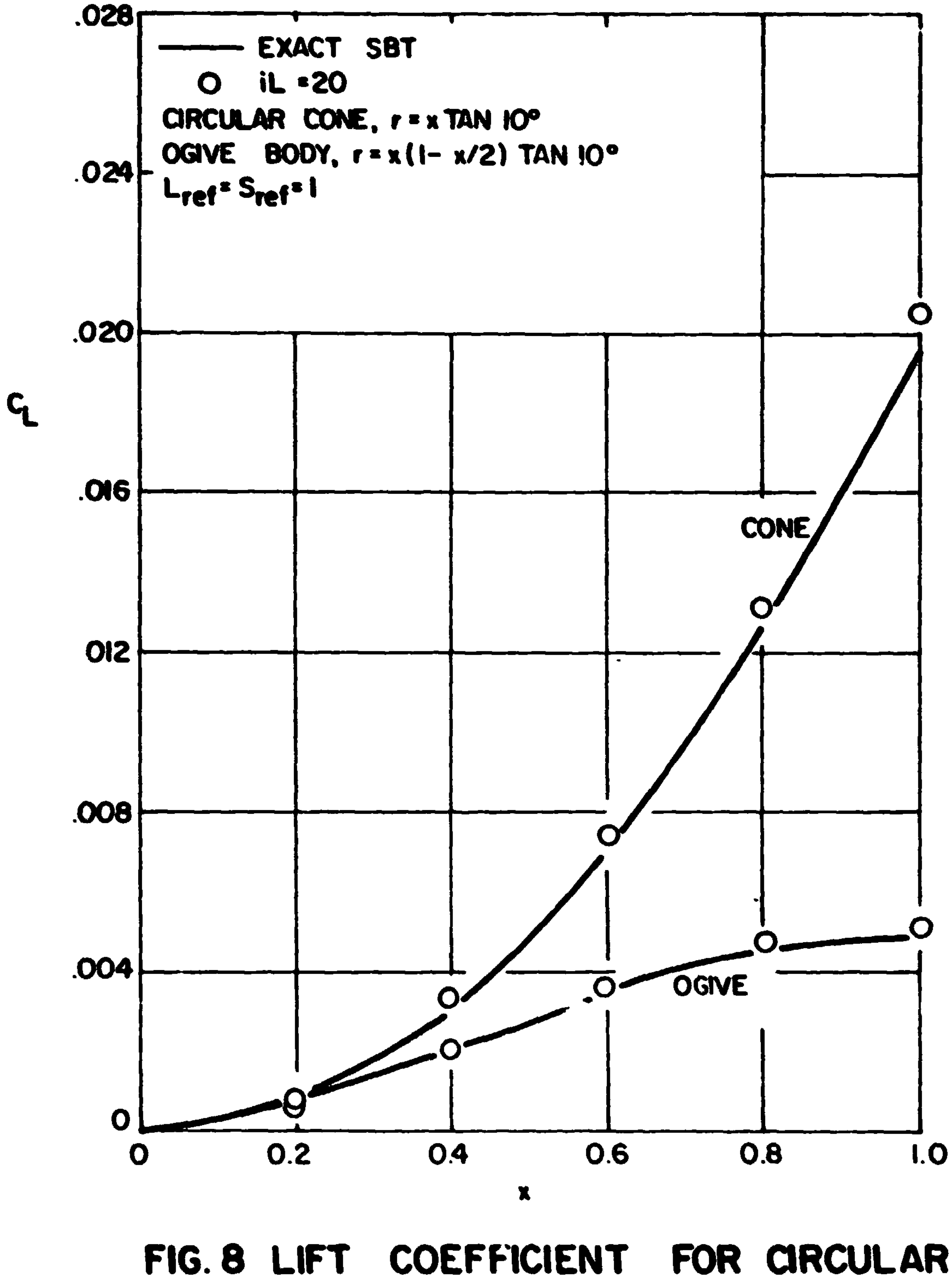


FIG. 8 LIFT COEFFICIENT FOR CIRCULAR CONE AND OGIVE AT ANGLE OF ATTACK, $\alpha = 0.1$

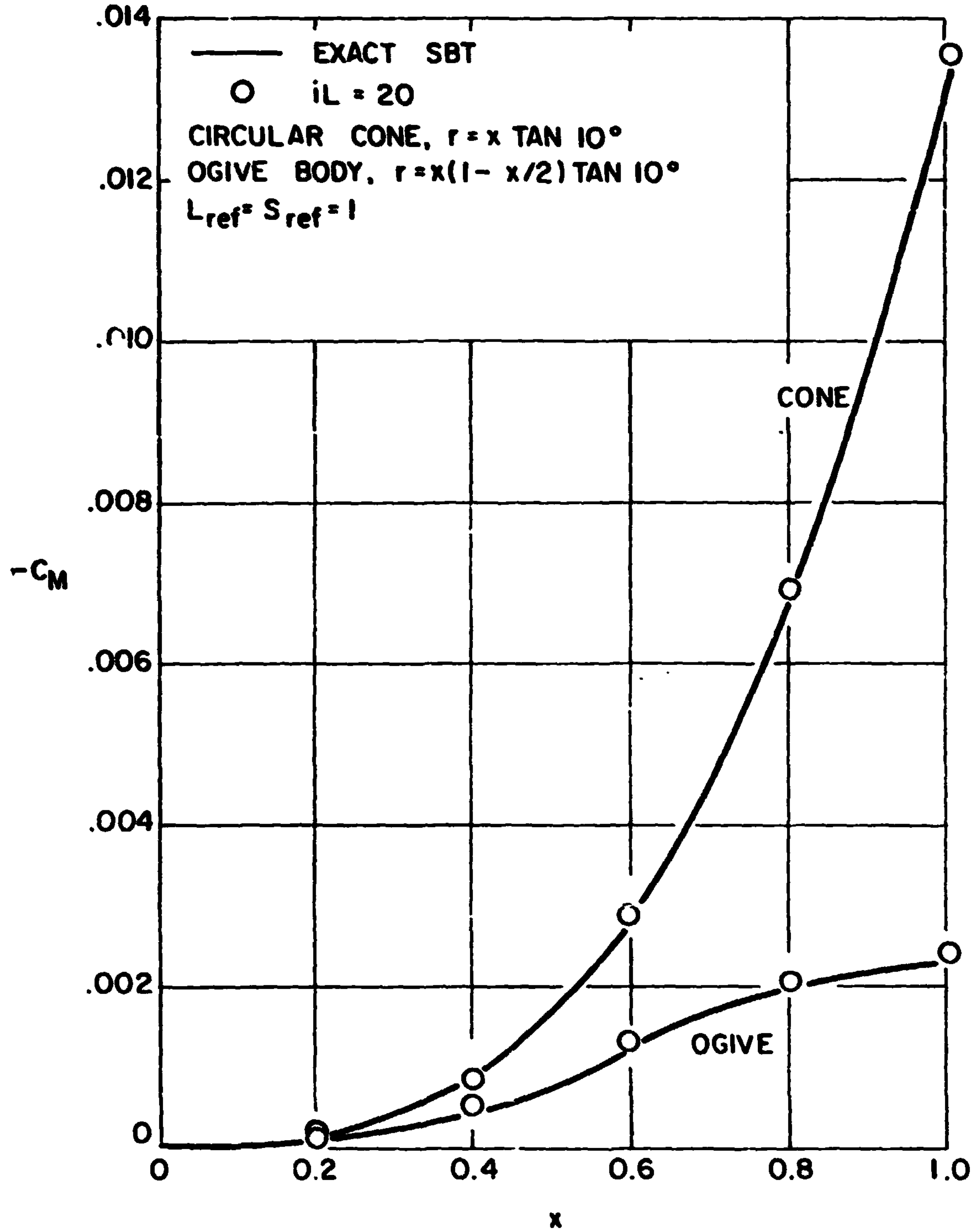


FIG. 9 MOMENT COEFFICIENT FOR CIRCULAR CONE AND OGIVE AT ANGLE OF ATTACK, α = 0.1

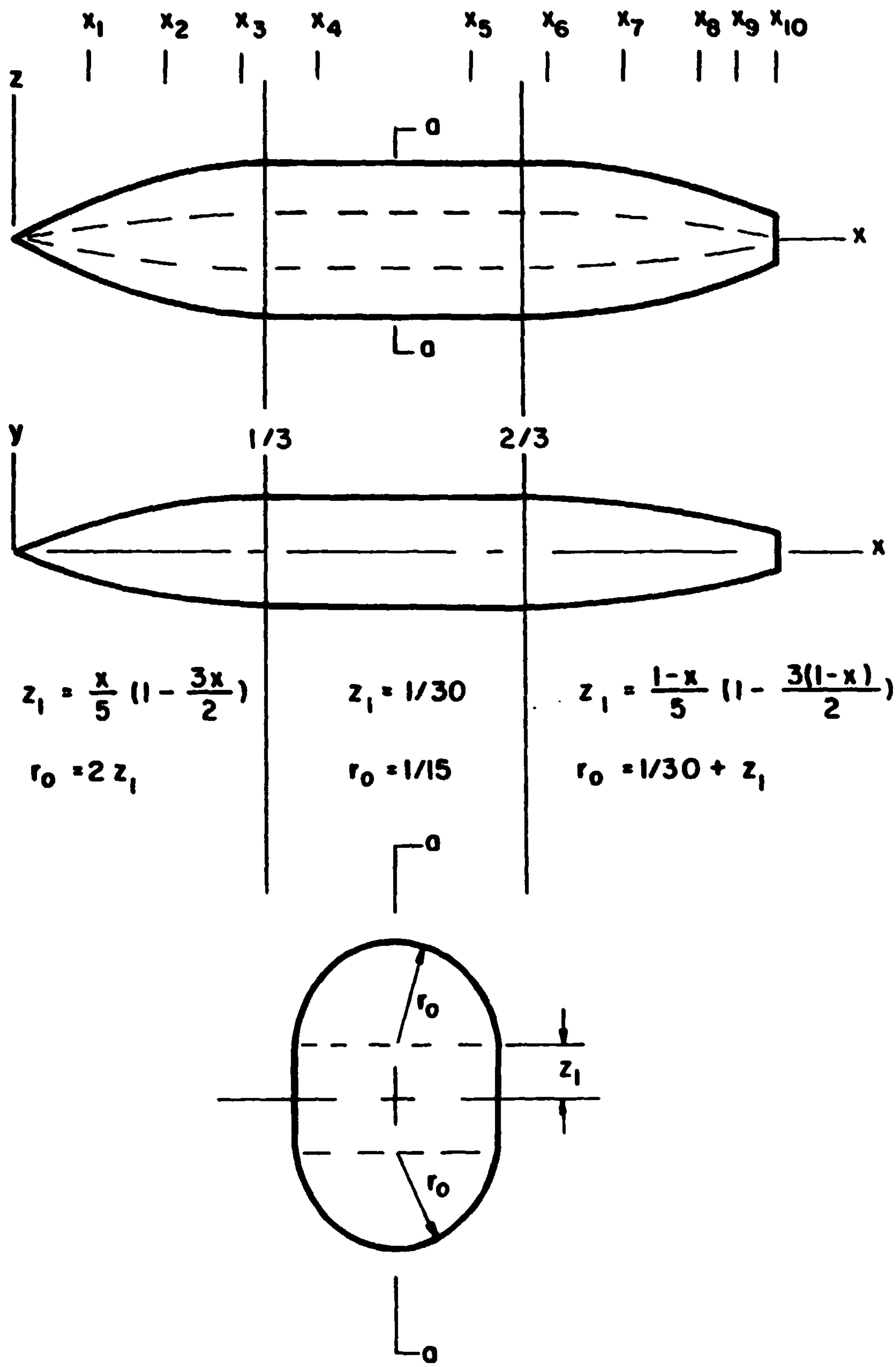


FIG. 10 SAMPLE FUSELAGE

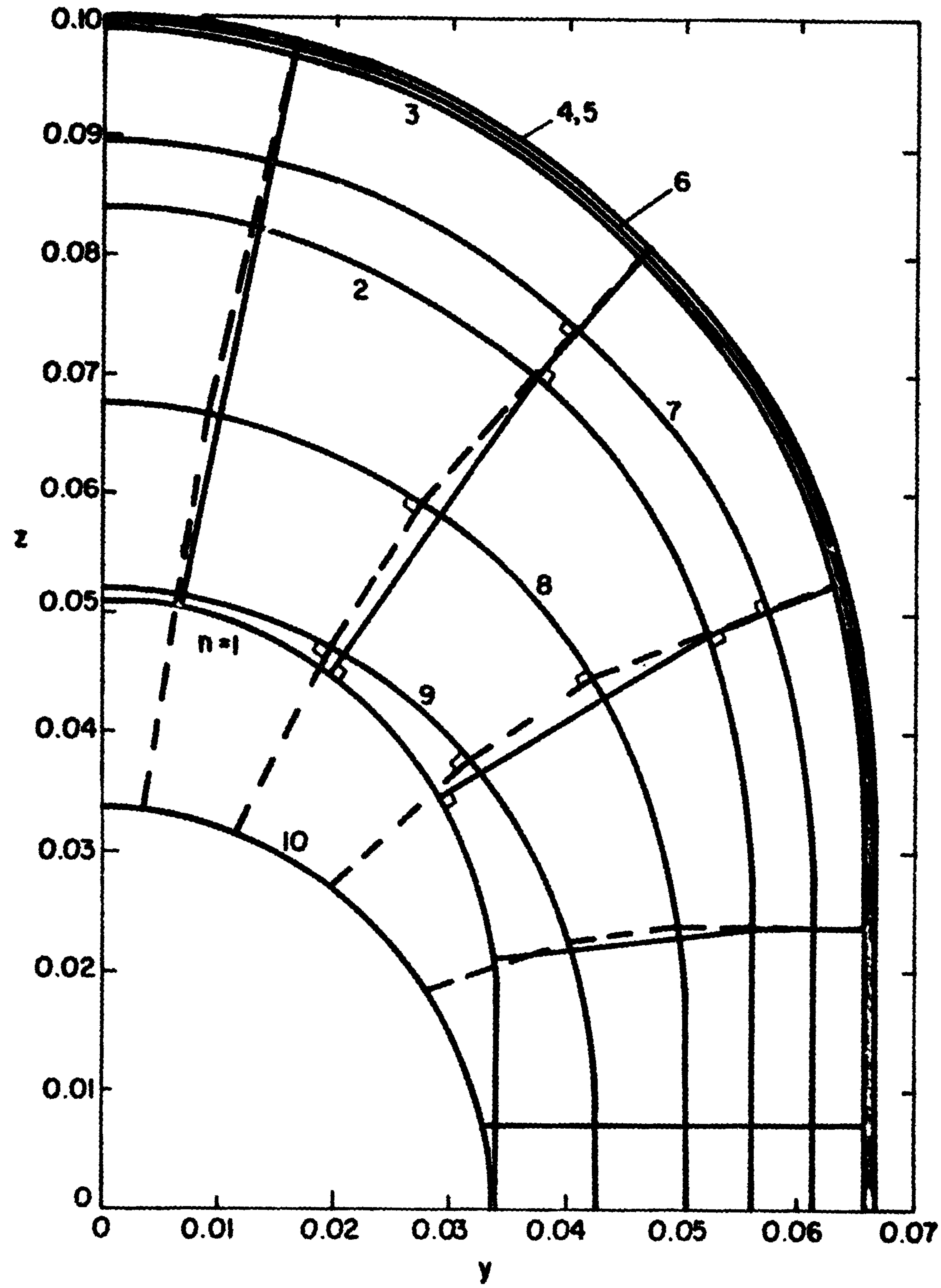


FIG. 11 GENERATION OF INPUT DATA FOR SAMPLE FUSELAGE

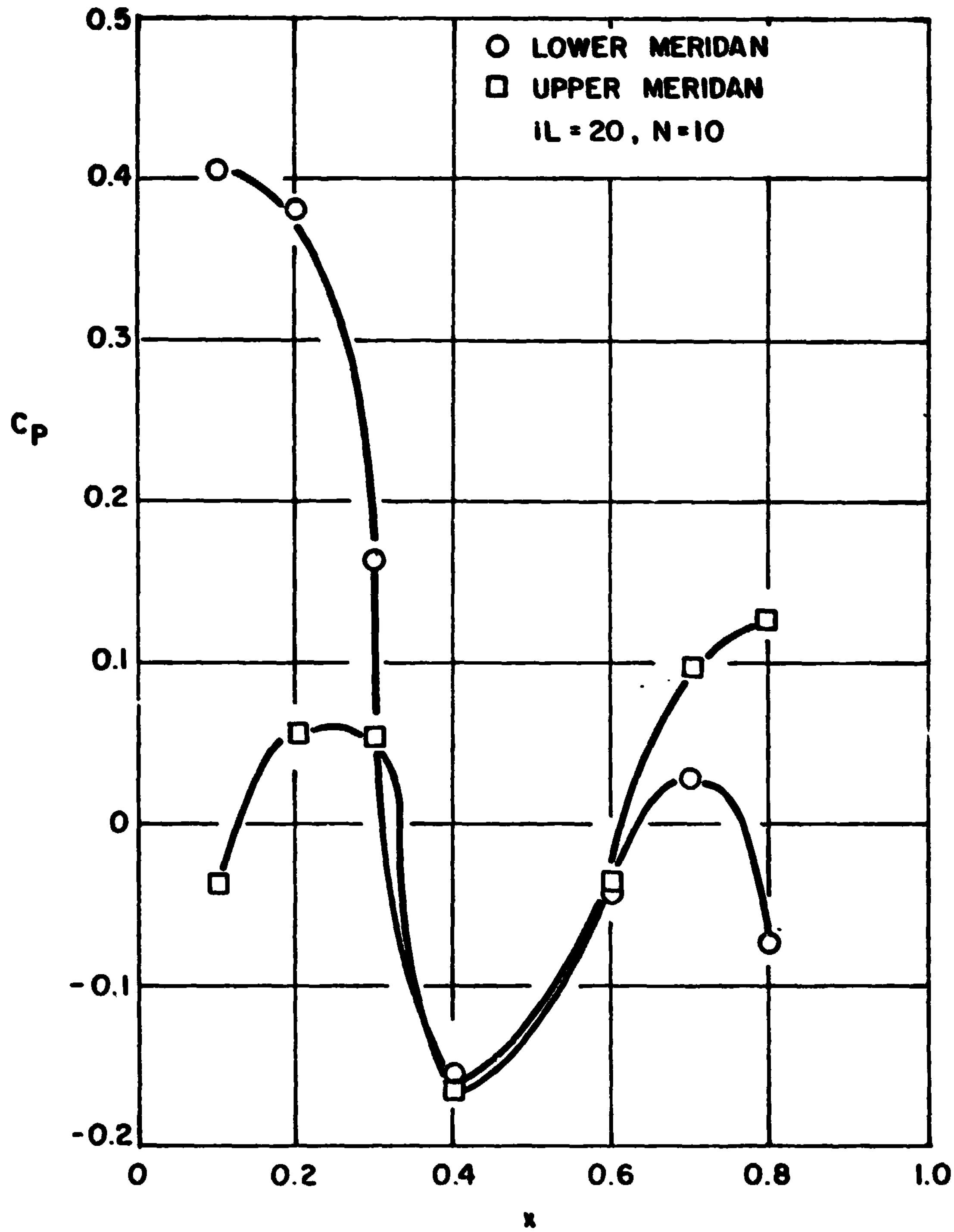


FIG. 12 C_P ALONG UPPER AND LOWER MERIDAN OF SAMPLE FUSELAGE AT ANGLE OF ATTACK, $\alpha = 10°$ AND MACH NO. = 0

ORIGINAL PAGE IS
OF POOR QUALITY

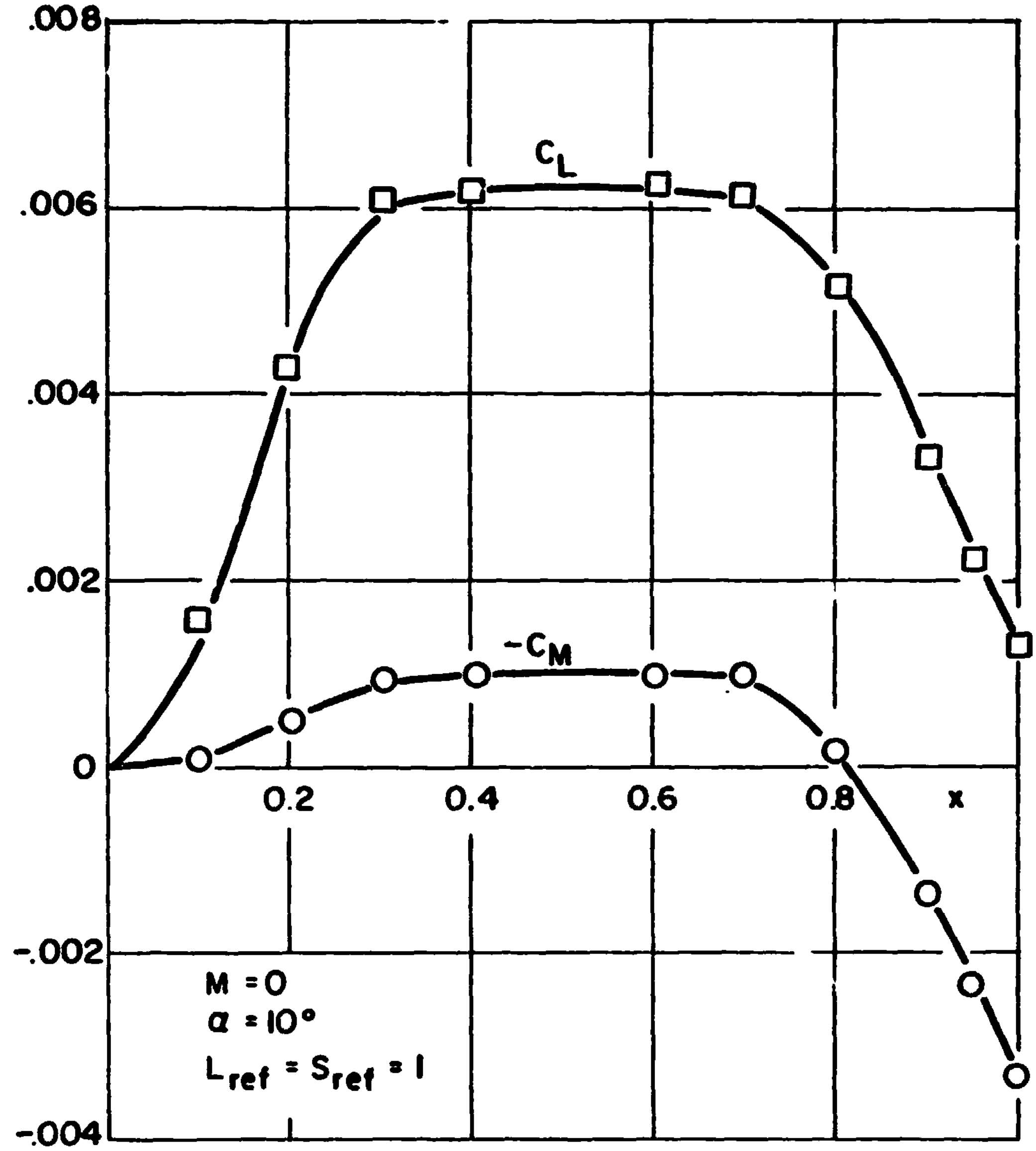


FIG. 13 C_L AND C_M FOR SAMPLE FUSELAGE

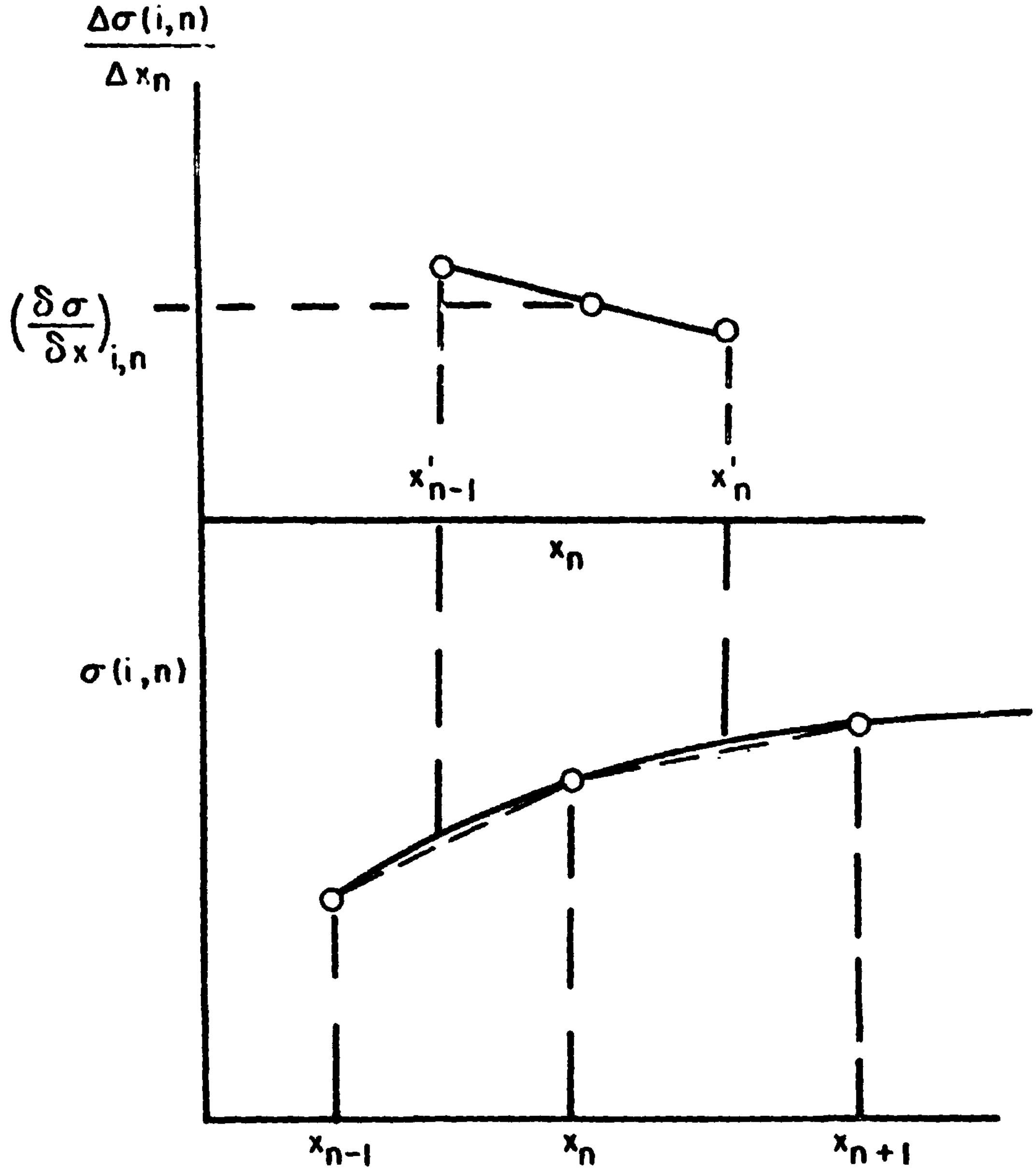


FIG. 14 INTERPOLATION PROCEDURE FOR DETERMINATION OF $(\delta\sigma/\delta x)_{i,n}$

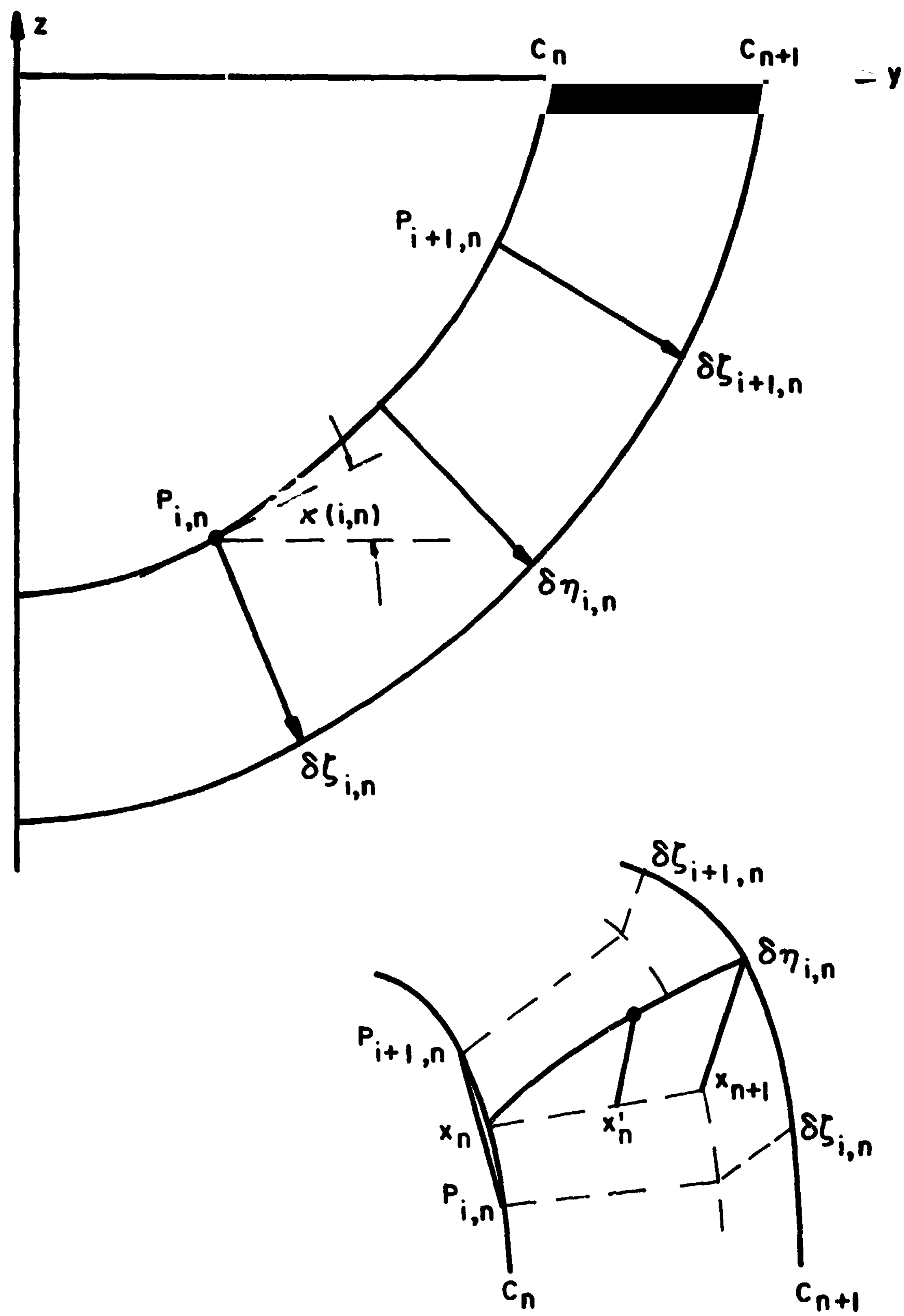
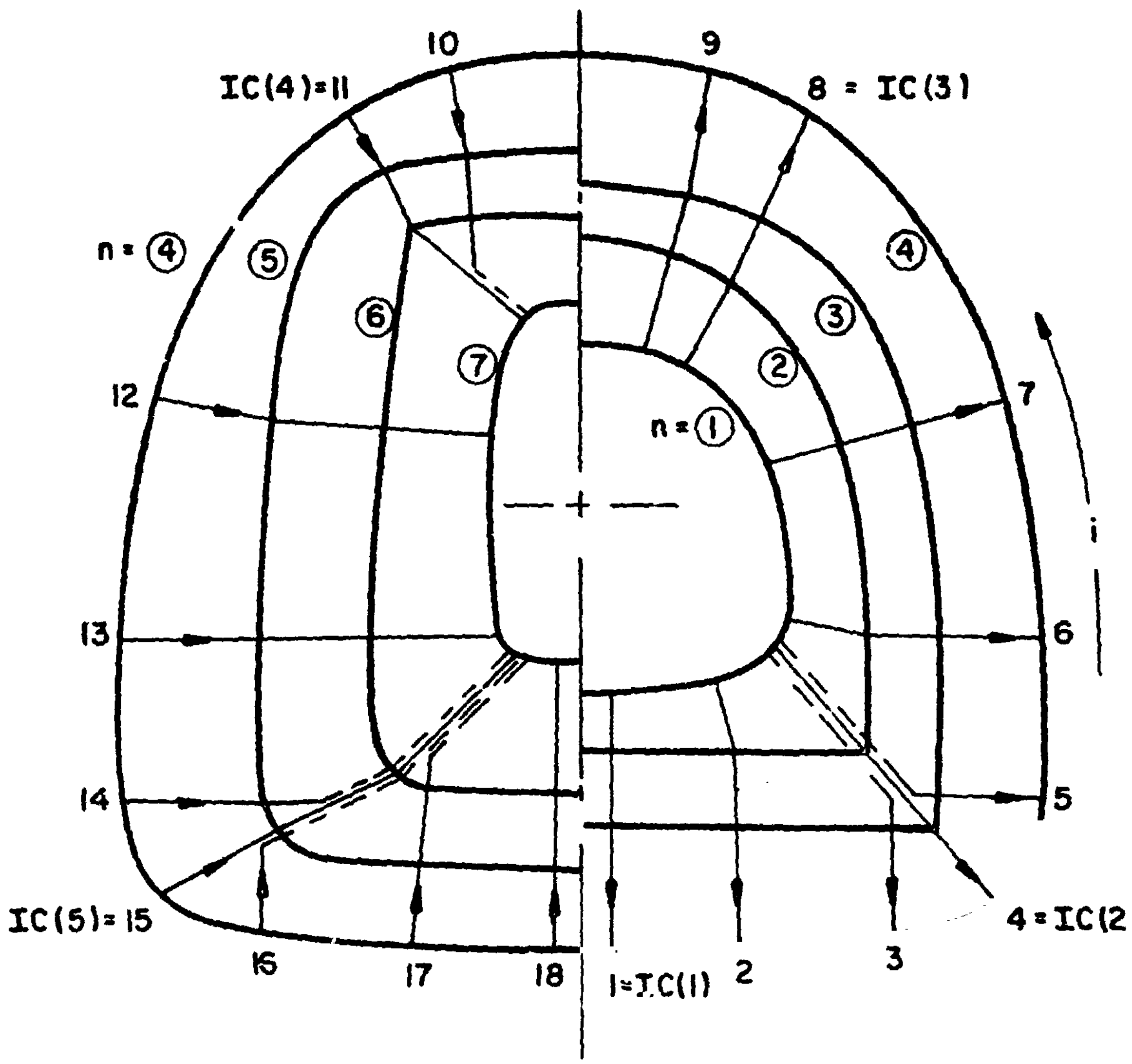


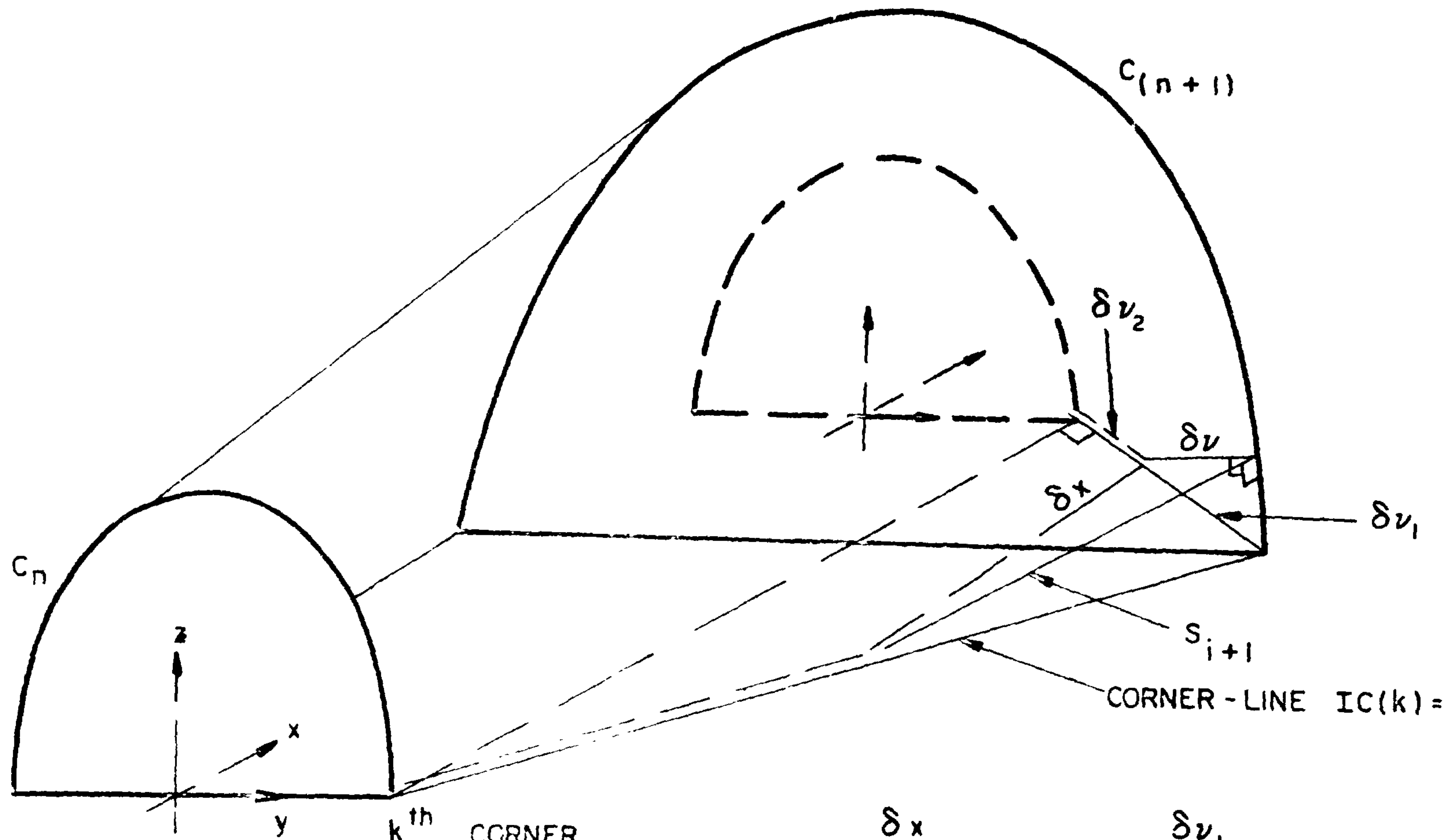
FIG. 15 INTERPOLATION PROCEDURE FOR DETERMINATION OF $(\delta v_0/\delta x)_{i,n}$



IBP(k,n)

	n = 1	2	3	4	5	6	7
k = 1	1	1	1	1	1	1	1
2	5	5	4	4	5	5	5
3	8	8	8	8	8	8	9
4	11	11	11	11	11	11	11
5	16	16	15	15	16	16	16

FIG. 16 ILLUSTRATION OF SEGMENTING SCHEME FOR CONTOURS WITH CORNERS



$$\frac{\delta x}{x(n+1) - x(n)} = \frac{\delta\nu_1}{\delta\nu_1 + \delta\nu_2}$$

FIG. 17 GEOMETRY NEAR A CORNER-LINE