

RETURN TO
B
R
R
DES
JW
R
M

uf



AERODYNAMIC FORCES AND TRAJECTORIES OF SEPARATED STORES IN DISTURBED FLOW FIELDS

W. N. MacDermott and P. W. Johnson

ARO, Inc.

March 1973

Approved for public release; distribution unlimited.

**PROPULSION WIND TUNNEL FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE**

NOTICES

When U. S. Government drawings specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Documentation Center.

References to named commercial products in this report are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

AERODYNAMIC FORCES AND TRAJECTORIES
OF SEPARATED STORES
IN DISTURBED FLOW FIELDS

W. N. MacDermott and P. W. Johnson
ARO, Inc.

Approved for public release; distribution unlimited.

FOREWORD

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC) under sponsorship of the Air Force Armament Laboratory (AFATL), Air Force Systems Command (AFSC), under Program Element 62602F. Technical monitor for AFATL was Major William Miller.

The results of this research were obtained by ARO, Inc. (a subsidiary of Sverdrup & Parcel and Associates, Inc.), contract operator of AEDC, AFSC, Arnold Air Force Station, Tennessee. The work was performed from August 16, 1971, through June 30, 1972, under ARO Project No. PW5280, and the manuscript was submitted for publication on September 21, 1972.

This technical report has been reviewed and is approved.

MAURICE A. CLERMONT
Major, CF
Research and
Development Division
Directorate of Technology

ROBERT O. DIETZ
Director of Technology

ABSTRACT

A vortex-lattice potential flow computer program capable of accepting nonuniform flow boundary conditions but previously restricted to incompressible flows with symmetry was modified to eliminate these restrictions. The program was structured in such a way that, after preliminary calculations of a purely geometric nature were performed one time for a given body, potential flow solutions for any set of boundary conditions on that body could be obtained in computer times measured in seconds rather than minutes. The aerodynamic characteristics of an M-117 bomb, represented by a network of 312 vortices, were calculated for uniform flow at a Mach number of 0.5 and were found to agree with wind tunnel measurements to within 10 percent, except for drag. The program was also used to compute forces on an M-117 bomb at a number of different locations in the disturbed flow field generated by an F-4C parent aircraft. In this case, absolute values of the force coefficients were generally in poor agreement with wind tunnel values, but the incremental variations of the calculated coefficients through the nonuniform flow field were within the range from 5 to 10 percent of wind tunnel measurements. A store separation routine was added to the potential flow program, and several representative store separation trajectories were calculated.

CONTENTS

	<u>Page</u>
ABSTRACT	iii
NOMENCLATURE	viii
I. INTRODUCTION	1
II. DESCRIPTION OF ANALYTICAL METHODS	
2.1 Potential Flow Calculations	2
2.2 Six-Degree-of-Freedom Store Trajectory Calculations	6
III. VORTEX-LATTICE AND STORE TRAJECTORY COMPUTER PROGRAM	
3.1 Description of Program A	8
3.2 Description of Program B	10
3.3 Description of Program C	11
3.4 Computing Time	13
IV. RESULTS OF CALCULATIONS	
4.1 Corrections to Calculated Force Coefficients	14
4.2 Correction of Experimental Force Data	16
4.3 Comparison of Calculations and Measurements in Uniform Flow	16
4.4 Comparison of Calculations and Measurements in F-4C Flow Field	17
4.5 Typical Store Separation Trajectories	19
V. SUMMARY	20
REFERENCES	23

APPENDIXES

I. ILLUSTRATIONS

Figure

1. Shape of M-117 Bomb
 - a. Dimensional Sketch of 1/20-Scale M-117 Bomb
 Model 27
 - b. 312-Vortex Model for Approximation of M-117
 Bomb 27

<u>Figure</u>	<u>Page</u>
2. Comparison of Calculated and Experimental Force Coefficients on M-117 Bomb, Uniform Flow	
a. C_N versus Angle of Pitch	28
b. C_m versus Angle of Pitch	28
3. Effect of Mach Number on Calculated and Experimental Force Coefficients, Uniform Flow	
a. Calculated Normal-Force Coefficient	29
b. Calculated Pitching-Moment Coefficient	29
c. Calculated Axial-Force Coefficient	29
4. Locations of Bomb under F-4C Parent Aircraft at Which Calculated and Measured Forces Were Compared . . .	30
5. Calculated Variation of C_N and C_m with Pitch Angle in F-4C Flow Field, $Y_P = -3.16$, $X_P = 12.99$, $M = 0.50$	31
6. Comparison of Absolute Values of Calculated and Experimental Force Coefficients in F-4C Flow Field, $M = 0.5$, $Y_P = -3.16$	
a. C_N versus ZP, Constant Pitch, Zero Yaw . . .	32
b. C_m versus ZP, Constant Pitch, Zero Yaw . . .	32
c. C_Y versus ZP, Constant Yaw, Zero Pitch . . .	33
d. C_n versus ZP, Constant Yaw, Zero Pitch . . .	33
7. Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, $M = 0.5$	
a. C_N versus ZP, Constant Pitch, Zero Yaw, $Y_P = -3.16$	34
b. C_N versus ZP, Constant Pitch, Zero Yaw, $Y_P = -4.66$	34
c. C_N versus ZP, Constant Pitch, Zero Yaw, $Y_P = -6.16$	35
d. C_m versus ZP, Constant Pitch, Zero Yaw, $Y_P = -3.16$	35
e. C_m versus ZP, Constant Pitch, Zero Yaw, $Y_P = -4.66$	36
f. C_m versus ZP, Constant Pitch, Zero Yaw, $Y_P = -6.16$	36
g. C_Y versus ZP, Constant Yaw, Zero Pitch, $Y_P = -3.16$	37

FigurePage

7.	Continued	
h.	C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -4.66	37
i.	C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -6.16	38
j.	C_n versus ZP, Constant Yaw, Zero Pitch, YP = -3.16	38
k.	C_n versus ZP, Constant Yaw, Zero Pitch, YP = -4.66	39
l.	C_n versus ZP, Constant Yaw, Zero Pitch, YP = -6.16	39
8.	Average Discrepancy Between Calculated and Experimental Force Coefficient, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.5	
a.	Average Discrepancy in C_N , C_m , C_Y , C_n , and Overall	40
b.	Variation of Average Theoretical/Experimental Discrepancy with ZP	40
9.	Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, M = 0.85	
a.	Best Comparison, C_Y versus ZP, Constant Yaw, Zero Pitch, YP = -6.16	41
b.	Worst Comparison, C_m versus ZP, Constant Pitch, Zero Yaw, YP = -6.16	41
10.	Average Discrepancy Between Calculated and Experimental Force Coefficients, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.85	
a.	Average Discrepancy in C_N , C_m , C_Y , C_n , and Overall	42
b.	Variation of Average Discrepancy with ZP	42
11.	Comparison of Separation Trajectories with Uniform Flow Force Coefficients and Nonuniform Flow Force Coefficients, F-4C Aircraft, 250-lbm M-117 Bomb, on No. 2 TER Station, Inboard Pylon, M = 0.50, 5000-ft Altitude, Pitch Angle of A/C = 0.30 deg, 1000-lb Ejector Force at 45 deg to Vertical	43
II.	USER'S GUIDE TO COMPUTER PROGRAM	44

NOMENCLATURE

$A^{(1)}, A^{(2)}$	Submatrices of the H^{-1} matrix, defined in Section 3.2
$\vec{B}$	Matrix of direction cosines of unit normals to surface
ΔC_A	Base pressure correction, Eq. (20)
$\Delta C_{A_{base}}$	Correction to calculated axial-force coefficient for a base force, Eq. (16)
$\Delta C_{A_{fric}}$	Frictional correction to axial-force coefficient, Eq. (19)
C_A, C_N, C_Y	Axial, normal, side-force coefficients, respectively
C_F	General force coefficient
C_l, C_m, C_n	Rolling, pitching, yawing-moment coefficients, respectively
C_p	Pressure coefficient
$\vec{F}_i$	Force on i^{th} segment of a horseshoe vortex, defined by Eq. (10)
$\vec{G}^{(1)}, \vec{G}^{(2)}$	Submatrices of the $\vec{G}$ matrix, defined in Section 3.1
$\vec{G}_{k,j}$	Geometric influence coefficient, Eq. (14), Ref. 1.
H	Matrix of geometric factors defined by Eq. (8)
$H^{(1)}, H^{(2)}$	Submatrices of the H matrix, defined in Section 3.2
M	Mach number
N_j	Number of horseshoe vortices in a network representing a surface
$NUFF$	Nonuniform flow field
$\vec{n}$	Unit vector normal to surface
p	Static pressure
$\vec{q}_{k,j}$	Velocity at field point k induced by vortex j , Eq. (5)
Re_x	Reynolds number based on model length
S_{ref}	Reference area, 0.5025 in. ²
$\vec{U}_\infty$	Velocity in flow field undisturbed by store
$\vec{V}$	Velocity in flow field, including induced velocity, $= \vec{U}_\infty + \Sigma \vec{q}$, Eq. (15)

X, Y, Z	Cartesian coordinates in parent aircraft reference system, positive forward, to the right, and down, as seen by pilot
XP, YP, ZP	Coordinates of nose of store model in parent aircraft coordinate system
x, y, z	Cartesian coordinates in store reference system
β	$\sqrt{1 - M_\infty^2}$
Γ_j	Strength of horseshoe vortex j
ν	Angle of pitch, positive—nose up
ρ	Density
Φ	Velocity potential
Ψ	Angle of yaw, positive—nose right

SUBSCRIPTS

b	Base
base	Base of bomb
comp	Compressible flow
cross	Cross-sectional area
i	Straight-line segment of a horseshoe vortex
inc	Incompressible flow
j	Horseshoe vortex
k	Field point at which velocity is calculated
plan	Planform area
S	Area
∞	Free-stream condition

SECTION I INTRODUCTION

In recent years, testing of aircraft store separation characteristics has emerged as one of the major uses of developmental wind tunnels. As a consequence of the many possible configurations and flight conditions of interest, such testing is characterized by a voluminous production of test data. To assist in the evaluation of tunnel data and in its extrapolation to full scale, several analytical techniques of store separation trajectory calculation have been under development. These techniques possess varying degrees of dependence on experiment, from nil to a major dependence. Because of shortcomings of aerodynamic theory and present digital computer limitations, any completely theoretical calculation apparently will remain an unattainable ideal for a number of years in the future. An analytical method which places a major reliance on wind tunnel measurements is the so-called "grid method," in which force characteristics on a given store are measured at a large number of points within a volume beneath a given parent aircraft, stored in some accessible form, and finally are used as a source of force coefficient data (interpolated) at points along a distinct trajectory developed by double integration of the equations of motion. This technique has achieved a modicum of use, but is limited to a single store/aircraft combination for a given set of experimental measurements.

During the past two years, the staff of the AEDC Propulsion Wind Tunnel (PWT) has attempted to develop an alternate technique having one step less dependence on wind tunnel measurements. This technique is built around the use of potential flow solutions for a given store, using as boundary conditions on the store the disturbance flow angles measured in the nonuniform flow field for a given parent aircraft. Ideally, this approach would allow a single set of wind tunnel measurements for a single parent aircraft to be used with any number of different store calculations.

As reported in Ref. 1, a potential flow program based on vortex singularities was developed in a form allowing imposition of boundary conditions based on known nonuniform flow fields. Considerable effort was expended in developing a vortex network representation of the M-117 bomb which would produce calculated force characteristics sufficiently close to values measured in the wind tunnel. A network composed of 140 vortices representing one half of the M-117 bomb shape and 16 vortices representing a wake at the base of the bomb (Fig. 1, Appendix I) was found to give 10-percent agreement (except for drag) with force measurements made in a uniform flow. This model was then used to

calculate force characteristics on the M-117 bomb at various locations within the nonuniform flow field (NUFF) of the F-4C aircraft. This calculation was made only in an approximate sense, since only the downwash components of the NUFF were allowed for, and even these were averaged laterally. In this approximation, the absolute values of the calculated force coefficients differed appreciably from wind tunnel measurements, but the incremental behavior of the calculated force coefficients within the NUFF showed the same trends as did the wind tunnel measurements. Based on this observation, the project effort was continued, and the results of subsequent calculations are presented in this report.

SECTION II DESCRIPTION OF ANALYTICAL METHODS

2.1 POTENTIAL FLOW CALCULATIONS

2.1.1 Boundary-Value Problem for a Velocity Potential

As discussed in Ref. 2, the analytical description of the flow field surrounding an aerodynamic body can be obtained from a solution of the differential equation for a velocity potential, i. e., Laplace's equation, in the incompressible case,

$$\nabla^2 \Phi = 0 \quad (1)$$

This assumes steady, irrotational, inviscid, attached flow. In the general case of compressible flow, the differential equation of the velocity potential is highly nonlinear, and exact solutions are not generally possible to obtain. Almost all work in the compressible regime has been done in the linearized approximation assuming perturbations to the free stream are small. The velocity potential, Φ , then satisfies

$$(1 - M_\infty^2) \frac{\partial^2 \Phi}{\partial X^2} + \frac{\partial^2 \Phi}{\partial Y^2} + \frac{\partial^2 \Phi}{\partial Z^2} = 0 \quad (2)$$

the Neumann condition on the planform surface

$$\vec{V} \cdot \vec{n} = -\nabla \Phi \cdot \vec{n} = 0 \quad (3)$$

and the boundary condition at infinity

$$\nabla \Phi = -\vec{U}_\infty \text{ at } X=Y=Z = \pm \infty \quad (4)$$

Since the Laplace equation is linear, superposition of elementary solutions (sources, sinks, doublets, and vortices) will allow buildup of an incompressible flow over quite general shapes satisfying the boundary conditions, Eqs. (3) and (4).

Since it was desired to allow for trailing vorticity in the flow, caused by three-dimensional lifting surfaces, a potential flow program was developed based on superposition of vortex elements alone, excluding use of source, sinks, and doublets. This program was modelled after that of Ref. 3.

Prandtl introduced the concept that a solid body immersed in a flow is equivalent in external flow to a sheet of continuously distributed vorticity at the surface of the body. A relation exists between the velocity at any point and the strength and orientation of the distributed vorticity. Since this relationship cannot be solved in closed form, it is necessary, for computational reasons, to replace the distributed vortex sheet representation of a solid body (an exact concept) with a series of concentrated vortex elements having the same net strength (an approximation), for which a simple relationship does exist. This relationship is identical with the Biot-Savart law, which describes the magnetic field induced by an electric current flowing through a conductor. In its fluid-mechanical formulation, the Biot-Savart law is a differential relationship (see Eq. (7) of Ref. 1) which can be integrated in closed form along a straight-line segment of a vortex (the description of a vortex segment is given in Fig. 4c of Ref. 1). The result for a single horseshoe vortex composed of a number of straight-line vortex segments can be expressed as

$$\vec{q}_{k,j} = \vec{G}_{k,j} \Gamma_j \quad (5)$$

where $\vec{q}_{k,j}$ is the velocity at field point k due to vortex j , Γ_j is the strength of vortex j , and $\vec{G}_{k,j}$ is a geometric influence coefficient representing a summation over all segments of the horseshoe vortex given in functional form by Eq. (14) of Ref. 1.

The strengths of the individual vortices, Γ_j , are determined by constraining the flow to be directed in a prescribed manner. Generally this is done by requiring the flow to be tangent to the surface at a set of points referred to as boundary points. The criterion for boundary point location used in obtaining the results described in this report combines simplicity and plausibility - the boundary point coordinates are taken to be centroids of quadrilateral panels formed by overlapping

horseshoe vortices. A single boundary condition is expressed as

$$\vec{V}_k \cdot \vec{n}_k = \left(\vec{U}_{\infty k} + \sum_{j=1}^{N_j} \vec{G}_{k,j} \Gamma_j \right) \cdot \vec{n}_k = 0 \quad (6)$$

which can be recognized as Eq. (3) written at a specific boundary point. The system of N_j equations representing all the required boundary conditions is thus

$$\sum_{j=1}^{N_j} (\vec{n}_k \cdot \vec{G}_{k,j}) \Gamma_j = -\vec{n}_k \cdot \vec{U}_{\infty k} \quad k = 1, \dots, N_j \quad (7)$$

It is convenient to introduce the matrix notation

$$[H_{k,j}] = [\vec{n}_k \cdot \vec{G}_{k,j}] \quad (8)$$

giving

$$[H_{k,j}] \{ \Gamma_j \} = \{ -\vec{n}_k \cdot \vec{U}_{\infty k} \} \quad (9)$$

as the matrix equation to be solved for the unknown vortex strengths. Upon solving Eq. (9) by inversion of the H matrix, the values of vorticity, Γ_j , are substituted into Eq. (5) to obtain the velocity distribution at the field points.

The distribution of aerodynamic forces on a planform is obtained by application of the Kutta-Joukowski law

$$\vec{F}_i = \rho \vec{V}_i \times \vec{\Gamma}_i \quad (10)$$

on each segment i of every vortex j . Finally, summation over all vortex segments is performed to obtain the forces that are used in the definitions of aerodynamic force coefficients. A precaution must be observed in the application of Eq. (10); namely, the velocity vector $\vec{V}_i$ should not include any induced effect of the i^{th} vortex element. That is, $\vec{V}_i$ should be the flow field velocity vector which would exist in the absence of the vortex element, but at the physical location of the element usually assumed to be its midpoint.

The foregoing method of construction of a potential flow is valid whatever the nature of $\vec{U}_{\infty k}$ in the boundary conditions shown in Eq. (7).

In cases of uniform onset flow, it is a constant vector. In the case of

present interest, disturbed flow fields, it is assumed to vary over the surface of the body and is obtained by experimental flow-field measurements in the absence of the store. This approach is legitimate only in the case of simple interference; that is, a large body (parent aircraft) produces a disturbance flow felt by a smaller body (store), but not vice versa. In case the disturbance field of the small body is also felt at the larger body, mutual interference is said to occur, and solution of the potential flow problem is required simultaneously for the two bodies. In the simple interference case it is necessary to solve the potential flow problem for only the smaller body. In practice this means that the present approach is strictly valid only beyond a certain distance of separation defining the limits of mutual interference.

2.1.2 Compressibility Corrections to Incompressible Flow Calculations

Since the Biot-Savart law is valid only for incompressible flow, the vortex-lattice calculations are performed as solutions of the exact equation for the velocity potential of an incompressible flow, Eq. (1). Allowance for compressibility is then made by use of the Goethert similarity rule (Ref. 2), which relates a compressible flow (linearized approximation, Eq. (2)) to a corresponding incompressible flow about an affinely related shape which is decreased in all lateral dimensions by the factor $\beta = \sqrt{1 - M_\infty^2}$. The rule states that the pressure coefficients at corresponding points in these two flows are related by

$$C_{p_{comp}} = \frac{1}{\beta^2} C_{p_{inc}} \quad (11)$$

and, for characteristic force coefficients obtained by use of a reference area which is equal to or proportional to the area over which ΔC_p is integrated to obtain the force (or moment),

$$C_{F_{comp}} = \frac{1}{\beta^2} C_{F_{inc}} \quad (12)$$

For geometrically similar bodies, all areas related to the body are proportional to all other areas, and there are no qualifications required by the choice of reference area. The Goethert rule is not applied to geometrically similar bodies, however, and the form of the force coefficient correction, Eq. (12), does depend on the specific reference area chosen. For slender, axisymmetric shapes such as the M-117 bomb, the cross-sectional area is usually chosen as the reference area, and this area is proportional to the area of integration of C_p only for the axial-force coefficient. All other force coefficients are formed by integration over a planform area; thus,

$$\begin{aligned}
C_{F_{comp}} \left(\frac{S_{plan}}{S_{cross}} \right)_{comp} &= \frac{1}{\beta^2} \left(\frac{S_{plan}}{S_{cross}} \right)_{comp} \left(\frac{S_{cross}}{S_{plan}} \right)_{inc} \left(\frac{S_{plan}}{S_{cross}} C_F \right)_{inc} \\
(C_{F_{comp}})_{S_{cross}} &= \frac{1}{\beta^2} \frac{(S_{plan})_{comp}}{(S_{plan})_{inc}} \times \frac{(S_{cross})_{inc}}{(S_{cross})_{comp}} (C_{F_{inc}})_{S_{cross}} \\
(C_{F_{comp}})_{S_{cross}} &= \frac{1}{\beta^2} \left(\frac{1}{\beta} \times \beta^2 \right) (C_{F_{inc}})_{S_{cross}} \\
(C_{F_{comp}})_{S_{cross}} &= \frac{1}{\beta} (C_{F_{inc}})_{S_{cross}}
\end{aligned} \tag{13}$$

Equation (13) was used for correction of normal-force, side-force, pitching-moment, yawing-moment, and rolling-moment coefficients. It is noted that, in application to NUFF flows, the NUFF is also contracted by the factor β in all directions transverse to the flow.

2.2 SIX-DEGREE-OF-FREEDOM STORE TRAJECTORY CALCULATIONS

A routine was added to the potential flow computer program to allow trajectory calculations through nonuniform flow fields. The trajectory computer program described in this report incorporates the same differential equations in describing six-degree-of-freedom motion as those used in the AEDC/PWT Captive Trajectory System (CTS). The differences between the two approaches are primarily that (1) each program uses different numerical integration techniques and (2) the program of this report uses theoretical force coefficients, whereas the CTS program uses experimentally determined forces.

2.2.1 Equations of Motion

The differential equations describing conservation of linear momentum are Eqs. (II-34), (II-35), and (II-36) of Ref. 4; the angular momentum relationships are given by Eqs. (II-37), (II-38), and (II-39) of Ref. 4. (Equation (II-37) contains a misprint; see Eq. (11) of Ref. 5.) The store rotational sequence is assumed to be pitch-yaw-roll. The equations resulting from this assumption relating store linear velocity in wind axes coordinates to velocity in body axes coordinates are Eqs. (II-13), (II-14), and (II-15) of Ref. 4. (These equations also contain a misprint; the transformation matrix, A , should be replaced by the transpose of A .) The differential equations relating pitch, yaw, and

roll velocities to store rotational velocities about body axes are Eqs. (II-4), (II-6), and (II-8) of Ref. 4. See also page 23, case 4, of Ref. 5 for the coordinate transformations used in the above expressions.

2.2.2 Numerical Integration Method

The motivation that led to the adoption of the particular numerical integration scheme incorporated in the trajectory program of this report was optimization of computer execution time consistent with the accuracy of its results. It was adjudged that accurate results could be obtained using the largest possible step size (in time) from the classical fourth-order Runge-Kutta method. To apply the Runge-Kutta algorithm in its usual form would require transferal of data (the H^{-1} and $\vec{G}$ matrices) from external memory into the computer core each time a single set of force coefficients is to be determined. (Since the size of the H^{-1} and $\vec{G}$ matrices is such that they cannot be stored in-core in their entirety, they must be input repeatedly.) As the data input operation constitutes a significant fraction of the total time required to perform vortex-lattice calculations (and, consequently, trajectory calculations), a procedure has been adopted to compute a number of sets of force coefficients each time the H^{-1} and $\vec{G}$ matrices are input. In order to accomplish this saving in execution time it was necessary to devise a scheme that would incorporate this approach in a modified Runge-Kutta algorithm. The procedure that was constructed is a predictor-corrector method and is described in the following.

A first approximation to the solution for a trajectory is obtained by performing Runge-Kutta calculations over a specified number of time steps (two in the solution subsequently reported herein) using extrapolated values of force coefficients. The values of store orientation obtained in this manner are used to recompute the force coefficients, this time by the vortex-lattice method. The Runge-Kutta calculations are then performed a second time. The force coefficients used in this second approximation to a trajectory are obtained in the form of Taylor series expansions, where the leading terms in the expansions are the vortex-lattice results. The correction term to the drag force is assumed to be a quadratic function of the store angle of incidence with respect to the free stream; the correction terms to the normal force, side force, pitching moment, and yawing moment are all assumed to be proportional to computed values of the static stability derivatives.

SECTION III

VORTEX-LATTICE AND STORE TRAJECTORY COMPUTER PROGRAM

The determination of aerodynamic forces by the vortex-lattice method has been performed by a digital computer. The overall set of calculations has been divided into three separate programs; for descriptive purposes, these will be referenced in this report as Programs A, B, and C. The reasons for this subdivision are twofold:

1. For a given vortex network and Mach number, certain of the calculations are of a purely geometric nature and consequently are performed only one time for a given store (in Programs A and B) and then are stored on magnetic tape, to be used in conjunction with parametric variations of store angle of incidence and location in a surrounding NUFF (in Program C);
2. The computer requirements for performing the geometric calculations, in terms of internal memory and execution time, are of such a magnitude that these computations cannot all be performed in one program; hence, they are performed in two separate programs (A and B).

Program C, which computes aerodynamic characteristics (the distributions of velocity, pressure coefficient, and force coefficients, as well as the total-force coefficients on a store), is written in two separate versions. One of these computes the variables just mentioned for a specified set of spatial orientations of a store with respect to its parent aircraft where the store is assumed to be motionless with respect to wind axes; the other version solves for both the force coefficients and the trajectory of a store separated from its parent aircraft. The portion of the computer code that pertains to the trajectory of a separated store is based upon the nomenclature and equations used in Ref. 4.

A brief description and user's guide of these programs follows. A more detailed description may be obtained from the authors.

3.1 DESCRIPTION OF PROGRAM A

This program consists of a MAIN program and SUBROUTINE ICOEFF. The principal results are the calculation of the $\vec{G}$ and H variables and the transferal of these data to magnetic tape.

The $\vec{G}$ matrices are geometric influence coefficients, the resultant values of which are equal to the sum of contributions from the bound spanwise vortex segments (denoted by the FORTRAN name DIC), bound chordwise segments (EIC), and trailing vortices (FIC). The H matrix is computed from

$$H = \vec{B} \cdot \vec{G}_1 \quad (14)$$

where $\vec{B}$ is the matrix of direction cosines of the unit normals to the surface at the boundary points. $\vec{G}_1$ is the value of $\vec{G}$ obtained when the influence of vortices at boundary points is considered. (It may be noted that Eq. (14) is equivalent to Eq. (8), the only difference being the use of alternate notations.)

The first step leading to the calculation of influence coefficients is the specification of a network of discrete line vortices on the surface of the aerodynamic planform being analyzed. A general description of vortex networks is presented in Section III of Ref. 3 and Fig. 4 of Ref. 1. The specific application to the M-117 bomb that was used to obtain the results of this report is described in Sections 4.3 and 5.1.1 of Ref. 1 and is presented in Fig. 1 of this report.

The equations for G are shown combined in succinct notation in Eq. (14) of Ref. 1; they are presented in the detailed form in which they are incorporated in the computer code in Eqs. (2a), (4a), (7a), (10a), and (12a) of the Appendix to Ref. 3.

In general, all matrices of geometric factors are computed in partitioned form if the planform being analyzed possesses geometric symmetry about its longitudinal axes. In the special case of zero yaw attitude combined with lateral symmetry of the flow field about the store, the matrices can be further simplified (see p. 18 of Ref. 1). In particular, it can be shown in the general case that the x, y, and z components of $\vec{G}$ are

$$\left[\begin{array}{c|c} G_x^{(1)} & G_x^{(2)} \\ \hline -G_x^{(2)} & -G_x^{(1)} \end{array} \right], \quad \left[\begin{array}{c|c} G_y^{(1)} & G_y^{(2)} \\ \hline G_y^{(2)} & G_y^{(1)} \end{array} \right], \quad \left[\begin{array}{c|c} G_z^{(1)} & G_z^{(2)} \\ \hline -G_z^{(2)} & -G_z^{(1)} \end{array} \right]$$

where $\vec{G}^{(1)}$ contains the influence of vortices on the +y (or -y) side of the x-z symmetry plane at field points on the +y (or -y) side; $\vec{G}^{(2)}$ contains the influence of vortices on the +y (or -y) side at field points on the -y (or +y) side. The partitioned form of H is discussed in the following section.

3.2 DESCRIPTION OF PROGRAM B

The purpose of Program B is twofold: (1) to perform the inversion of the H matrix, and (2) to create a magnetic tape on which the matrices HDIAG, H^{-1} , H, and $\vec{G}$ are written. (HDIAG is a column matrix whose elements are formed from the principal diagonal of H.)

Prior to inversion, the elements of the H matrix are normalized by the elements of its principal diagonal. Thus the resulting matrix has the value of unity everywhere on the principal diagonal and is written in partitioned form (see Eq. (26) of Ref. 1):

$$H = \left[\begin{array}{c|c} H^{(1)} & H^{(2)} \\ \hline H^{(2)} & H^{(1)} \end{array} \right]$$

Likewise, it can be shown that the inverse of H is written in partitioned form (see Eq. (28) of Ref. 1):

$$H^{-1} = \left[\begin{array}{c|c} A^{(1)} & A^{(2)} \\ \hline A^{(2)} & A^{(1)} \end{array} \right]$$

The FORTRAN code written to perform the calculation of $A^{(1)}$ and $A^{(2)}$ has been devised with the objectives of (1) being able to invert the largest possible matrix within the limitations imposed by the size of the available core, and (2) minimizing the computer time required to execute the calculations. The salient feature of the algorithm that has been derived to satisfy these mutual requirements is that storage locations are reserved in-core to accommodate only one square matrix. The size of this matrix is NVORT rows x NVORT columns, where NVORT is the number of vortices in the network on one side of the x-z plane of symmetry of a planform. In addition, an option has been provided for segmenting the program into two separately executed submittals. This is to allow for those instances in which the size of the H matrix is so large that an excessive amount of computer time would be required to perform the inversion from beginning to end in one submittal.

In order to facilitate the reading of the computer code for Program B, a summary of the sequence of the intermediate calculations is presented for the special case where the entire inversion is performed in one submittal of the program:

Normalize $H^{(1)}$ by the elements of its principal diagonal (HDIAG)

WRITE(21) HDIAG

Compute $H^{(1)-1}$

Normalize $H^{(2)}$ by the elements of the principal diagonal of $H^{(1)}$

Compute $H^{(2)}H^{(1)-1}$

Compute $H^{(2)}H^{(1)-1}H^{(2)}$

Compute $H^{(1)} - H^{(2)}H^{(1)-1}H^{(2)}$

Compute $A^{(1)} \equiv [H^{(1)} - H^{(2)}H^{(1)-1}H^{(2)}]^{-1}$ and WRITE(21) $A^{(1)}$

Compute $A^{(2)} \equiv -A^{(1)}H^{(2)}H^{(1)-1}$ and WRITE(21) $A^{(2)}$

WRITE(21) $H^{(1)}$, $H^{(2)}$, $\vec{G}$

3.3 DESCRIPTION OF PROGRAM C

This program uses the output from Program B to compute aerodynamic characteristics and (as an option) separated store trajectories. The computer requirements in terms of internal memory are of such magnitude that it is necessary to subdivide the program into three separately compiled job steps. Each of these will be described in turn.

3.3.1 First Job Step

The purpose of this step is to transfer a subset of the data written on magnetic tape in Program B onto a direct-access device. The purpose of this operation is to achieve faster execution times in the third job step.

There are two options which may be exercised in this step. If it is desired to show by substitution (a posteriori in the third job step) that the computed values of vortex strengths satisfy the matrix equation for $\{\Gamma_j\}$, Eq. (9), then the H matrix is transferred from tape to disk; otherwise, it is not. The second option concerns the calculation of the pressure coefficient distribution (in the third step). If it is desired to perform this calculation, the influence coefficients at boundary points are transferred from tape to disk; otherwise, they are not.

3.3.2 Second Job Step

The primary functions of this step are as follows: (1) to read input data, which comprise values for parameters of the problem and the NUFF, from punched cards and to pass this data to the following job step, (2) to determine the coordinates of the line segments comprising the vortex network, and (3) to compute the locations of the control points at which the flow-field vector is constrained to be in a prescribed direction. (Items 2 and 3 are also computed in Program A.)

3.3.3 Third Job Step

The FORTRAN code for this step is composed of a MAIN program and subroutines VORLAT, FRESTR, VELOCITY, ACOEFF, and AXES. There are two optional versions of this MAIN program, depending upon whether or not it is desired to perform trajectory calculations. The subroutines are all identical for these two versions.

The primary purpose of the trajectory version of the MAIN program is to provide values for many of the parameters which enter into the equations of motion, and then to numerically integrate these equations. The other version has been coded for the purpose of defining values of variables used in calculations in the subroutines.

Vorticity and velocity distributions are computed in SUBROUTINE VORLAT. Values of Γ_j are given by the solution of Eq. (9) by matrix inversion, and the formula for velocity at the field point k is (see Eq. (15) of Ref. 1)

$$\vec{V}_k = \vec{U}_{\infty k} + \sum_{j=1}^{N_j} \vec{G}_{k,j} \Gamma_j \quad (15)$$

The purpose of SUBROUTINE FRESTR is to interpolate on experimental values of the NUFF to determine the downwash, sidewash, and magnitude of velocity at points on the surface of the store. Three sets of such interpolations are performed; the first set of interpolated values is obtained at the boundary points, the second at the midpoints of spanwise segments of the vortex network, and the third set at midpoints of chordwise segments.

No calculations are performed in SUBROUTINE VELOCITY. The sole purpose of this subroutine is to print (as an option) the distribution of velocities.

SUBROUTINE ACOEFF performs aerodynamic calculations. As an optional capability the pressure coefficient can be computed at each of the boundary points, using Eqs. (23) and (24) of Ref. 1 (see also Section IV-h of the Appendix to Ref. 3). The distribution of aerodynamic forces on a planform is determined using the Kutta-Joukowski law, Eq. (10), and these forces are then summed. The detailed equations to accomplish these results, which are incorporated in the computer code, are given in Eqs. (IV-a) through (IV-g) of Ref. 3.

The coordinate transformations relating components of variables in wind axes to the components in a body axis reference system utilize the coefficients computed in SUBROUTINE AXES. The formulas for these coefficients can be found on p. 31 of Ref. 4.

3.4 COMPUTING TIME

The vortex-lattice solutions reported herein were obtained on an IBM 370/155 digital computer whose Central Processing Unit (CPU) can store a source program of 45,000 decimal words using internal memory. Examples will be cited of computational times required to execute solutions. The time required varies with the number of vortices, since the execution time is a function of the size of the matrices containing the geometric factors and the number of elements in these matrices varies approximately as the number of vortices squared. A representative calculation assuming 156 vortices on each side of the plane of geometric symmetry and asymmetry of flow about the x-z plane, equivalent to a 312-vortex problem, requires 10 min to execute Program A, and 14 min to execute Program B.

The time required to execute Program C is virtually all consumed in performing the calculation of the force coefficients in the third job step of this program. Since a significant amount of this time is expended in transferring data from external memory into the core of the CPU, a procedure to optimize the calculations has been devised which consists of computing the force coefficients for a number of cases each time the data are read from external memory. (A case means one orientation of a store with respect to the parent aircraft; the external memory used is a direct access device.) The maximum number of cases that can be computed each time the external memory is read is limited by the amount of core available to store intermediate results. In the present application the programmed procedure is to read the direct access device, compute five cases (the number of cases is an input to Program C), rewind the direct access device, and repeat this sequence any desired number

of times. It is found that the execution of five cases requires 1 min 25 sec; therefore, each potential flow solution requires an average of 17 sec of CPU time.

The particular option of the numerical method devised to compute trajectories that has been exercised to produce the results of Section 4.5 requires five vortex-lattice cases to be computed for each advance of two time steps in the numerical integration of the equations of motion. Thus, for example, in the sample trajectories reported herein in which each of 16 time steps was specified to be 0.05 sec of real time, a trajectory of 0.8 sec total real time required 11 min of CPU time.

SECTION IV RESULTS OF CALCULATIONS

As shown in Fig. 1, the physical shape of the M-117 bomb body was represented by a system of horseshoe vortices having the "trailing" elements running forward to the nose tip, whence they became superimposed and trailed back downstream to infinity along the bomb axis. The fins were modeled by horseshoe vortices having trailing segments in the usual direction. To obtain a reasonable degree of agreement with experiment, it was found necessary to impose a wakelike character on the flow in the base region by extension of the vortex network downstream of the actual bomb body (Ref. 1). These wake vortices were used only to control the velocity pattern near the base of the bomb; forces on these vortices were not included as force on the bomb.

4.1 CORRECTIONS TO CALCULATED FORCE COEFFICIENTS

To ensure that the calculated forces represented the best approach to reality, three different corrections were added to the force coefficients resulting from the potential flow calculations.

4.1.1 Base Force Correction

As described in Ref. 1, summation of the Kutta-Joukowski forces over the vortex network (with wake vortices excluded from the summation) results in a large, unbalanced internal pressure force and a net thrust on the network. This occurs because the vorticity representing the solid base and, hence, the internal and external force on the base, is not accounted for. To allow for this neglected base force, a correction to

the calculated axial-force coefficient is given by

$$\Delta C_{A_{base}} = \frac{S_{base}}{S_{ref}} [C_{p_i} - C_{p_b}] \quad (16)$$

For the purpose of comparison with wind tunnel data, the usual reduction to zero base drag conditions, $C_{p_b} = 0$, is made. The pressure coefficients inside the vortex network in the base region were observed to vary with angle of pitch or yaw and were given approximately by the curve-fit:

$$C_{p_i} = 0.992 - 0.00018 \times (\nu \text{ or } \Psi, \text{ deg})^2 \quad (17)$$

The base-force correction can be seen to be very nearly the stagnation pressure acting over the base area. Lastly, the base area in this case was taken to be the area of the octagonal base of the vortex network, rather than the area of the circular base of the actual bomb shape. The resulting correction was thus given by

$$\Delta C_{A_{base}} = (0.495 \times 0.9) (0.992 - 0.00018) \times (\nu \text{ or } \Psi, \text{ deg})^2 \quad (18)$$

4.1.2 Compressibility Correction

After the addition of the base-force correction, all force coefficients calculated for incompressible flow over the (possibly contracted) body shape were corrected for compressibility by Eq. (12) or (13).

4.1.3 Skin-Friction Correction

Finally, an approximate allowance was made for contribution of skin friction to axial force:

$$\begin{aligned} \Delta C_{A_{fric}} &= 0.0316 \text{ for a laminar boundary layer} \\ &= 0.0935 \text{ for a turbulent boundary layer} \end{aligned} \quad (19)$$

These corrections were based on flat plate skin friction at $Re_x = 1.1 \times 10^6$, corresponding to the wind tunnel Reynolds number at $M = 0.5$. Since drag exerts only a minor influence on separation trajectories, high accuracy on C_A was not considered important, and the skin-friction corrections were simply applied as constants at all conditions.

4.2 CORRECTION OF EXPERIMENTAL FORCE DATA

Because of the marked sensitivity of base-region flows to Reynolds number, experimental axial-force data on blunt-base bodies are usually corrected to a standard condition of zero base drag, defined as the condition $p_b = p_\infty$ or $C_{p_b} = 0$. The standard correction expression for this condition is

$$\Delta C_A = C_{p_b} \frac{S_{base}}{S_{ref}} \quad (20)$$

Since base pressure measurements were not made during the test reported in Ref. 6, it was simply assumed that the base pressure was equal to pressure on the bomb just forward of the base, as given by the vortex-lattice calculations,

$$\Delta C_A = -0.065 \times 0.495 = -0.0322$$

4.3 COMPARISON OF CALCULATIONS AND MEASUREMENTS IN UNIFORM FLOW

Calculated force coefficients and measured force coefficients on the M-117 bomb in uniform flow are compared in Figs. 2 and 3. Both theory and experiment have been adjusted to as nearly the same basis of comparison as possible by use of the corrections discussed above.

In Fig. 2, the normal-force and pitching-moment coefficients are plotted versus angle of pitch. Theoretical curves are given for $M = 0$, 0.5, and 0.85. Experimental data are given for $M = 0.5$ and 0.85. A general agreement of about 10 percent exists at $M = 0.5$.

The effect of Mach number on the theoretical-experimental force coefficient comparison is given in Fig. 3. The calculated normal-force coefficient at $\nu = 10$ deg (Fig. 3a) displays a nearly constant 10-percent displacement from the measured values. The calculated pitching-moment coefficient at $\nu = 10$ deg (Fig. 3b) however, displays an appreciably greater influence of Mach number than does the experimental data. The calculated axial-force coefficient with laminar boundary layer at zero pitch (Fig. 3c) is 17 percent greater than the wind tunnel measurement at $M = 0.5$, but the theoretical drag rise given by Eq. (12) is much greater than observed in the wind tunnel data. At $M = 0.85$, the calculated axial-force coefficient, corrected for compressibility, is 3.5 times the measured value.

4.4 COMPARISON OF CALCULATIONS AND MEASUREMENTS IN F-4C FLOW FIELD

The force characteristics of the M-117 bomb were calculated for twelve different locations of the bomb in the disturbed flow field of the F-4C parent aircraft, three spanwise locations and four vertical locations (Fig. 4). The outboard pylon is shown in Fig. 4 in a vertical orientation, which corresponds to that of the full-scale aircraft when carrying an external fuel tank. The flow-field data of Ref. 6, however, on which the present results are based, were obtained for a model having the outboard pylon canted outboard at 7.5 degrees. The flow-field data used as boundary conditions were obtained from the test reported in Ref. 6. These calculations were similar to those reported in Ref. 1, with the following exceptions:

1. All three components of the nonuniform flow field, not just downwash angles, were used as boundary conditions on the bomb.
2. Both yaw and pitch orientations (but not in combination) were included.
3. Compressibility corrections to the potential flow calculations were included.
4. Bomb model was rolled to put fins in vertical and horizontal planes, corresponding to the configuration used in the wind tunnel.

4.4.1 Comparison at $M = 0.50$

The variation of normal-force and pitching-moment coefficients with pitch angle for $ZP = 3.13$ to 9.13 in. at $YP = -3.16$ in. and $M = 0.50$ is given in Fig. 5. The uniform flow force characteristics are also presented for comparison. In Fig. 6 are given C_N , C_m , C_Y , and C_n versus the vertical separation distance, ZP , at constant pitch angles, zero yaw, and $YP = -3.16$ in. The absolute values of the calculated coefficients are not in good agreement with the measured values, although the calculated force coefficients are in slightly better agreement than are the moment coefficients. The discrepancies are on the order of 40 to 50 percent, and the compressibility corrections do not result in any systematic reduction of the differences.

If, however, the incremental effects of the NUFF on the force coefficients are considered, there is clearly a better agreement between

the calculations and the measurements. In almost every case in Fig. 6, the shapes of the theoretical curves are quite similar to the shapes of the experimental curves. This fact is demonstrated for all the cases calculated in Figs. 7a through 1, in which the theoretical results have been arbitrarily adjusted to the experimental values at the maximum distance of separation from the parent aircraft, $ZP = 9.13$ in. These data cover all twelve locations of the bomb and the four force coefficients, C_N , C_m , C_Y , and C_n . Throughout all of these plots the trend of the individual coefficient with ZP is well predicted by the potential flow calculations. The largest single discrepancy is 26 percent, in C_m at $YP = -3.16$, $ZP = 3.13$, and $\nu = 2$ deg, but the average discrepancy is from 4 to 6 percent (Fig. 8). Variation of average discrepancy with vertical separation distance is also shown in Fig. 8. The increasing discrepancy at small separations is probably the result of mutual interference, which is not accounted for in the potential flow calculations. By extrapolation, it is estimated that the average discrepancy does not exceed 10 percent until the separation distance is about one-half the bomb diameter (0.4 in.). The compressibility correction appears to be slightly beneficial at large separation, but it becomes of negative value in the mutual interference region.

This result, that incremental behavior of force coefficients is well predicted by calculations, suggests the possibility of a combined theoretical/experimental approach in which experimental force coefficients measured at some convenient location away from a parent aircraft are adjusted by means of potential flow results. Ideally, the experimental force coefficients would be measured only in the free stream. In the present case, values measured at $ZP = 9.13$ in. are used as a reference because of an unexplained ambiguity between the uniform flow and the NUFF measurements. This ambiguity is manifested as a failure of the experimental data to fair smoothly and asymptotically to the measured uniform flow values of the force coefficients, e.g., C_N at $\nu = -4$ deg, $YP = -3.16$ in. (Fig. 6a) and C_m at $\nu = -4$ deg, $YP = -3.16$ in. (Fig. 6b). The calculated results from the NUFF, on the other hand, can be faired into the potential flow results in uniform flow with a minimum effort.

4.4.2 Comparison at $M = 0.85$

As might be expected from results in uniform flow (Fig. 3), the discrepancies in absolute values of calculated and experimental force coefficients were found to be even greater at $M = 0.85$ than at $M = 0.5$. On the incremental basis, however, the results are much closer to the $M = 0.5$ results. Instead of the complete incremental comparison of

calculated and measured coefficients given in Fig. 7 for $M = 0.5$, in Fig. 9 only two representative comparisons are given at $M = 0.85$. The best incremental comparison is for C_N versus ZP at $YP = -3.16$ in. (Fig. 9a), and the worst comparison is for C_m versus ZP at $YP = -6.16$ in. (Fig. 9b). A summary of the average discrepancy between theory and experiment at $M = 0.85$ is given in Fig. 10. At the higher Mach number the NUFF has a greater effect on the moment coefficients than on the force coefficients. The average discrepancies are 4 percent for force coefficients and 8 percent for moment coefficients. Also, the variation of the average discrepancies with the separation distance shows that sensitivity to mutual interference near the aircraft is much greater at the higher than at the lower Mach number. Discrepancies reach 10 percent at about 2 diameter separations for force coefficients and at about 3.5 diameters for moment coefficients. At the separation at which there is a 10-percent discrepancy in force coefficient, there is a 20-percent discrepancy in moment coefficient. The compressibility correction ranges from slightly beneficial to slightly non-beneficial.

4.5 TYPICAL STORE SEPARATION TRAJECTORIES

In order to demonstrate the viability of the trajectory/potential flow routine of Program C and to determine time of computation, calculations were performed for several typical separation trajectories. The solutions were obtained for a 250-lbm M-117 bomb launched at a 5000-ft altitude at $M = 0.5$ from Station No. 2 of an otherwise empty Triple Ejection Rack (TER) located on the left inboard pylon of an F-4C (Fig. 4). The bomb was assumed to be at a pitch attitude of -0.7 deg with respect to the free stream upon release from the carriage. Pitch and yaw damping rates were both set equal to -2.319 per radian (assuming the definitions of each rate to be based upon body length). Roll damping was neglected. The initial trajectory conditions resulted from an assumed ejector force of 1000 lb applied through a stroke of 0.2552 ft at 45 deg from the vertical in a direction passing through the center of gravity of the bomb. The trajectories shown in Fig. 11 commence at the end of the ejector stroke, i. e., at the instant the store separates from the TER. The translational displacements are computed with respect to a wind axes coordinate system where the origin is located at the store center of gravity at the instant of separation; the rotational displacements are with respect to the store body axes reference system.

In order to indicate the possible magnitude of perturbations to store motion caused by the nonuniform flow field, results of trajectory calculations for a store assumed ejected into both uniform and nonuniform flow fields are shown. For convenience, the nonuniform flow calculations were performed using absolute values of force coefficients obtained from the potential flow calculations, rather than by use of the incremental approach suggested in Section 4.4.1. This approach was considered compatible with the limited intent of demonstration of the program capability, independent of any considerations of accuracy. The most notable differences caused by flow nonuniformity are in the angular motion; the most significant effect on linear translation is in the x-direction.

The amplitudes of the pitch and yaw oscillations attenuate; the roll amplitude (not shown in Fig. 11) increases linearly with time in the absence of roll damping. It may be seen that even in a uniform flow field the instantaneous pitch and yaw motions occur about average amplitudes which are nonzero. The principal reason for this is that the existence of finite velocity components of the store in the Y- and Z-directions gives rise to effective angles of pitch and yaw different from the initial values of these angles. Hence, it is expected that these effective angles will tend to be mean values of the rotational excursions.

SECTION V SUMMARY

Results of calculations may be summarized as follows:

1. A potential flow computer program based on vortex singularities, capable of accepting nonuniform flow boundary conditions but previously restricted to flows having a plane of symmetry, was modified to eliminate this restriction. The added capability did not alter any of the basic mathematics of the program but did effectively double the size of the system of simultaneous equations representing the flow for any given body.
2. In order to perform the expanded calculations within the limits of computer internal memory and desired computer execution time, the program was reorganized into three separate programs. The first two of these programs contain purely geometrical calculations and require execution only one time for a given

body geometry and Mach number. The third program performs calculations of vorticity distribution, pressure coefficient distribution, and force coefficients on the body and is executed one time for each set of boundary conditions (i. e., orientation and location in a nonuniform flow field). Once the initial geometric calculations are completed, potential flow solutions are generated by the third program in an average computer time of 17 sec for a vortex network composed of 312 horseshoe vortices.

3. The first program performs mainly the calculation of influence coefficients, and performs the computations in 10 min for a network of 312 vortices.
4. The second program performs mainly the inversion of the 312 rows by 312 columns coefficient matrix necessary to the potential flow solutions. Since the vortex network possessed geometric symmetry, even though in general there was no flow symmetry, the geometric matrices were partitioned, and the inversion then involved processing of only 156 rows by 156 columns matrices. A computer routine was devised which performs the inversion in 14 minutes.
5. In addition to reorganization of the structure of the program, additions were made to account for a proper base force on the vortex network, a linearized subsonic compressibility correction, and a skin-friction contribution. With these corrections, the force characteristics computed for a 312-vortex model of the M-117 bomb in a uniform flow were found to agree with wind tunnel measurements (except for drag) to within 10 percent at a Mach number of 0.5. To achieve this result, it was necessary to include in the vortex modeling of the bomb a number of vortices having the sole purpose of imposing a wakelike flow at the base of the bomb.
6. The force characteristics on the M-117 bomb in pitch and yaw were calculated by the potential flow program at 12 different locations in the disturbed flow field under an F-4C aircraft having an empty TER on the inboard pylon and a 370-gal external fuel tank on the outboard pylon. The absolute values of these force

coefficients were observed to differ from comparable wind tunnel measurements by up to 50 percent at a Mach number of 0.5 and by greater amounts at a Mach number of 0.85.

7. On the other hand, the incremental variations of the calculated coefficients in the nonuniform flow field were found to be in much closer agreement with measurements. The average discrepancy between calculated and measured incremental variations was from 4 to 6 percent at a Mach number of 0.5. At a Mach number of 0.85, the average discrepancy was from 4 to 6 percent for force coefficients and from 8 to 10 percent for moment coefficients.
8. Discrepancies significantly greater than the average were observed at small separations of the bomb and aircraft, apparently because of neglect of mutual interference in the potential flow program. At a Mach number of 0.5, a 10-percent discrepancy was reached at a separation distance of $1/2$ the bomb diameter, while at a Mach number of 0.85, the 10-percent discrepancy occurred at two diameters for force coefficients and 3.5 diameters for moment coefficients.
9. In view of the relatively accurate predictions of incremental force coefficient behavior (as opposed to absolute values), it was apparent that a hybrid method of force coefficient determination in a disturbed flow field was possible. Force coefficients measured on a store at a limited number of points far from a parent aircraft could apparently be incremented at other points in the disturbed flow field by reference to potential flow solutions.
10. The linearized compressibility correction was found to have very little influence in determination of the incremental variation of force coefficients.
11. A six-degree-of-freedom trajectory routine was appended to the third program, using the same basic equations of motion as in the AEDC/PWT CTS. In order to minimize the number of time steps required for a given accuracy of trajectory (and, hence, the

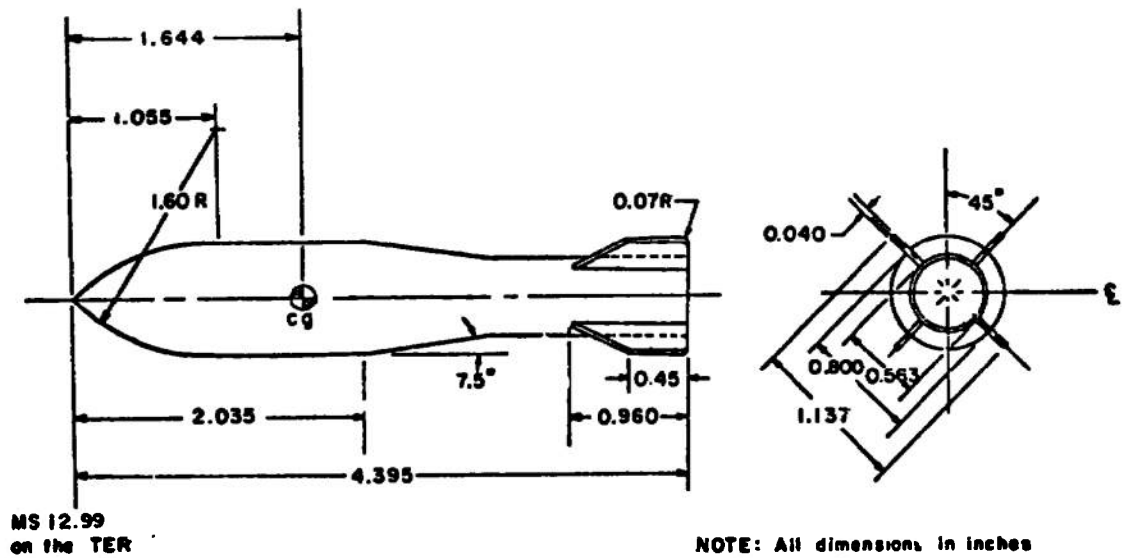
number of potential flow solutions), a more sophisticated numerical integration procedure (fourth-order Runge-Kutta) was used. A limited number of M-117/F-4C trajectories were computed with non-uniform and uniform flow force coefficients, respectively, to display the magnitude of effects of the disturbed flow field.

REFERENCES

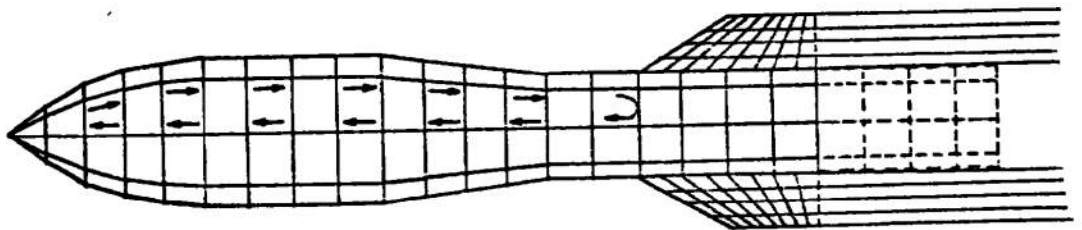
1. MacDermott, W.N. and Johnson, P.W. "Calculation for Forces on Aircraft Stores Located in Disturbed Flow Fields for Application in Store Separation Prediction." AEDC-TR-71-186 (AD733325), November 1971.
2. Ward, G.N. Linearized Theory of Steady High-Speed Flow. Cambridge University Press, Cambridge, 1955.
3. Rubbert, Paul E. "Theoretical Characteristics of Arbitrary Wings by a Non-Planar Vortex Lattice Method." Boeing Company, Document No. D6-9244, February 1964.
4. Christopher, J.P. and Carleton, W.E. "Captive-Trajectory Store-Separation System of the AEDC-PWT 4-Foot Transonic Tunnel." AEDC-TR-68-200 (AD839743), September 1968.
5. Thelander, J. A. "Aircraft Motion Analysis." FDL-TDR-64-70 (AD617354), March 1965.
6. Davis, Ronald E. "Flow Field Characteristics Beneath the F-4C Aircraft at Mach Numbers 0.50 and 0.85." AEDC-TR-70-8 (AD864702), February 1970.

APPENDIXES

- I. ILLUSTRATIONS**
- II. USER'S GUIDE TO COMPUTER PROGRAM**



a. Dimensional Sketch of 1/20-Scale M-117 Bomb Model



156-Vortex Model

b. 312 Vortex Model for Approximation of M-117 Bomb
Fig. 1 Shape of M-117 Bomb

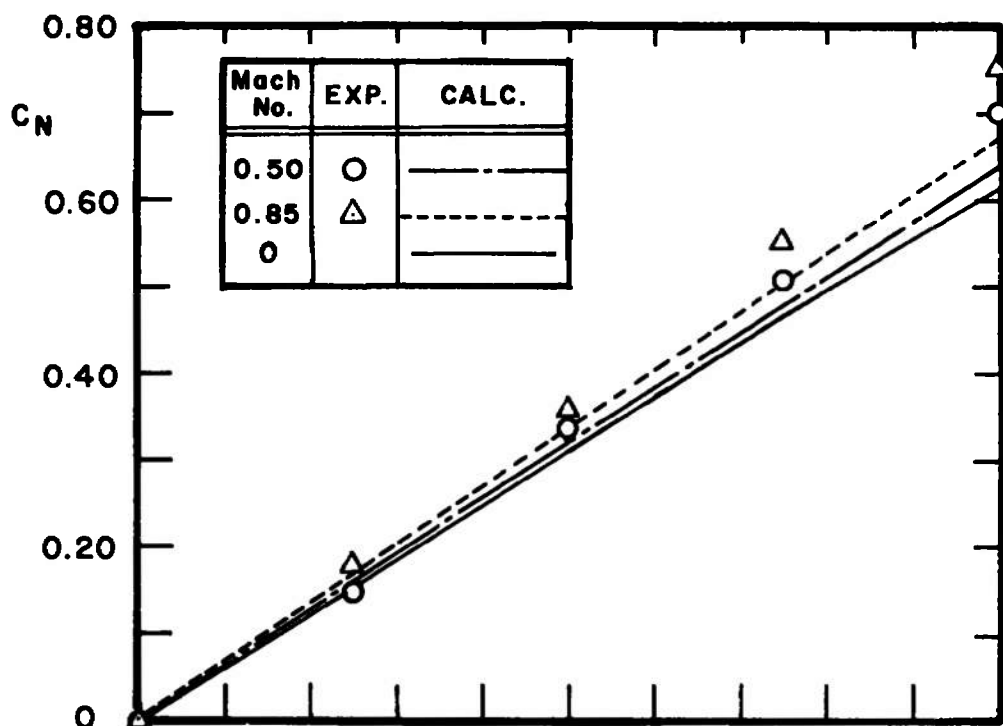
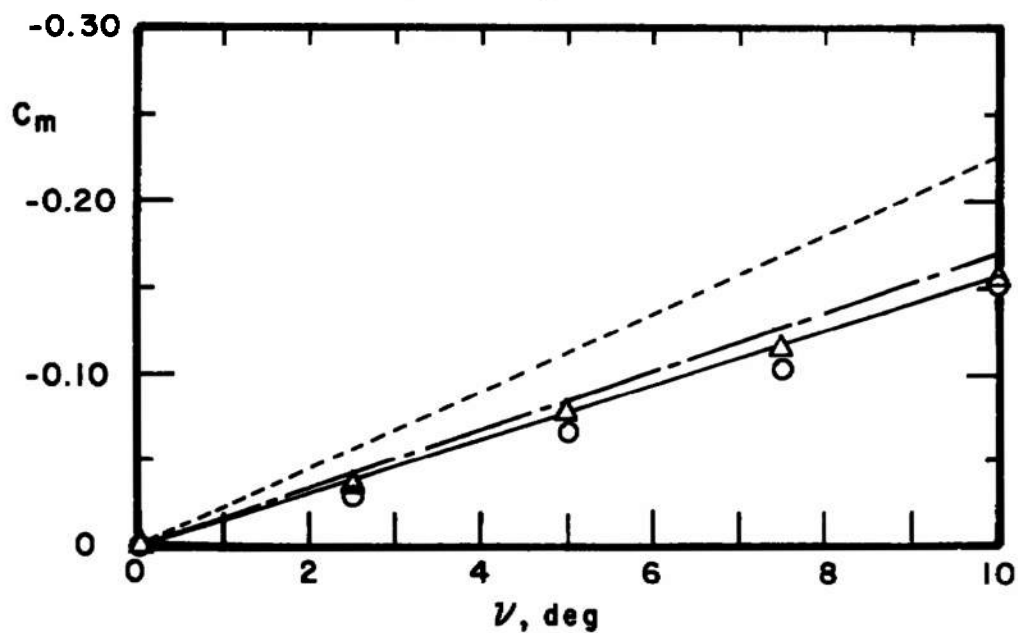
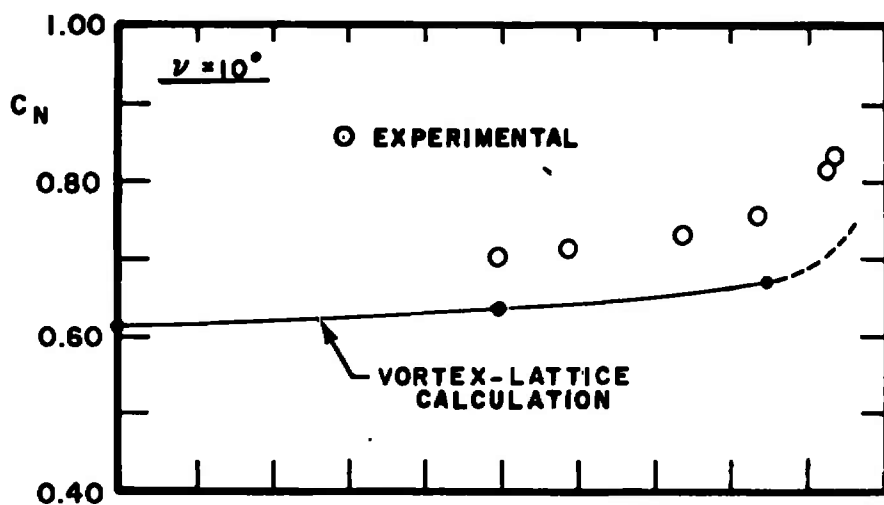
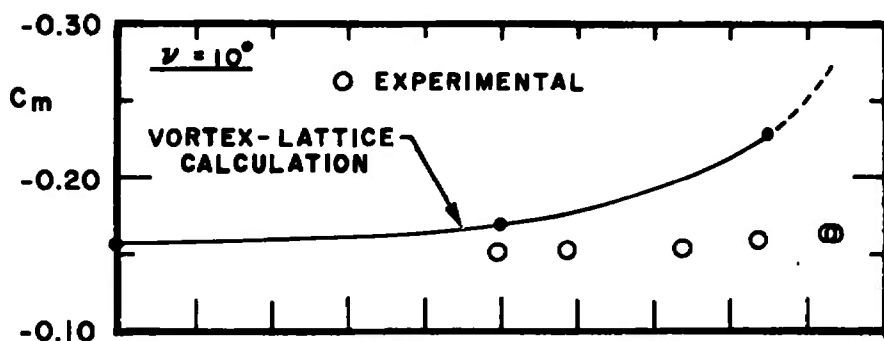
a. C_N versus Angle of Pitchb. C_m versus Angle of Pitch

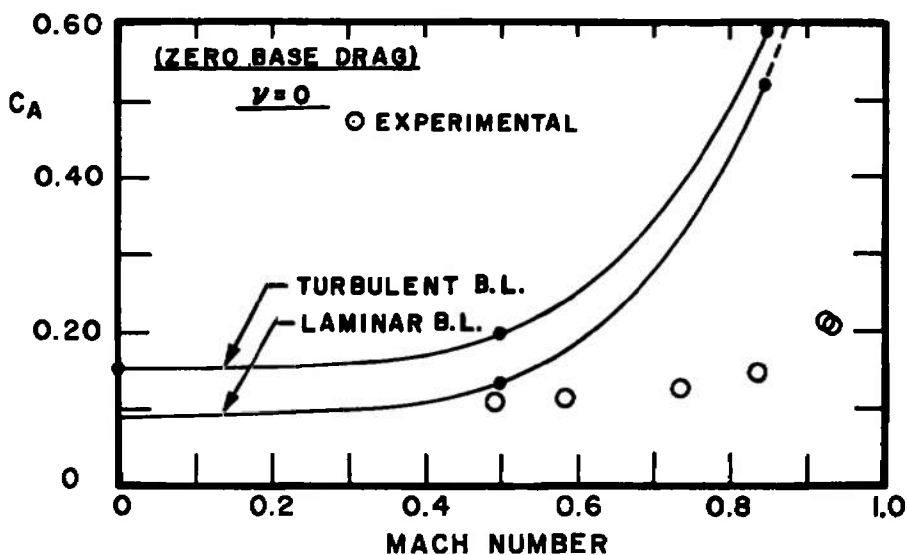
Fig. 2 Comparison of Calculated and Experimental Force Coefficients on M-117 Bomb, Uniform Flow



a. Calculated Normal-Force Coefficient



b. Calculated Pitching-Moment Coefficient



c. Calculated Axial-Force Coefficient

Fig. 3 Effect of Mach Number on Calculated and Experimental Force Coefficients, Uniform Flow

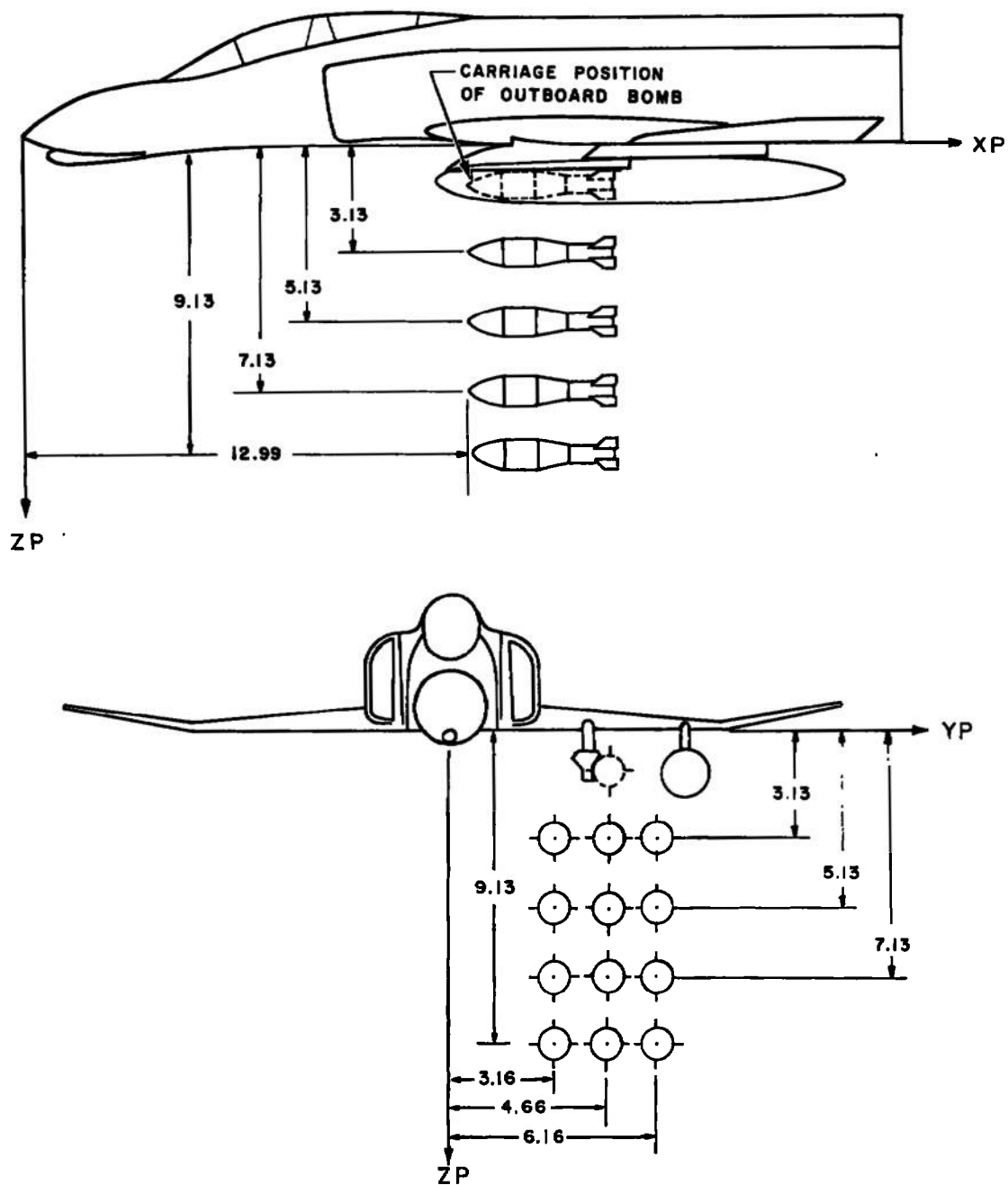


Fig. 4 Locations of Bomb under F-4C Parent Aircraft at Which Calculated and Measured Forces Were Compared

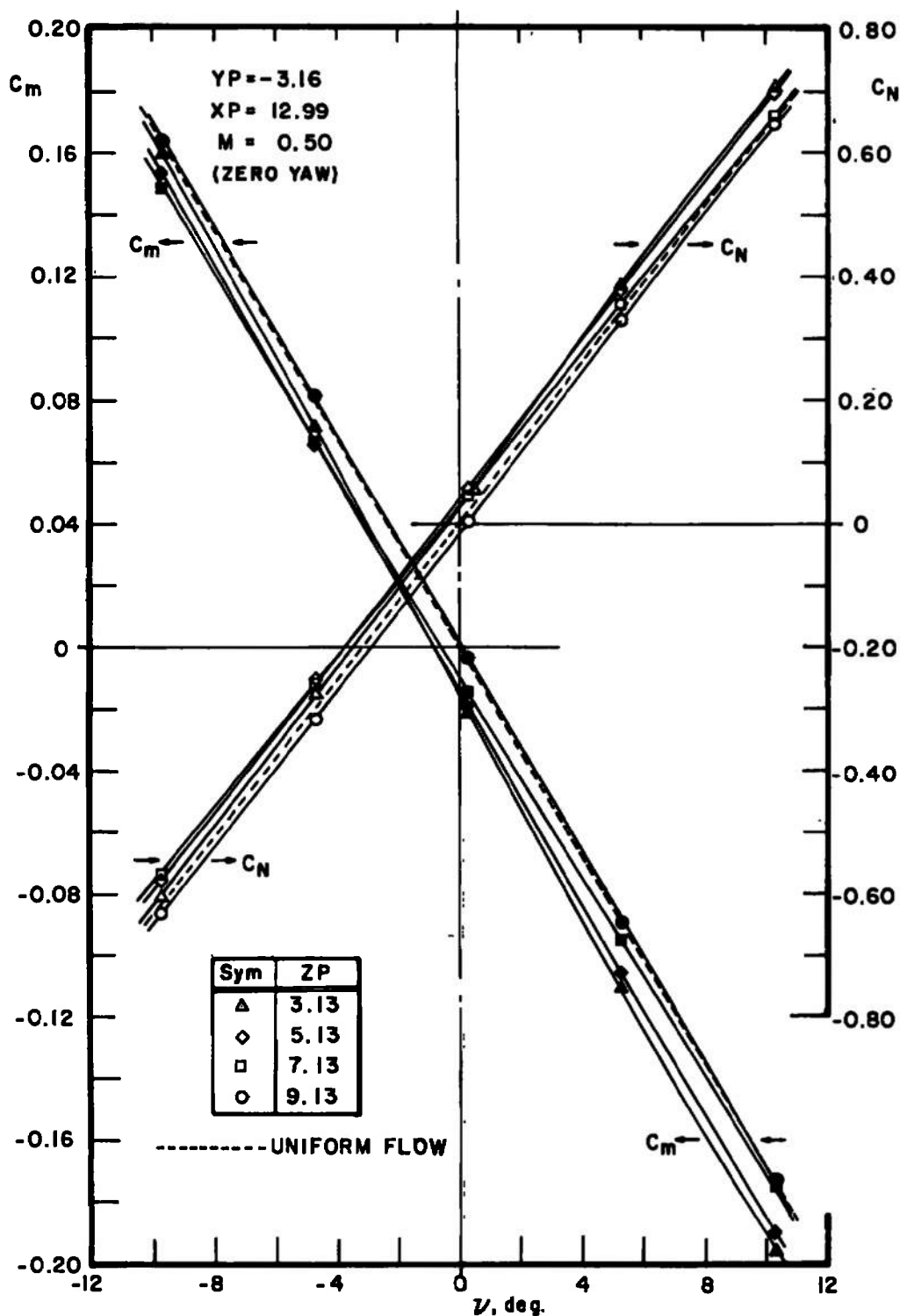


Fig. 5 Calculated Variation of C_N and C_m with Pitch Angle in F-4C Flow Field, YP = -3.16, XP = 12.99, M = 0.50

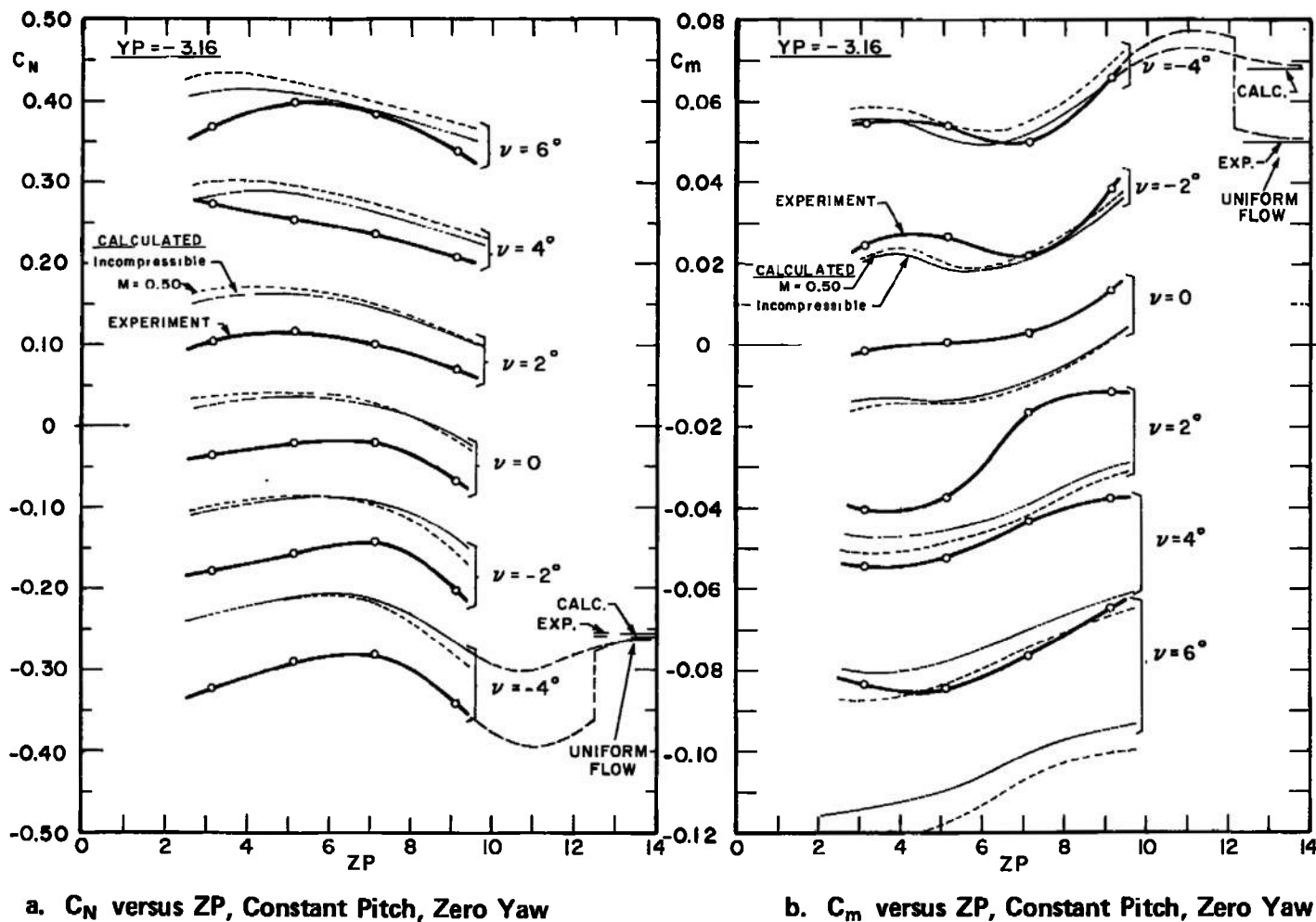


Fig. 6 Comparison of Absolute Values of Calculated and Experimental Force Coefficients in F-4C Flow Field, $M = 0.5$, $YP = -3.16$

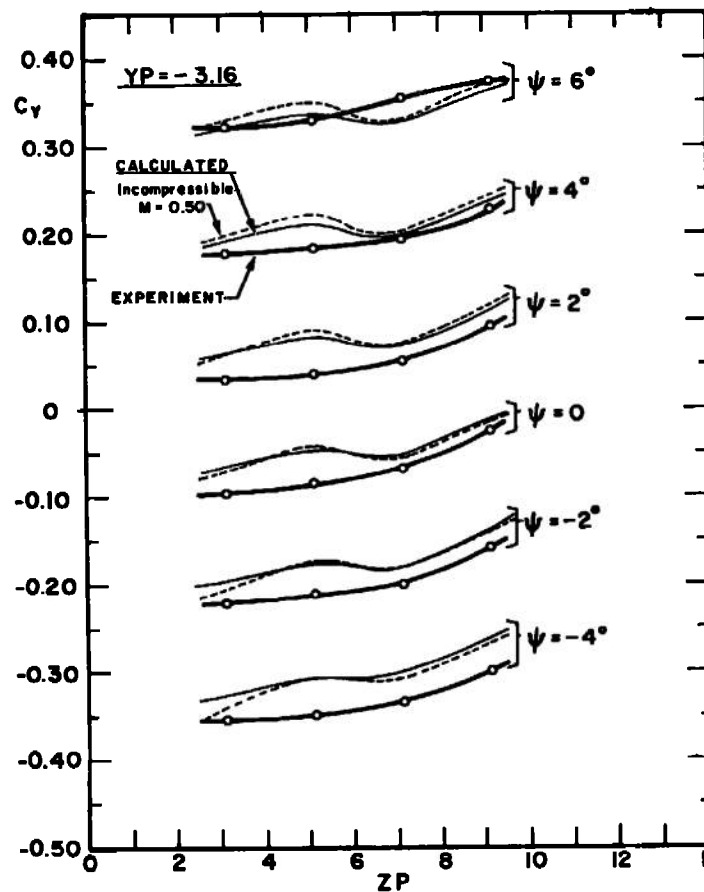
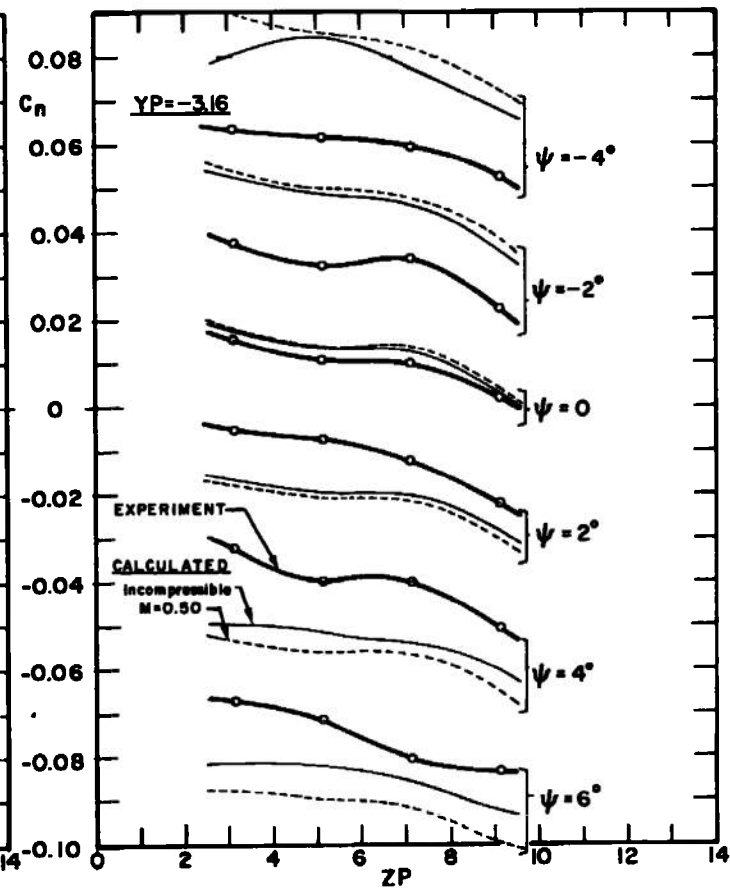
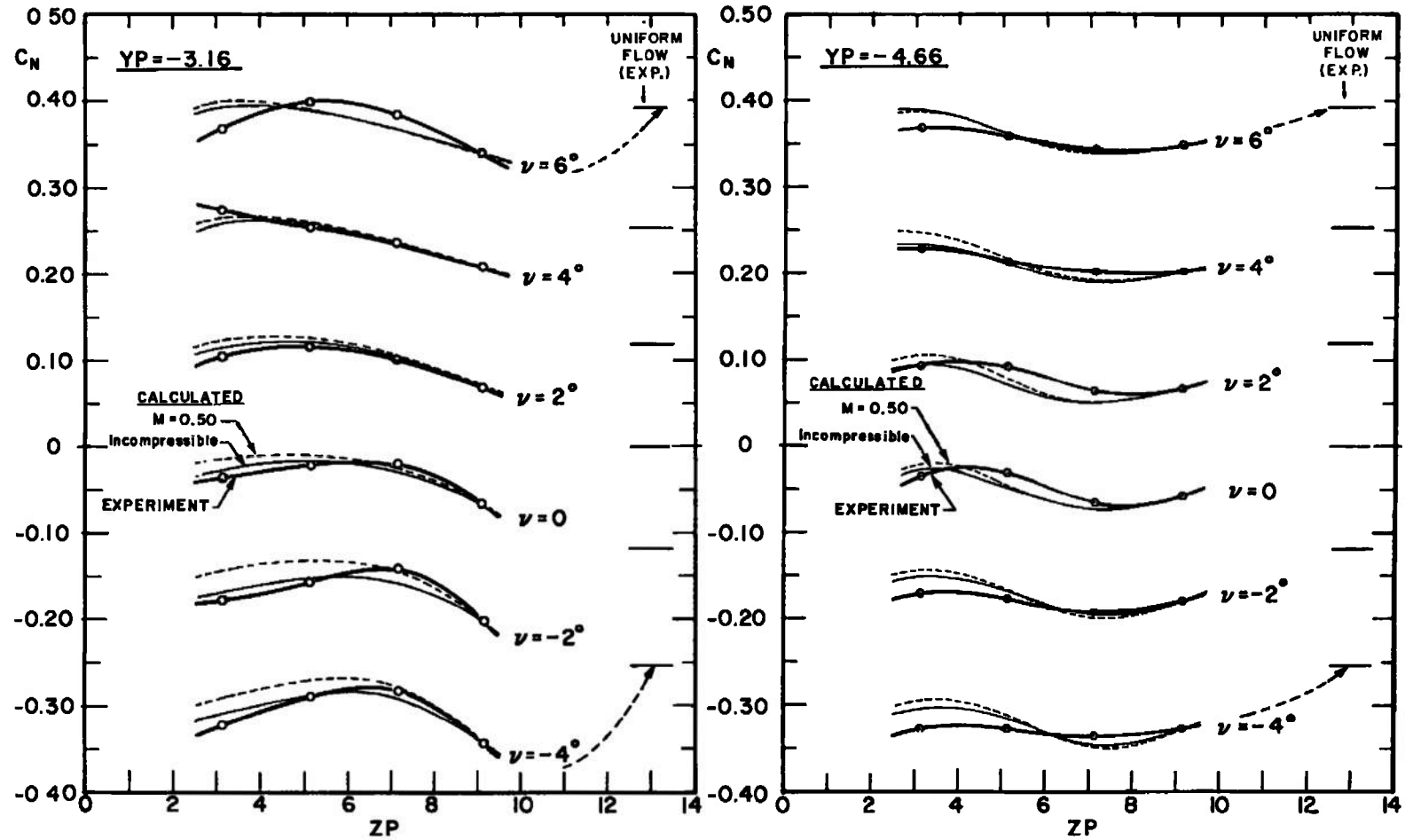
c. C_Y versus ZP, Constant Yaw, Zero Pitchd. C_N versus ZP, Constant Yaw, Zero Pitch

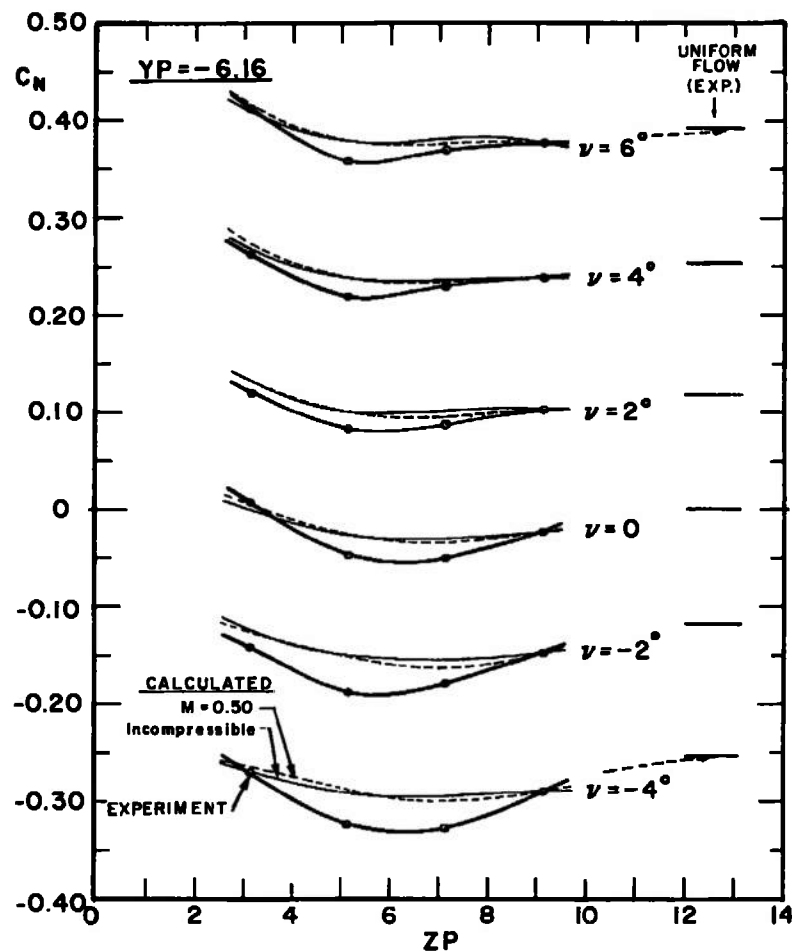
Fig. 6 Concluded



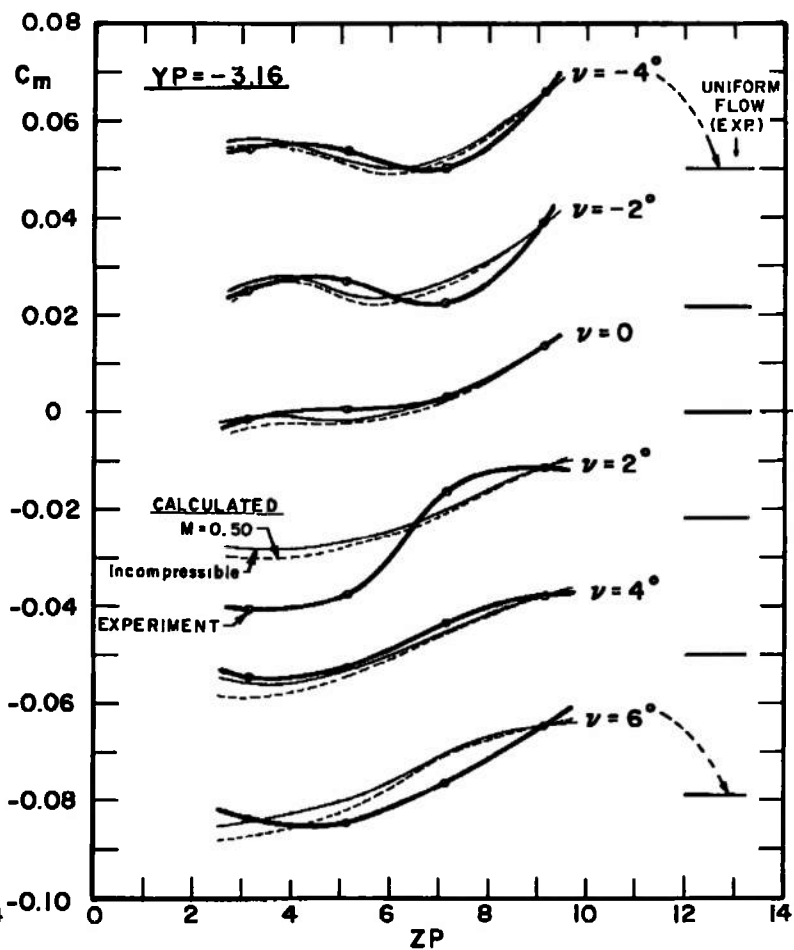
a. C_N versus ZP , Constant Pitch,
Zero Yaw, $YP = -3.16$

b. C_N versus ZP , Constant Pitch,
Zero Yaw, $YP = -4.66$

Fig. 7 Comparison of Incremental Variation of Calculated and Experimental Force Coefficients
in F-4C Flow Field, $M = 0.5$

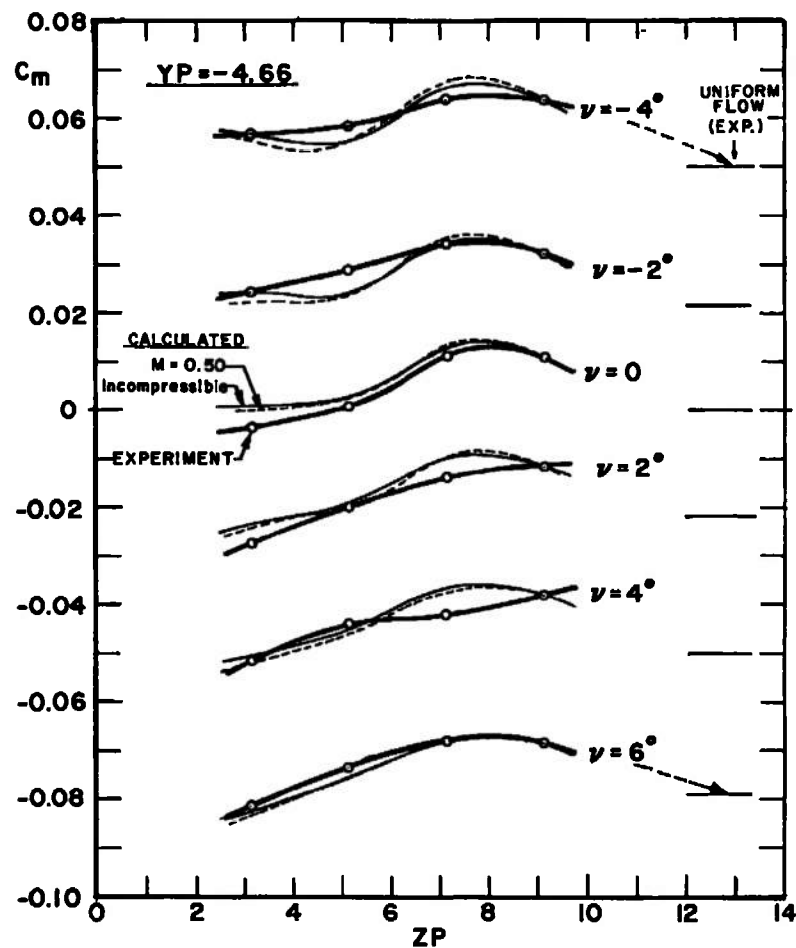


c. C_N versus ZP, Constant Pitch,
Zero Yaw, $Y_P = -6.16$

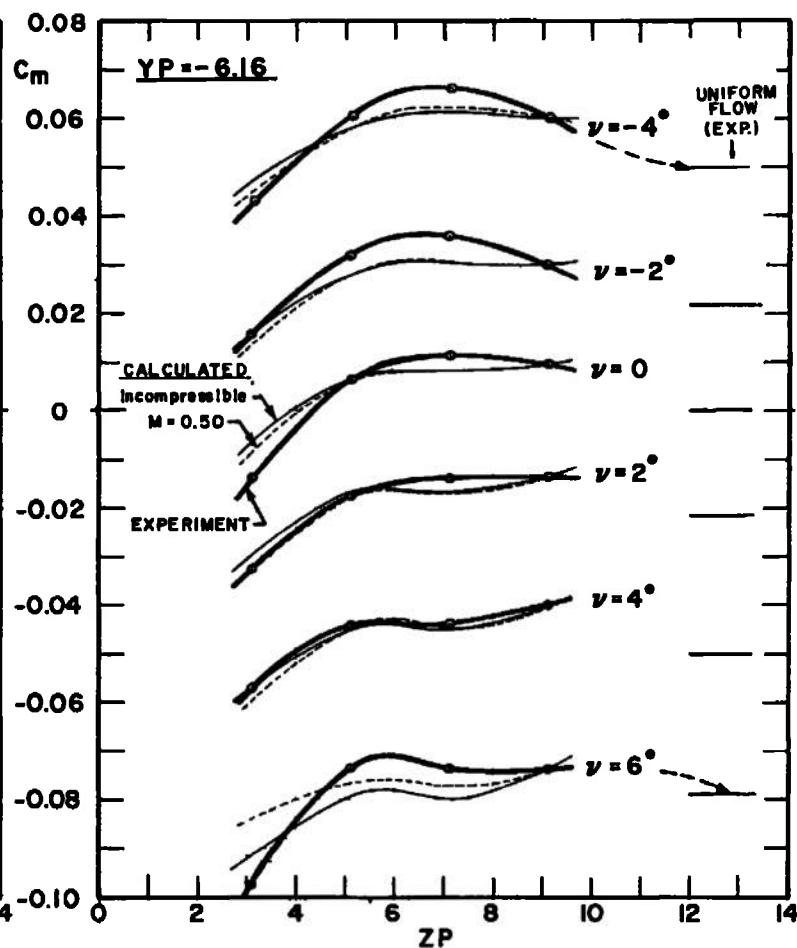


d. C_m versus ZP, Constant Pitch,
Zero Yaw, $Y_P = -3.16$

Fig. 7 Continued

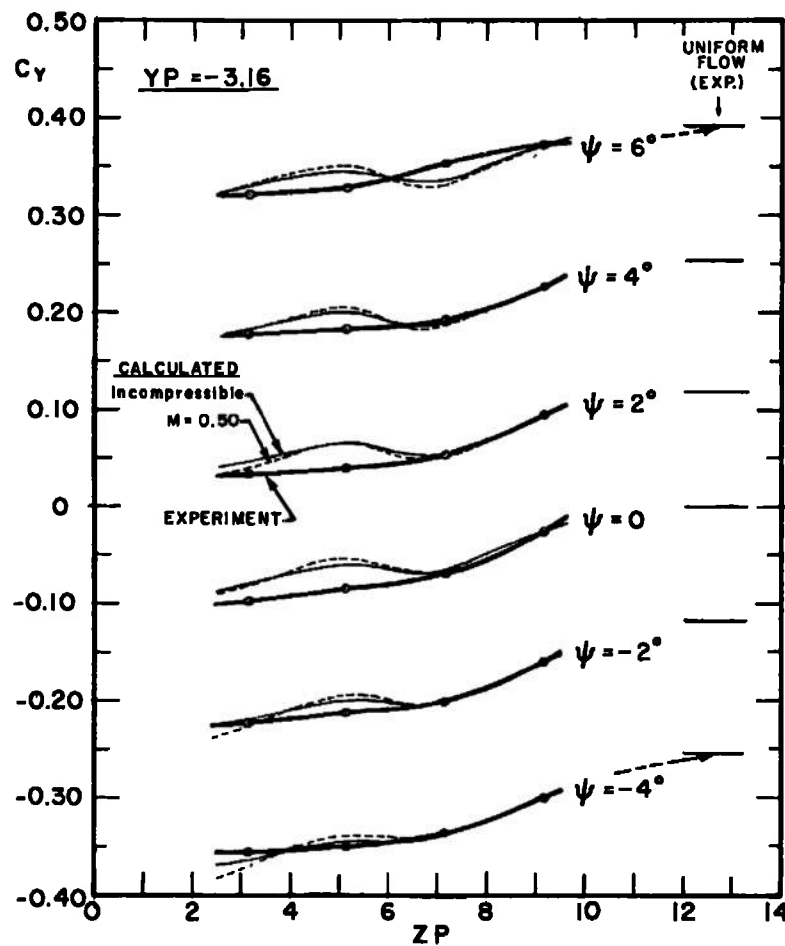


e. C_m versus ZP , Constant Pitch,
Zero Yaw, $YP = -4.66$

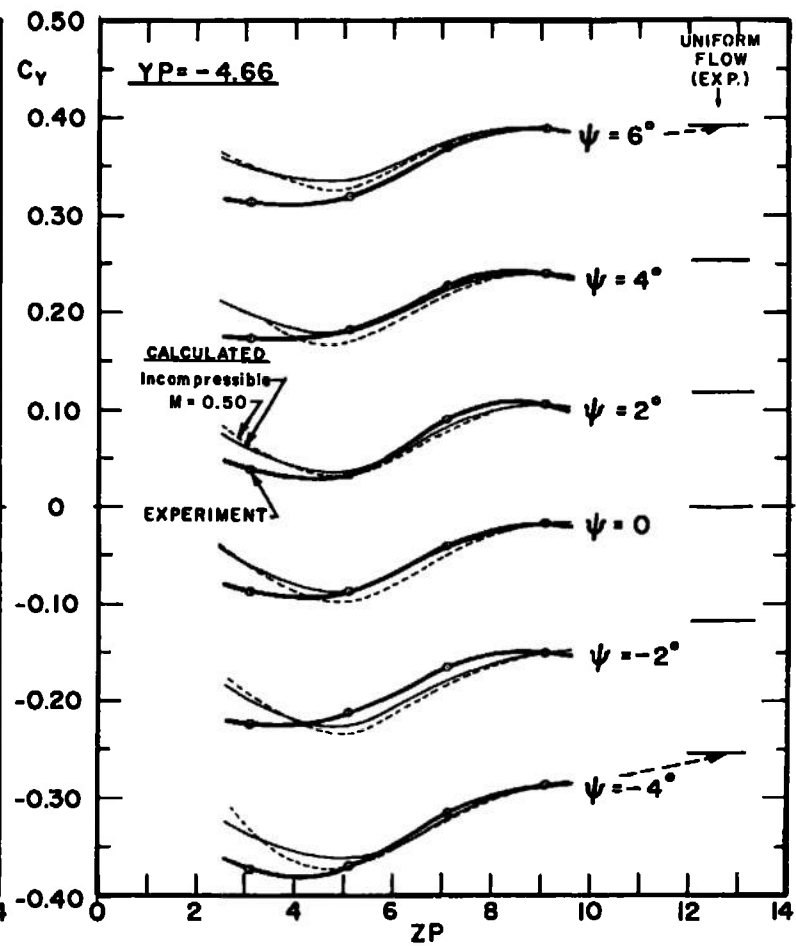


f. C_m versus ZP , Constant Pitch,
Zero Yaw, $YP = -6.16$

Fig. 7 Continued

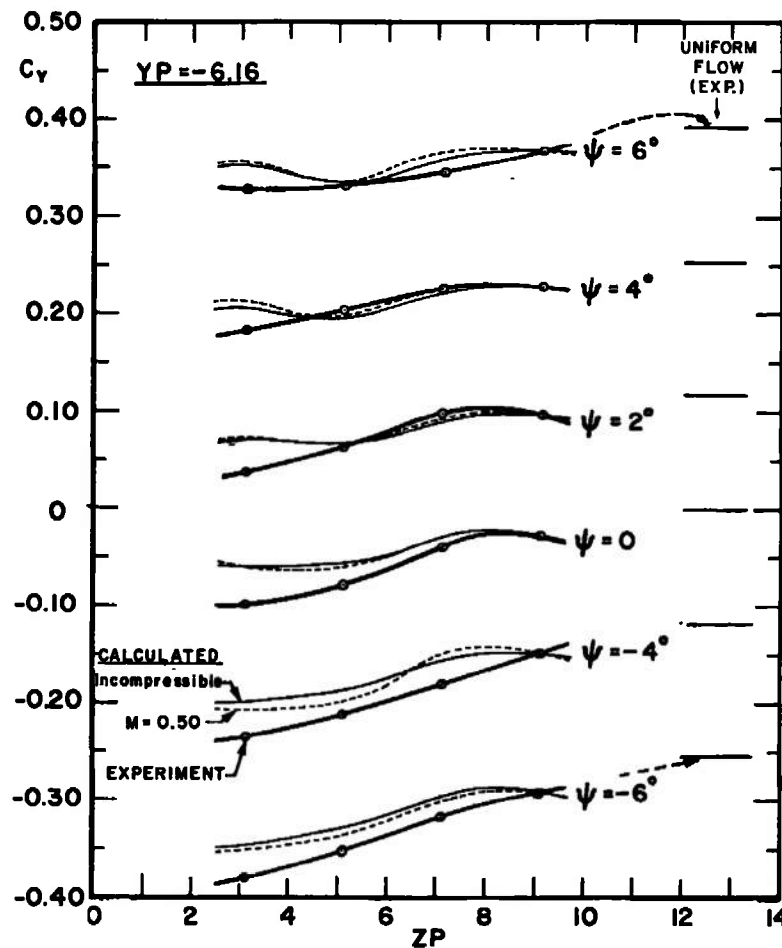


g. C_Y versus Z_P , Constant Yaw,
Zero Pitch, $Y_P = -3.16$

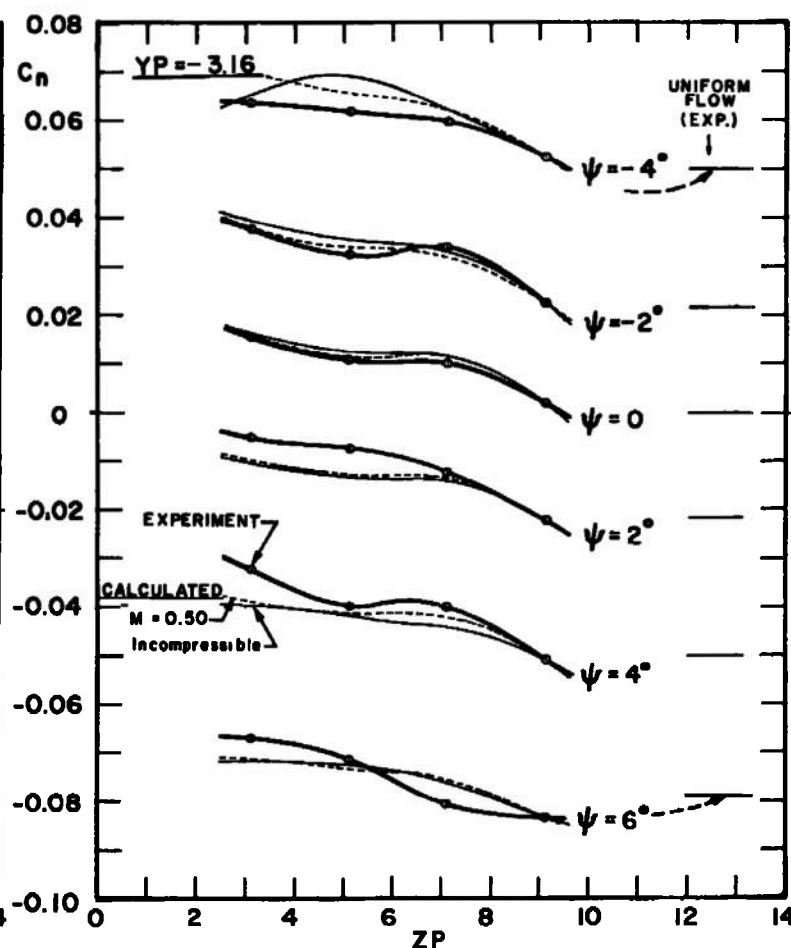


h. C_Y versus Z_P , Constant Yaw,
Zero Pitch, $Y_P = -4.66$

Fig. 7 Continued

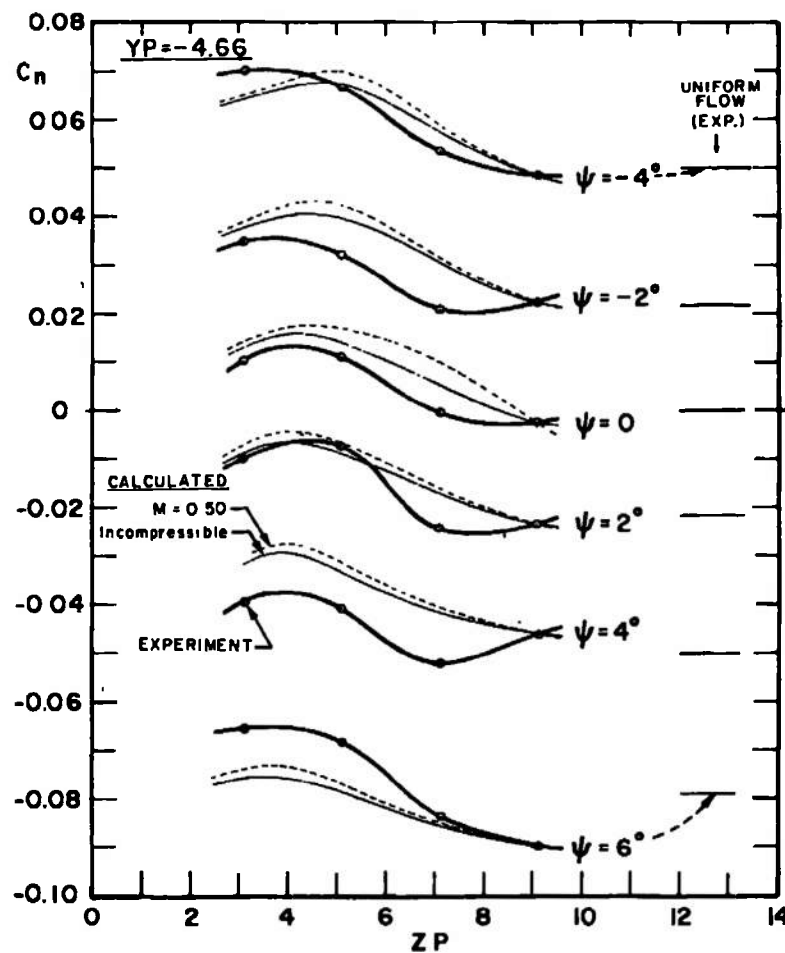


i. C_Y versus ZP , Constant Yaw,
Zero Pitch, $YP = -6.16$

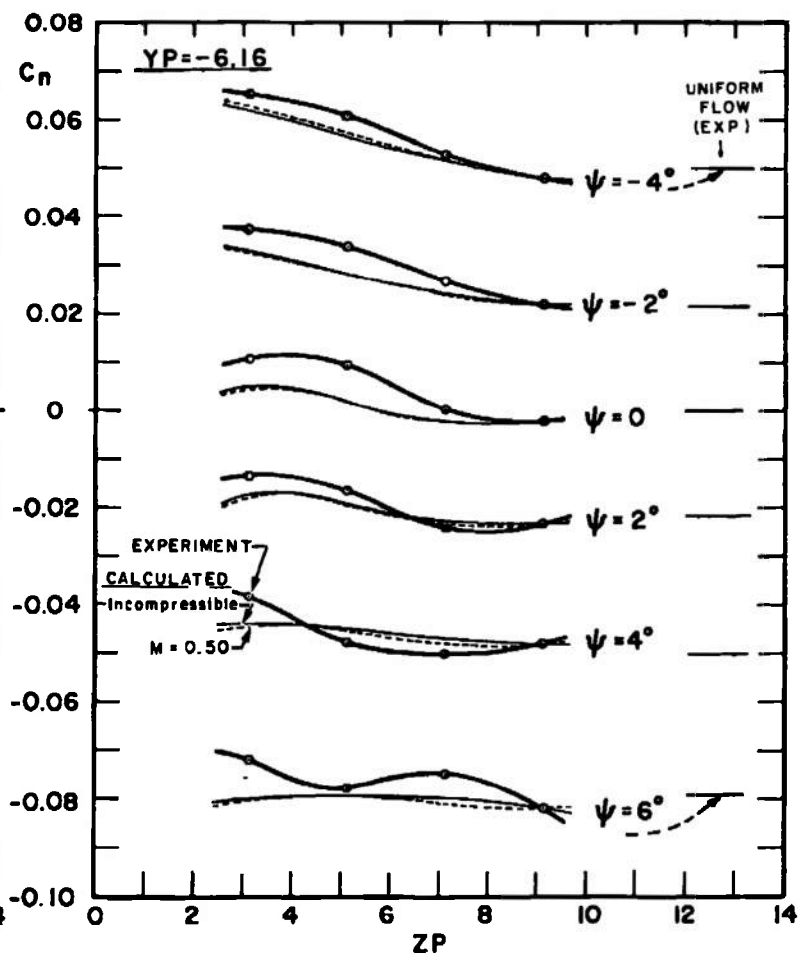


j. C_n versus ZP , Constant Yaw,
Zero Pitch, $YP = -3.16$

Fig. 7 Continued

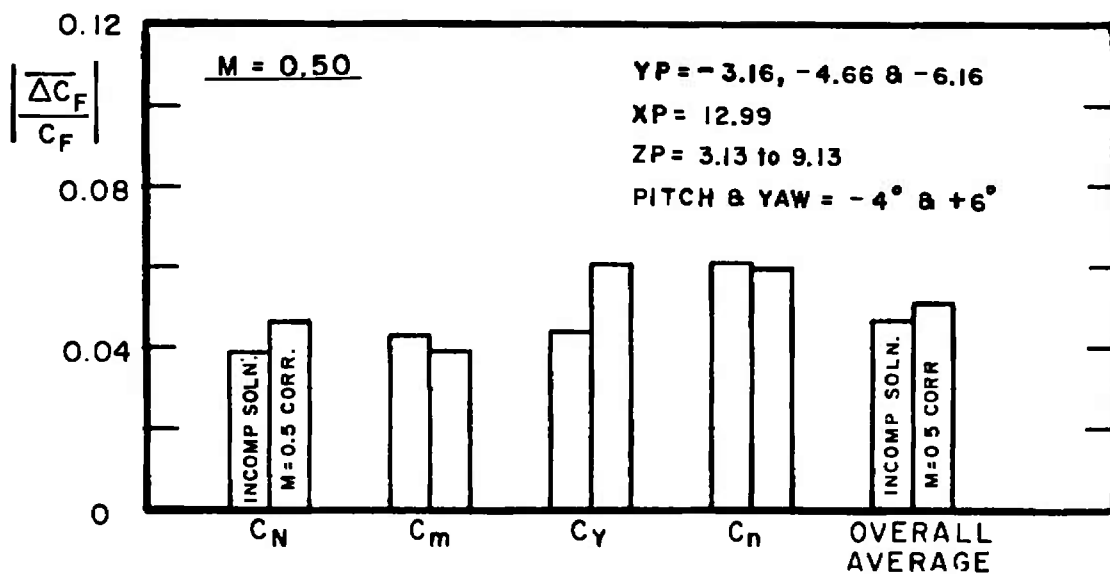


k. C_n versus ZP, Constant Yaw,
Zero Pitch, YP = -4.66

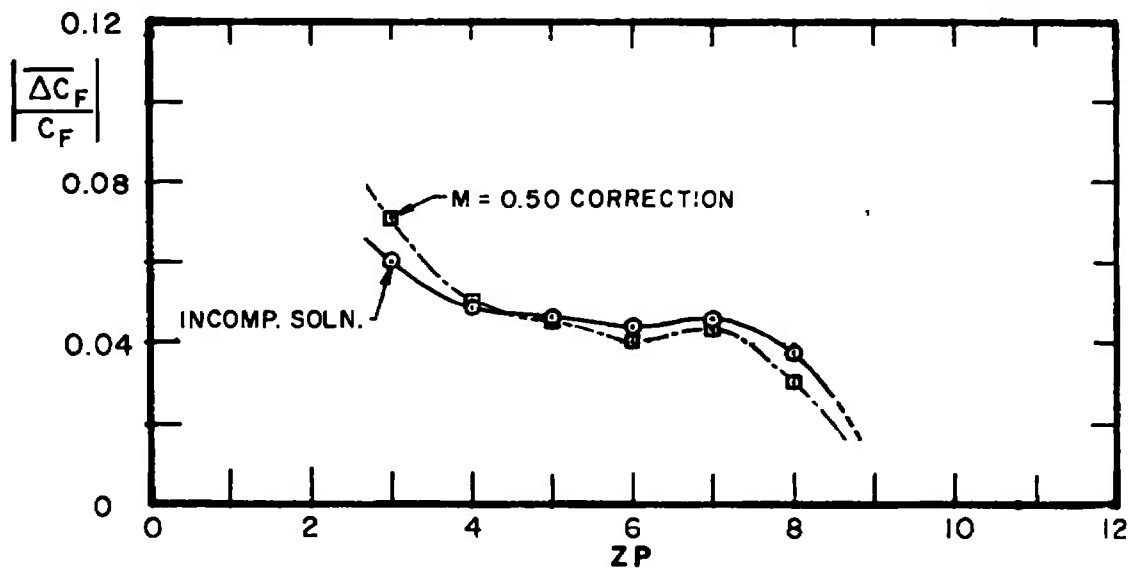


l. C_n versus ZP, Constant Yaw,
Zero Pitch, YP = -6.16

Fig. 7 Concluded

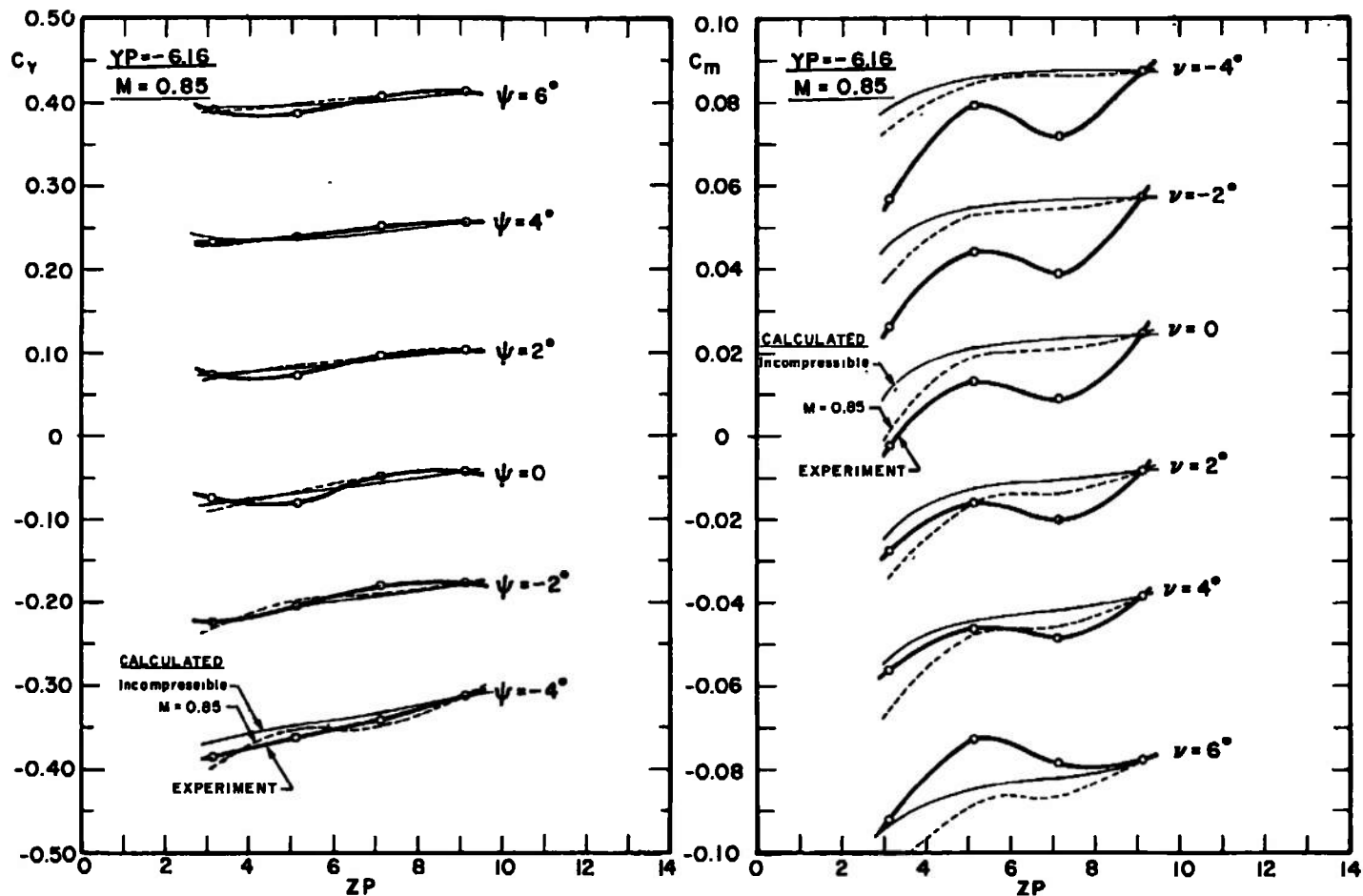


a. Average Discrepancy in C_N , C_m , C_Y , C_n , and Overall



b. Variation of Average Theoretical/Experimental Discrepancy with ZP

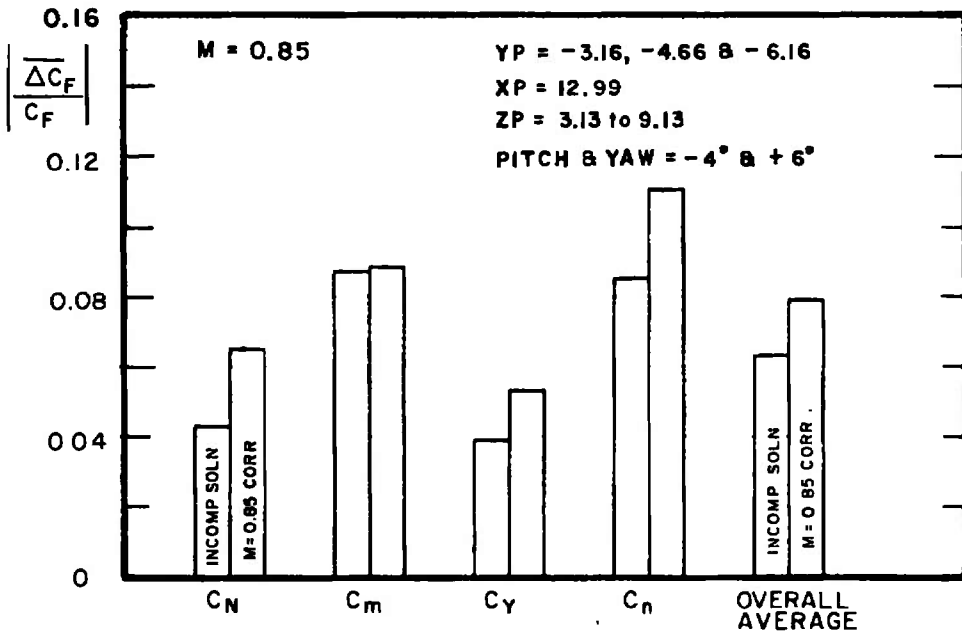
Fig. 8 Average Discrepancy Between Calculated and Experimental Force Coefficient, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, M = 0.5



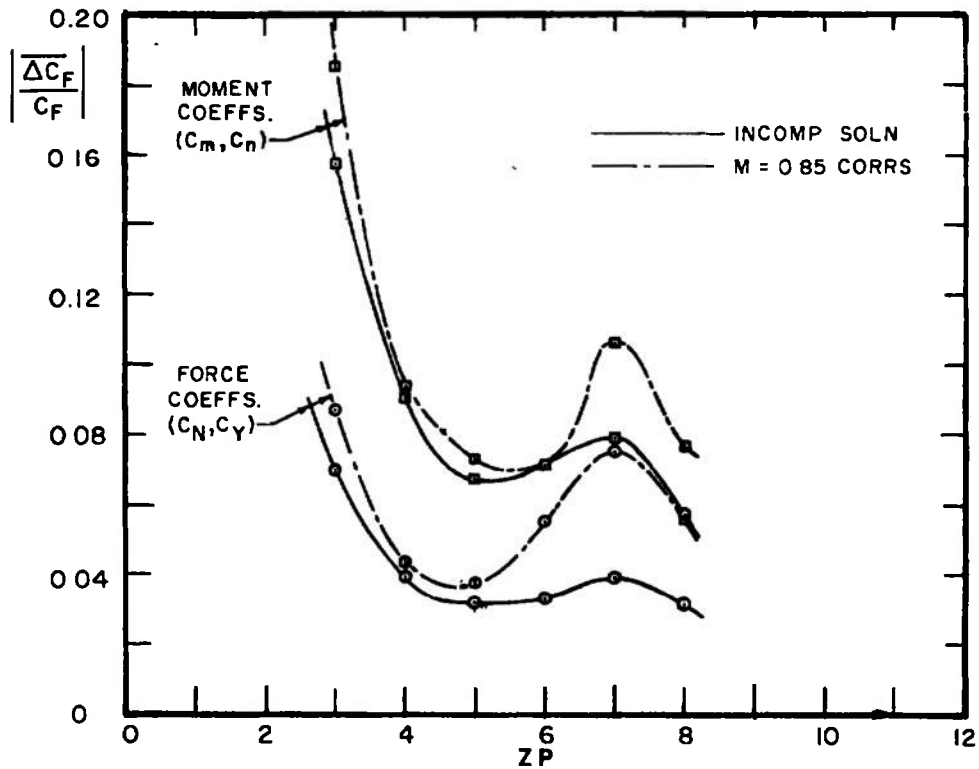
a. Best Comparison, C_Y versus ZP, Constant Yaw, Zero Pitch, $YP = -6.16$

b. Worst Comparison, C_m versus ZP, Constant Pitch, Zero Yaw, $YP = -6.16$

Fig. 9 Comparison of Incremental Variation of Calculated and Experimental Force Coefficients in F-4C Flow Field, $M = 0.85$



a. Average Discrepancy in C_N , C_m , C_Y , C_n , and Overall



b. Variation of Average Discrepancy with ZP

Fig. 10 Average Discrepancy Between Calculated and Experimental Force Coefficients, Incremental Basis, Pitch and Yaw = -4 deg and 6 deg, $M = 0.85$

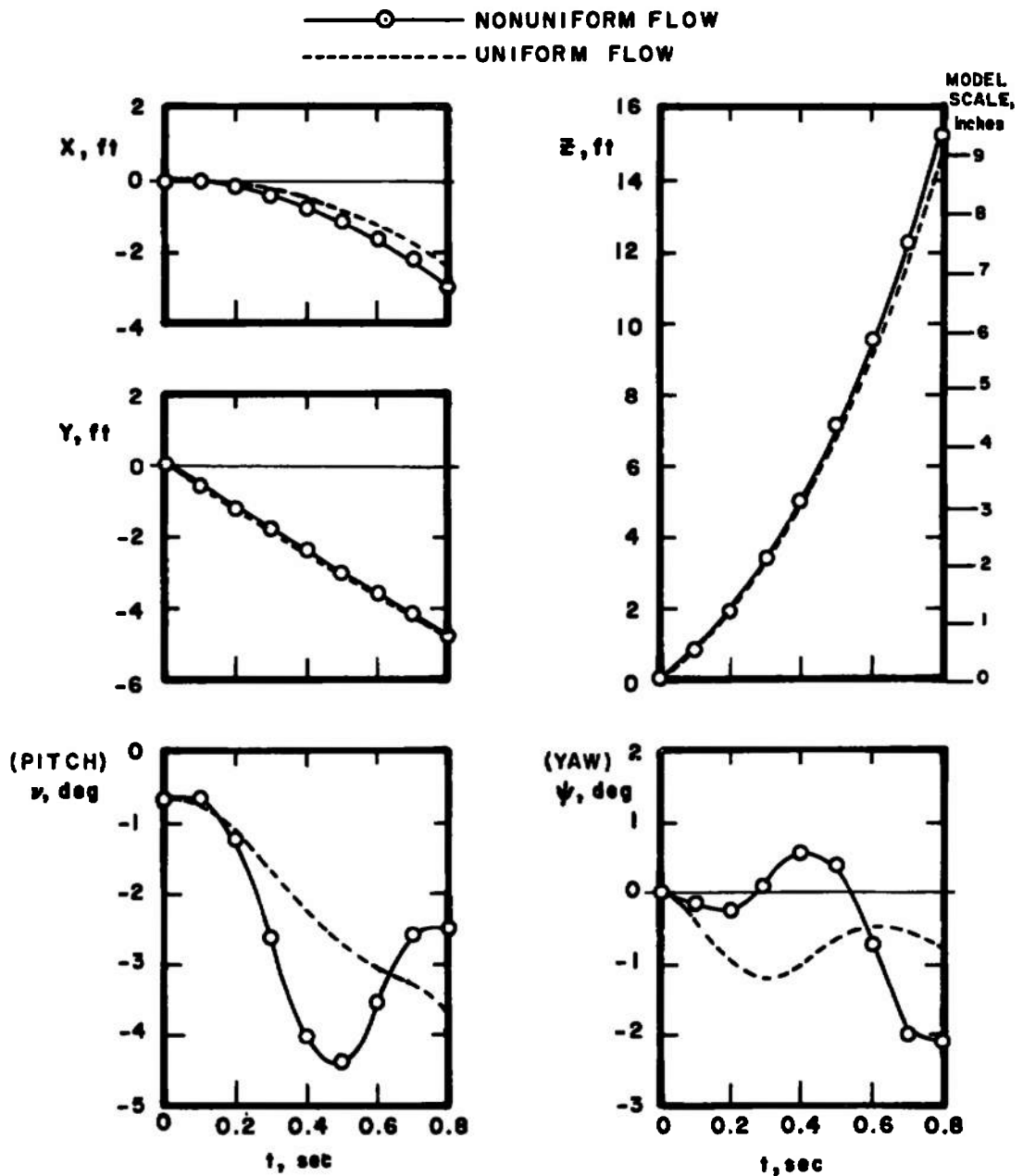


Fig. 11 Comparison of Separation Trajectories with Uniform Flow Force Coefficients and Nonuniform Flow Force Coefficients, F-4C Aircraft, 250-lbm M-117 Bomb, on No. 2 TER Station, Inboard Pylon, $M = 0.50$, 5000-ft Altitude, Pitch Angle of A/C = 0.30 deg, 1000-lb Ejector Force at 45 deg to Vertical

APPENDIX II

USER'S GUIDE TO COMPUTER PROGRAM

A user's guide for application of the vortex-lattice and store trajectory computer program is presented. This includes a listing of the source program, together with a description of the input data and other variables for which values must be furnished by the user. A diagram of general program structure and subroutine function is given in Fig. II-1. In order to assist the user in becoming familiar with the operational aspects of the program, input and output data are displayed for a sample run.

The program is coded in the FORTRAN IV Language for execution on the IBM System/360 or 370 computer. The values for the variables in the program which must be supplied by the user are provided in the following forms: inputs from punched card data sets, magnetic tape units, and disk storage units; arithmetic expressions; and maximum values of subscripts in DIMENSION statements.

The unit of length for all the variables defined in the MAIN program of Program C is feet. An exception to this occurs in the version of the MAIN program of Program C, which is used when trajectories are not computed. The units of the coordinates of the point of rotation of the store (denoted by XORIG, YORIG, and ZORIG) are inches. For all other variables in Programs A and C (the unit of length does not enter into Program B), length is prescribed in inches. The reason for this distinction is that a convenient unit for displacement of a full-scale store along a trajectory is feet, while a convenient unit for a model store and its associated NUFF is inches.

The sample run assumes that the store being analyzed is an M-117 bomb modeled with 156 vortices on each side of the x-z plane of geometrical symmetry, and a uniform flow field. (This latter assumption has been introduced to avoid the necessity of displaying NUFF input data below.) The store geometry and the vortex lattice used to model the store are shown in Fig. 1. For the sake of brevity, only a few representative values of output data are presented; these should be adequate for the user to verify his execution of the program.

VORTEX-LATTICE PROGRAM

(With Trajectory Option)

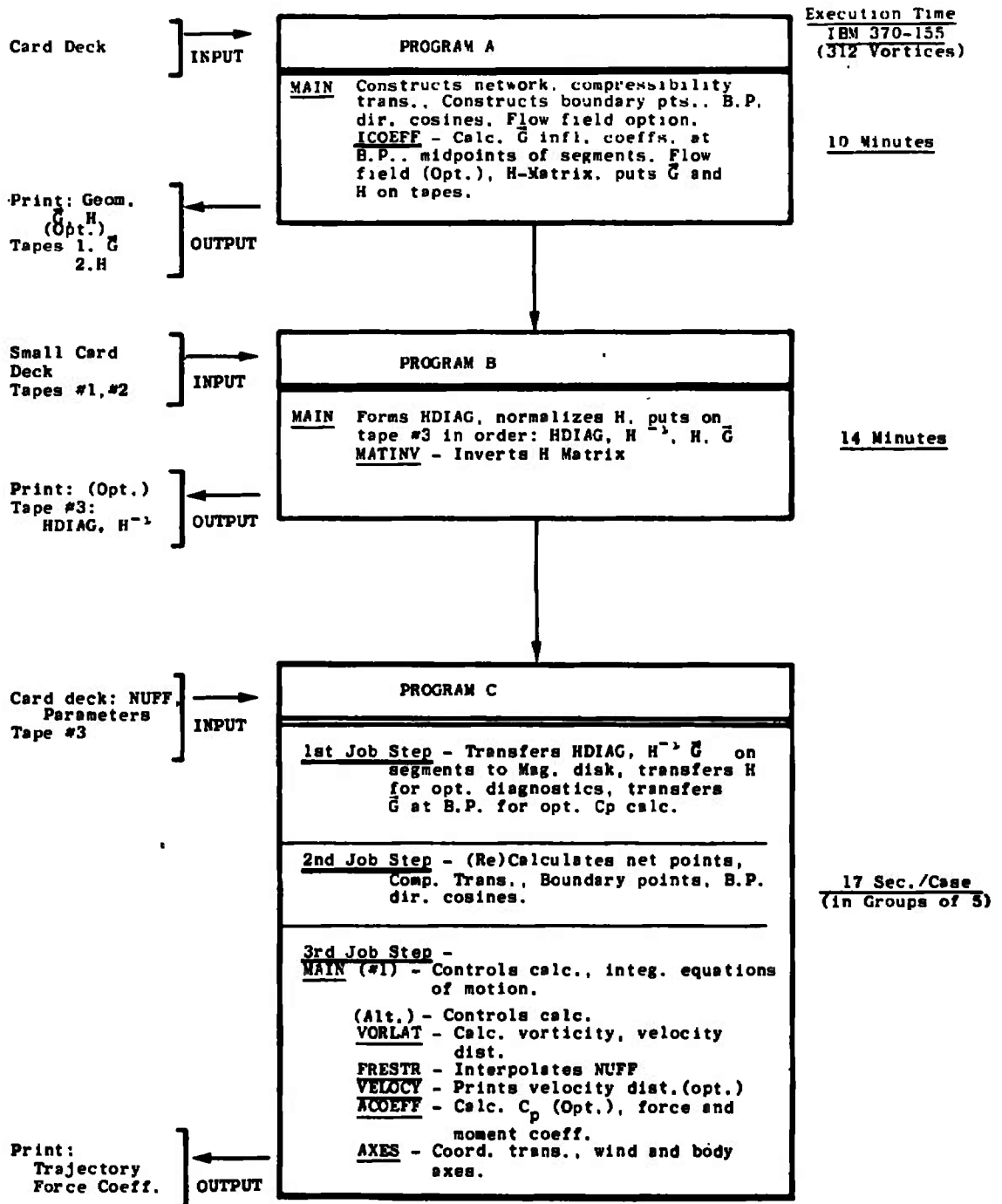


Fig. II-1 Diagram of Structure of Vortex-Lattice Program with Trajectory Option

2.1 USER'S GUIDE TO PROGRAM A

Input Data

The following data are read from punched cards. (The first card contains arbitrary descriptive information furnished by the user.)

AMACH	Free-stream Mach number
NWP	Number of wing parts (a wing part is defined in Ref. 3, p. 12)
NCHORD(IW)	Number of vortex network points on wing part IW in the chordwise direction (Ref. 3, p. 10)
NSPAN(IW)	Number of network points in the spanwise direction
MTIP = 0	Contiguous wing part tips (e. g. , closed body)
= 1	Terminal wing part tips (e. g. , wing, tail, fins)
MROOT = 0	Contiguous wing part roots (e. g. , closed body)
= 1	Terminal wing part roots (e. g. , vertical tail, fins)
NSYM = 1	Symmetry about x-z plane prevails (uniform flow field and no yaw or roll)
= 2	Without symmetry about x-z plane (nonuniform flow field or yaw or roll)

Influence coefficients can be optionally computed at as many as four sets of points. The input which specifies whether or not the calculations are to be performed at each of the sets is IVEL(ISOLVE). When ISOLVE = 1, the set of points is the boundary points; when ISOLVE = 2, the set of points is the midpoints of spanwise vortex segments; when ISOLVE = 3, the set of points is the midpoints of the chordwise vortex segments, and when ISOLVE = 4, the points are arbitrarily located off the surface of the planform.

IVEL(ISOLVE) = 0	Do not compute $\vec{G}$
= 1	Compute $\vec{G}$
IPRG = 0	Do not print $\vec{G}$
= 1	Print $\vec{G}$
IPRH = 0	Do not print H
= 1	Print H
EPS, EPSR	Magnitude tests used in SUBROUTINE ICOEFF (Ref. 3, p. 26.) The criteria used to specify values of magnitude tests in the analysis reported herein are as follows: (1) that EPSR be an order of magnitude less than the smallest

of the distances of boundary points to neighboring vortex segments, and (2) that EPS be an order of magnitude greater than the square of EPSR.

If velocity calculations are to be made at points in the flow field off the surface of the bomb (this requires specifying IVEL(4) = 1), the coordinates of these points (denoted by the names XFLOWF, YFLOWF, and ZFLOWF) are read from punched cards.

Variables which the user must define by FORTRAN arithmetic expressions are (1) the coordinates of the network points (XNET, YNET, and ZNET) and (2) the direction cosines (AX, AY, and AZ) of the vortices extending downstream from trailing edge network points. An aerodynamic planform can not be represented by a vortex network in an arbitrary manner. Guidelines for proper selection of the coordinates of network points are given in Section 5.1 of Ref. 1 and in Section VIII of Ref. 3. Considerations involved in specifying the trailing vortex direction cosines are discussed in Section VIII.F of Ref. 3. In performing the calculations discussed in this report, the trailing vortices were assumed to be parallel to the axis of symmetry.

Dimensions of Arrays

The values of the dimensions of certain variables depend upon the detailed manner in which the aerodynamic planform is modeled with vortex networks. These variables, and the corresponding dimensions are as follows:

NCHORD(NWP), NSPAN(NWP), NSCORD(NWP),
 NVOR(NWP), NUC(NWP, 4), NUS(NWP, 4), NVEL(NWP, 4),
 BDC(NVORT, 3), AX(NWP), AY(NWP), AZ(NWP),
 MTIP(NWP), MROOT(NWP), H(NVORT), G(NVORT, 3)

where NVORT is the total number of vortices on half of a planform (i. e., on one side of the x-z plane of symmetry).

The dimensions of the first, second, and third subscripts, respectively, of the network point coordinates XNET, YNET, and ZNET are (1) the maximum value of NCHORD(IW), (2) the maximum value of NSPAN(IW), and (3) NWP. The dimensions of the first, second, and third subscripts, respectively, of boundary point coordinates AKSI, ETA, and ZETA are (1) the maximum of NCHORD(IW)-1, (2) the maximum of NSPAN(IW)-1, and (3) NWP.

The dimensions of the first subscripts of DIC and EIC in SUBROUTINE ICOEFF are each equal to the maximum value of NCHORD(IW).

PROGRAM A
INPUT DATA FROM PUNCHED CARDS
SAMPLE RUN

AEDC-TR-72-162

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75	77	79
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80
G AND H MATRICES FOR AN M-17 BOMB AT M=0.0 ARE COMPUTED FOR SAMPLE RUN																																							
0.0																																							

3																																							
24					5					9					5					9					5														

0					0					1					1					1					1														
2																																							

1					1					1					0														
1					1																								

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75	77	79
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80
1. E-05										1. E-03																													

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75	77	79
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74	76	78	80

```

PROGRAM A
G AND H MATRICES ARE COMPUTED IN THIS PROGRAM
C COMMON/CNFI / NNP , NSCORD(3) , NVOR (3) ,
1 NCHORD(3) , NSPAN (3) , NVEL (3,4) , NIUT (4) ,
2 NUC (3,4) , NUS (3,4) , XNET (24,5,3) , YNET (24,5,3) , ZNET (24,5,3) ,
COMMON/CCORD/ AKS (23,4,3) , ETA (23,4,3) , ZETA (23,4,3) ,
1 XFLOW( 1, 1) , YFLOW( 1, 1) , ZFLOW( 1, 1) ,
2 ANE (156,3) , AX (13) , AY (13) , AZ (13) ,
3 COMMON/CPRIOT/ IPRG , IPRH ,
DIMENSION CXO (13) ,
1 MTIP(3) , MRDNT(3) , IVEL(4)
CALL FRSET (209,10,5,2)
CALL FRSET (221,10,5,2)
CALL FRSET (252,10,5,2)
CALL FRSET (753,10,5,2)
7100 FORMAT( 16I5
7200 FORMAT( AF10.0 )
7300 FORMAT( RMH
1 READ(5,7300)
RFAD(5,7200) ANACH
READ(5,7100) NNP
RFAD(5,7100) (NCHORD(IW), NSPAN(IW), IW=1,NNP)
READ(5,7100) MRDNT(IW), IW=1,NMP)
RFAD(5,7100) NSYM
READ(5,7100) IVEL(ISOLVE), ISOLVE=1,4)
READ(5,7100) IPRG : IPRH
READ(5,7200) EPS : FPSR
DO 1100 IW=1,NNP
AX(IW) = - 1.
AY(IW) = 0.
1100 AZ(IW) = 0.
C
C FOLLOWING FORMULAS FOR NETWORK POINTS APPLICABLE ONLY TO M-117 BDMR
XNET( 5,1,1) = 4.345
XNET( 9,1,1) = 3.435
XNET(11,1,1) = 2.935
XNET(15,1,1) = 2.035
XNET(19,1,1) = 1.058
XNET(24,1,1) = 0.
C
1254 DO 1254 I=1,A
XNET(I,1,1) = XNET(5,1,1) + ( I-5) * (XNET(11,1,1)-XNET(5,1,1))
1 XNET(10,1,1) = ( XNET(9,1,1) + XNET(11,1,1) ) / 2.
C
1256 DO 1256 I=12,14
XNET(I,1,1) = XNET(11,1,1) + ( I-11)* (XNET(15,1,1)-XNET(11,1,1))
1 XNET(13,1,1) = ( XNET(12,1,1) + XNET(14,1,1) ) / 2.
C
1258 DO 1258 I=16,18
XNET(I,1,1) = XNET(15,1,1) + ( I-15)* (XNET(19,1,1)-XNET(15,1,1))
1 XNET(17,1,1) = ( XNET(16,1,1) + XNET(18,1,1) ) / 2.
C
1260 DO 1260 I=20,23
XNET(I,1,1) = XNET(19,1,1) + ( I-19)* (XNET(24,1,1)-XNET(19,1,1))
1 XNET(21,1,1) = ( XNET(20,1,1) + XNET(22,1,1) ) / 2.
C
1262 DO 1262 J=2,NNS
J=2,NNS
XNET(I,J,1) = XNET(I,1,1)
C
XO = 1.055
ZO = - SORT( 1.6**2 - XO**2 )
ZNET( 1,1,1) = -.2815
ZNET(11,1,1) = -.2815
ZNET(15,1,1) = -.4
ZNET(19,1,1) = -.4
ZNET(24,1,1) = 0.
C
1272 DO 1272 I=-11, 1282, 1282, 1272
IF( I=-11 ) 1284, 1284, 1274
1274 IF( I=-15 ) 1284, 1284, 1274
1276 IF( I=-19 ) 1286, 1286, 1276
1276 IF( I=-24 ) 1288, 1290, 1290
C
1282 I=1,11
ZCL = - ZNET(1,1,1)
GO TO 1292
C
1284 I=12,15
ZCL = ZNET(11,1,1) + ( I-11)* (ZNET(15,1,1)-ZNET(11,1,1))
1 ZCL = - ZCL
GO TO 1292
C
1286 I=16,19
ZCL = - ZNET(15,1,1)
GO TO 1292
C
1288 I=20,23
ZCL = ZO
1 GO TO 1292
C
1290 I=24
ZCL = 0.
C
1292 YNET(1,1,1) = 0.
YNET(1,2,1) = ZCL * SIN( 3.1416 / 4. )
YNET(1,3,1) = ZCL * SIN( 3.1416 / 4. )
YNET(1,4,1) = 0.
ZNET(1,1,1) = - ZCL
ZNET(1,2,1) = - ZCL * SIN( 3.1416 / 4. )
ZNET(1,3,1) = 0.
ZNET(1,4,1) = ZCL * SIN( 3.1416 / 4. )
1294 CONTINUE
C
IF( NWP=1 ) 1380, 1380, 1300
NMC = NCHORD(2)
NMC1 = NCHORD(2)-1
NMS = NSPAN(2)
NMS1 = NSPAN(2)-1

```

```

XNET(1,1,2) = 3.9450
XNET(1,NNS,2) = 3.4350
XNET(NNC,1,2) = 4.3950
XNET(NNC,NNS,2) = 4.3950
DO 1312 J=1,NNS
  XNET(1,J,2) = XNET(1,1,2) + (XNET(NNC,1,2)-XNET(1,1,2) /
  1 * FLOAT(J-1) / (NNS-1)
  XNET(1,NNS,2) = XNET(1,NNS,2)
  2 + (XNET(NNC,NNS,2)-XNET(1,NNS,2) /
  * FLOAT(J-1) / (NNS-1)
DO 1310 J=1,NNS
  XNET(1,J,2) = XNET(1,1,2) + (XNET(1,NNS,2)-XNET(1,1,2) /
  1 * FLOAT(J-1) / (NNS-1)
1312 CONTINUE
C
  YNET(1,1,2) = 0.4020
  YNET(1,NNS,2) = 0.1991
  ZNET(1,1,2) = -YNET(1,1,2)
  ZNET(1,NNS,2) = -YNET(1,NNS,2)
  DO 1350 J=1,NNS
    YNET(1,J,2) = YNET(1,1,2) + (YNET(1,NNS,2) - YNET(1,1,2) /
    1 * FLOAT(J-1) / (NNS-1)
    ZNET(1,J,2) = ZNET(1,1,2) + (ZNET(1,NNS,2) - ZNET(1,1,2) /
    * FLOAT(J-1) / (NNS-1)
  C
    DO 1360 J=1,NNS
      DO 1360 J=1,NNS
        YNET(1,J,2) = YNET(1,J,2)
        ZNET(1,J,2) = ZNET(1,J,2)
      C
        DO 1370 J=1,NNS
          DO 1370 J=1,NNS
            XNET(1,J,3) = XNET(1,J,2)
            YNET(1,J,3) = YNET(1,J,2)
            ZNET(1,J,3) = -ZNET(1,J,2)
          C
            DO 1420 IW=1,NWP
              NNC = NCHORD(IW)
              NNS = NSPAN(IW)
              DO 1420 J=1,NNS
                XNET(1,J,IW) = XNET(1,J,IW)
                YNET(1,J,IW) = -YNET(1,J,IW)
              C
                1420 ABOVE FORMULAS FOR NETWORK POINTS APPLICABLE ONLY TO M-117 BOMB
              C
                BETA = SORT(1. - AMACH**2)
                DO 1500 IW=1,NWP
                  NNC = NCHORD(IW)
                  NNS = NSPAN(IW)
                  DO 1500 J=1,NNS
                    YNET(1,J,IW) = BETA * YNET(1,J,IW)
                    ZNET(1,J,IW) = BETA * ZNET(1,J,IW)
                  C
                    WRITE(6,7300)
                    WRITE(6,8004)
                    8004 FORMAT(1X,AMACH,1X,AMACH,1X,6F20.2)
                    WRITE(6,8110)
                    8110 FORMAT(1X,NWP,1X,NWP,1X,6I20)
                    C
                      WRITE(6,8120)
                      8120 FORMAT(1X,(NCHORD(IW),NSPAN(IW),IW=1,NWP)
                      1 NCHORD(1),NSPAN(1),NCHORD(3),NSPAN(3))
                      WRITE(6,8040)
                      8040 FORMAT(1X,(MTIP(IW,1),MR00T(1),
                      1 MTIP(IW,2),MR00T(2),MTIP(IW,3),MR00T(3))
                      2 MTIP(1),MR00T(1),MTIP(3),MR00T(3))
                      C
                        WRITE(6,8130)
                        8130 FORMAT(1X,NSYM,1X,NSYM,1X,6I20)
                        C
                          WRITE(6,8090)
                          8090 FORMAT(1X,(IVEL(1),ISOLVE=1,4(I
                          1 IVEL(3),IVEL(4),IVEL(1),IVEL(2))
                          C
                            WRITE(6,8160)
                            8160 FORMAT(1X,IPRG,1X,IPRG,1X,PRH,1X,PRH,1X,6I20)
                            C
                              WRITE(6,8170)
                              8170 FORMAT(1X,EPS,1X,EPS,1X,EPSP,1X,EPSP,1X,6E20.2)
                              WRITE(6,8140)
                              8140 FORMAT(1X,(IW,AX(IW),AY(IW),AZ(IW),IW=1,NWP)
                              1 / AX(IW),AY(IW),AZ(IW) / 11M,11M,11M,11M,6I20)
                              C
                                DO 2130 IW=1,NWP
                                  NUC(IW,1) = NCHORD(IW)-1
                                  NUS(IW,1) = NSPAN(IW)-1
                                  NUC(IW,2) = NCHORD(IW)
                                  NUS(IW,2) = NSPAN(IW)-1
                                  NUC(IW,3) = NCHORD(IW)-1
                                  NUS(IW,3) = NSPAN(IW)
                                C
                                  NVOR(1) = (NCHORD(1)-1) * (NSPAN(1)-1)
                                  DO 2140 ISOLVE=1,3
                                    NVEL(1,ISOLVE) = NUC(1,ISOLVE) * NUS(1,ISOLVE)
                                    2140 IF(NWP=1) 2158,2158,2148
                                    DO 2150 IW=2,NWP
                                      NVOR(IW) = NVOR(IW-1) + (NCHORD(IW)-1) * (NSPAN(IW)-1)
                                      DO 2150 ISOLVE=1,3
                                        NVEL(IW,ISOLVE) = NVEL(IW-1,ISOLVE)
                                        2150 NVEL(IW,ISOLVE) = NUC(IW,ISOLVE) * NUS(IW,ISOLVE)
                                      2158 DO 2160 ISOLVE=1,3
                                        NVOR(IW) = NVOR(IW,ISOLVE)
                                        2160 NVOR(IW) = NVOR(IW,ISOLVE)
                                  C
                                    IF AKSI, FTA, ZETA ARE LOCATED DIFFERENTLY WITH RESPECT TO XNET,
                                    YNET, ZNET THAN IS USED HEREIN, CORRESPONDING CHANGES MUST BE MADE
                                    IN SUBP CALCULATIONS (SUBROUTINE ACOEFF, ISOLVE=1)
                                    DO 2450 IW=1,NWP
                                      NVORC = NCHORD(IW)-1
                                      NVORS = NSPAN(IW)-1
                                      DO 2450 IP=1,NVORC
                                        I = IP

```

```

      DN 2650  IW=1,NVORS
      J = 1
      IF( IW-1 ) 2610, 2610, 2620
2610  IT = (IP-1)*NVORS + 10
      GO TO 2630
2620  IT = NVOR(IW-1) + (IP-1)*NVORS + 10
2630  CONTINUE
      AKSI(IP,IO,IW) = ( XNET(I,J,IW) + XNET(I+1,J+1,IW)
      + XNET(I+1,J,IW) + XNET(I+1,J+1,IW) ) / 4.
      ETA(IP,IO,IW) = ( YNET(I,J,IW) + YNET(I+1,J+1,IW)
      + YNET(I+1,J,IW) + YNET(I+1,J+1,IW) ) / 4.
      ZETA(IP,IO,IW) = ( ZNET(I,J,IW) + ZNET(I+1,J+1,IW)
      + ZNET(I+1,J,IW) + ZNET(I+1,J+1,IW) ) / 4.
      CX = XNET(I,J+1,IW) - XNET(I+1,J,IW)
      CY = YNET(I,J+1,IW) - YNET(I+1,J,IW)
      CZ = ZNET(I,J+1,IW) - ZNET(I+1,J,IW)
      DX = XNET(I,J,IW) - XNET(I+1,J+1,IW)
      DY = YNET(I,J,IW) - YNET(I+1,J+1,IW)
      DZ = ZNET(I,J,IW) - ZNET(I+1,J+1,IW)
      CXD(1) = CY * DZ - DY * CZ
      CXD(2) = - CX * DZ + DX * CZ
      CXD(3) = CX * DY - DX * CY
      ARSCXD = SQRT( CXD(1)**2 + CXD(2)**2 + CXD(3)**2 )
      DN 2650  N=1,3
      ROC(IT,N) = CXD(N) / ARSCXD
C
      DN 2290  IW=1,NWP
      NNC = NCHORD(IW)
      NNS = NSPAN(IW)
      GO TO( R210, R220, R230 ), IW
      R210  WRITE(6,R212) IW
      R212  FORMAT(//, ' NETWORK POINTS ON BODY J=1 IS TOP (-Z) CL J=NJ IS BOTTOM (+Z) CL ')
      GO TO R250
      R220  WRITE(6,R222) IW
      R222  FORMAT(//, ' NETWORK POINTS ON UPPER FIN J=1 IS TIP J=NJ IS ROOT ')
      GO TO R250
      R230  WRITE(6,R232) IW
      R232  FORMAT(//, ' NETWORK POINTS ON LOWER FIN J=1 IS TIP J=NJ IS ROOT ')
      GO TO R250
      R250  WRITE(6,R260)
      R260  FORMAT(//, ' XNET(I,J,IW) ')
      DO 2662  J=1,NNC
      R262  WRITE(6,R264) J, (XNET(I,J,IW), J=1,NNS)
      R264  FORMAT(//, ' I=1,12, 9F10.4 ')
      R270  WRITE(6,R270)
      R270  FORMAT(//, ' YNET(I,J,IW) ')
      DO 2772  J=1,NNC
      R272  WRITE(6,R274) J, (YNET(I,J,IW), J=1,NNS)
      R274  FORMAT(//, ' I=1,12, 9F10.4 ')
      R280  WRITE(6,R280)
      R280  FORMAT(//, ' ZNET(I,J,IW) ')
      DO 2882  J=1,NNC
      R282  WRITE(6,R284) J, (ZNET(I,J,IW), J=1,NNS)
      R284  FORMAT(//, ' I=1,12, 9F10.4 ')
      R290  CONTINUE
C
      DO 33RR  IW=1,NWP
      NVORC = NCHORD(IW)-1
      NVORS = NSPAN(IW)-1
      WRITE(6,3310) IW
      3310  FORMAT(//, ' BOUNDARY POINTS IW=1,11 ')
      R360  WRITE(6,R360)
      R360  FORMAT(//, ' AKSI(IP,IO,IW) ')
      DO 3362  IP=1,NVORC
      R362  WRITE(6,R364) IP, (AKSI(IP,IO,IW), IO=1,NVORS)
      R364  FORMAT(//, ' IP=1,12, 8F10.5 ')
      R370  WRITE(6,R370)
      R370  FORMAT(//, ' ETA(IP,IO,IW) ')
      DO 3372  IP=1,NVORC
      R372  WRITE(6,R374) IP, (ETA(IP,IO,IW), IO=1,NVORS)
      R374  FORMAT(//, ' IP=1,12, 8F10.5 ')
      R380  WRITE(6,R380)
      R380  FORMAT(//, ' ZETA(IP,IO,IW) ')
      DO 3382  IP=1,NVORC
      R382  WRITE(6,R384) IP, (ZETA(IP,IO,IW), IO=1,NVORS)
      R384  FORMAT(//, ' IP=1,12, 8F10.5 ')
      R388  CONTINUE
      R410  WRITE(6,R410)
      R410  FORMAT(//, ' ROC(IT,N) DIRECTION COSINES AT BOUNDARY POINTS
      N=1 IS X-COMPONENT N=2 IS Y N=3 IS Z ')
      DO 4420  IW=1,NWP
      NVORC = NCHORD(IW)-1
      NVORS = NSPAN(IW)-1
      DN 4420  N=1,3
      DN 4420  IP=1,NVORC
      R412  IF( IW-1 ) R412, 8412, 8414
      R412  IT1 = (IP-1)*NVORS + 1
      R412  IT2 = IP * NVORS
      GO TO 8416
      R414  IT1 = NVOR(IW-1) + (IP-1)*NVORS + 1
      R414  IT2 = NVOR(IW-1) + IP * NVORS
      R416  WRITE(6,841R) IW, N, IP, (ROC(IT,N), IT=IT1,IT2)
      R418  FORMAT(//, ' IW=1,11, N=1,11, IP=1,12, 8F10.5 ')
      R420  CONTINUE
C
      NWP = NWP
      DN 3400  ISOLVE=1,4
      IF( ISOLVE=1 ) 3400, 3400, 3310
      IF( ISOLVE=4 ) 3380, 3320, 3320
      3310  CONTINUE
      3320  CONTINUE
      THE FOLLOWING STATEMENTS, THROUGH 3372, ARE USED TO COMPUTE VELOCITY
      AT POINTS OFF THE SURFACE OF THE AERODYNAMIC PLANFORM
      READ(5,7100) NWP
      DN 3350  IW=1,NWP
      READ(5,7100) NUC(IW,4), NUS(IW,4)
      NVELC = NUC(IW,4)
      NVELS = NUS(IW,4)
      DO 7710  J=1,NVELC
      7710  READ(5,7200) (XFLOW(I,JJ), JJ=1,NVELS)
      DO 7720  J=1,NVELC
      7720  READ(5,7200) (YFLOW(I,JJ), JJ=1,NVELS)
      DO 7730  J=1,NVELC
      7730  READ(5,7200) (ZFLOW(I,JJ), JJ=1,NVELS)
      3350  CONTINUE
      3360  NVEL(I,4) = NUC(I,4) * NUS(I,4)

```

```

      IF( NIMP-1 ) 3372, 3372, 3364
3344  ON 3370 IW=2,NIMP
3370 NVEL(IW,4) = NVEL(IW-1,4) + NUC(IW,4) * NUS(IW,4)
3372 NUT(4) = NVEL(NIMP,4)
C 3380 NVELT = NUT(1SOLVE)
      CALL ICNFFF( NVELT, NVORT, 1SOLVE, NSYM,
     1  HTIP, MRROT,
     1  EPS, EPSQ )
C 3400 CONTINUE
C  IF( IVEL(1) ) 4200, 4200, 4100
4100 END FILE 23
      REWIND 23
4200 END FILE 24
      REWIND 24
      STOP
      END

```

```

SUBROUTINE ICOEFFI NVELT, NVORT, ISOLVE, NSYM,
1 MTIP, MRD, EPS, MRD, EPS
COMMON/CNET / NWP, NSCORO(3), NVOR(3),
2 NUC(3,4), MUS(3,4), NVEL(3,4), NUT(4),
COMMON/CCORO/ XNET(24,5,3), YNET(24,5,3), ZNET(24,5,3),
3 AKSI(23,4,3), ETA(23,4,3), ZETA(23,4,3),
XFLOW(1,1), YFLOW(1,1), ZFLOW(1,1),
4 ROC(156,3), AX(13), AY(13), AZ(13)
= COMMON/CPRINT/ IPRG, IPRM
DIMENSION
1 MTIP(3), MRD(3),
2 R(2), COSTH(2), ASEG(156), G(156,3),
3 OIC(24,3), EIC(24,2,3), FIC(2,3), SUMEIC(2,3)
APOKLF(A,B,C,D,E,F) = (A-B)*(C-E) - (D-E)*(F-A)
NJSYM = 3-NSYM

C
DO 6200 LSYM=1,NSYM
DO 6000 IW=1,NWP
WVELC = NUC(IW,ISOLVE)
WVELS = MUS(IW,ISOLVE)
DO 6000 II=1,WVELC
DO 6000 JJ=1,WVELS
IF( IW-1 ) 510, 510, 520
510 IT = (II-1)*WVELS + JJ
GO TO 530
520 IT = NVEL(IW-1,ISOLVE) + (II-1)*WVELS + JJ
530 GO TO 610, 630, 640, ISOLVE
610 IP = II
IO = JJ
XP = AKSI(IP,IO,IW)
YP = ETA(IP,IO,IW)
ZP = ZETA(IP,IO,IW)
GO TO 700

C
620 I = II
J = JJ
XP = ( XNET(I,J,IW) + XNET(I,J+1,IW) ) / 2.
YP = ( YNET(I,J,IW) + YNET(I,J+1,IW) ) / 2.
ZP = - ABS( YP )
GO TO 700

C
630 I = II
J = JJ
XP = ( XNET(I,J,IW) + XNET(I+1,J,IW) ) / 2.
YP = ( YNET(I,J,IW) + YNET(I+1,J,IW) ) / 2.
ZP = - ABS( YP )
GO TO 700

C
640 XP = XFLOW(I,J,JJ)
YP = YFLOW(I,J,JJ)
ZP = ZFLOW(I,J,JJ)

C
700 DO 5200 JSYM=1,NJSYM
DO 5000 JW=1,NWP
NWC = NCMORO(JW)
NVORC = NCMORO(JW)-1
NVORS = NSPAN(JW)-1
DO 5000 K=1,NVORC
IF( JW-1 ) 1110, 1110, 1120
1110 JT = (K-1)*NVORS + L
GO TO 1150
1120 JT = NVOR(JW-1) + (K-1)*NVORS + L
1150 CONTINUE

C
IF( ISOLVE-2 ) 1250, 1220, 1250
IF( LSYM-JSYM ) 1250, 1230, 1250
IS EQUIVALENT TO LSYM=1 AND JSYM=1
1220 IF( IW-JW ) 1250, 1232, 1250
1230 IF( II-K ) 1250, 1232, 1250
1232 IF( JJ-L ) 1250, 1240, 1250
1240 CONTINUE
WRITE(6,R100) LSYM,IW,II,JJ, JSYM,JW,K,L
R100 = 415, 10X, 415
GO TO 1450
1250 CONTINUE

C
FL = SORT( ( XNET(K,L+1,JW) - XNET(K,L,JW) ) **2
+ ( YNET(K,L+1,JW) - YNET(K,L,JW) ) **2
+ ( ZNET(K,L+1,JW) - ZNET(K,L,JW) ) **2 )
DO 1320 NN=1,2
R(NN) = SORT( ( XP - XNET(K,L-1+NN,JW) ) **2
+ ( YP - YNET(K,L-1+NN,JW) ) **2
+ ( ZP - ZNET(K,L-1+NN,JW) ) **2 )

C
DO 1334 NN=1,2
IF( ELOR(NN) - EPS ) 1332, 1332, 1330
COSTH(NN) = (R(1)**2 - R(2)**2 - (-1)**(NN) * EL**2) / (2.*FLOR(NN))
GO TO 1334
1332 COSTH(NN) = 0.D0
WRITE(6,R112) LSYM,IW,II,JJ, JSYM,JW,K,L, EL,NN,R(NN)
R112 = 415, 10X, 415, 10X, E10.2, 15, E10.2
1334 CONTINUE

C
IF( 1.-COSTH(2)**2 ) R130, R130, 1340
WRITE(6,R132) LSYM,IW,II,JJ, JSYM,JW,K,L, COSTH(2)
R132 = 415, 10X, 415, COSTH(2)**2 > 1 ON SPAN SEGMENT
GO TO 1450

C
1340 SMALLR = R(2) * SORT( 1.-COSTH(2)**2 )
IF( SMALLR - EPS ) R150, R150, 1350
WRITE(6,R152) LSYM,IW,II,JJ, JSYM,JW,K,L, SMALLR
R152 = 415, 10X, 415, 10X, E10.2
GO TO 1450

C
1350 ASEG(1)=APOKLF( YNET(K,L+1,JW), YNET(K,L,JW), ZP,
ZNET(K,L+1,JW), ZNET(K,L,JW), YP )
ASEG(2)=APOKLF( XNET(K,L+1,JW), XNET(K,L,JW), ZP,
ZNET(K,L+1,JW), ZNET(K,L,JW), XP )

```

```

1 ASEG(3)=APOKLF( XNET(I+1,L1,JW), XNET(I,L1,JW), YP,
ABSLXR = SORT( YNET(I+1,L1,JW), YNET(I,L1,JW), YP )
IF( ABSLXR - EPS ) 8170, 8170, 1360
WRITE(6,8172) LSYM, IW, II, JJ, JSYM, JW, K, L, ABSLXR
FORMAT( ' COEFF 8172 ABSLXR < EPS ON SPAN SEGMENT'
, 415, 10X, 415, 10X, E10.2 )
GO TO 1450
1360 COEFF = ( COSTH(1) - COSTH(2) )
/ ( 4.*3.1416 * SMALLR * ABSLXR )
DO 1420 N=1,3
OIC(K,N) = ASEG(N) * COEFF
GO TO 1500
1420 DO 1460 N=1,3
OIC(K,N) = 0.00
1460 CONTINUE
1500 C
ON 2500 LR=1,2
L1 = L-1+LR
DO 2500 I=1,NVORC
IF( L-1 ) 2050, 2050, 2020
GO TO( 2030, 2050 ), LR
ON 2040 N=1,3
EIC(I,1,N)=EIC(I,2,N)
GO TO 2500
2050 CONTINUE
C
ELIMINATION OF CHORDWISE SEGMENTS
IF( ISOLVE-3 ) 2250, 2100, 2250
IF( LSYM-JSYM ) 2210, 2110, 2210
C
2100 LSYM = 1
2110 JSYM = 1
IF( IW-JW ) 2118, 2114, 2118
IF( I-I ) 2250, 2116, 2250
IF( JJ-L1 ) 2250, 2240, 2250
2114 CONTINUE
2116 GO TO 2250
2118 C
2210 CONTINUE
IF( IW-JW ) 2250, 2222, 2250
IF( MTIP(IW) ) 2224, 2224, 2232
IF( I-I ) 2250, 2226, 2250
IF( JJ-L1 ) 2250, 2228, 2250
2222 IF( JJ-I ) 7000, 2240, 2232
2224 C
2226 IF( MROOT(IW) ) 2236, 2236, 2250
2228 IF( I-I ) 2250, 2234, 2250
IF( JJ-L1 ) 2250, 2238, 2250
IF( JJ-NVELS ) 2250, 2240, 2250
C
2240 CONTINUE
WRITE(6,8200) LSYM, IW, II, JJ, JSYM, JW, I, L1
8200 FORMAT( ' COEFF 8200 COUNTER TEST DELETE CHORD SEGMENT'
, 415, 10X, 415 )
GO TO 2450
2250 CONTINUE
C
EL = SORT( ( XNET(I+1,L1,JW) - XNET(I,L1,JW) )**2
+ ( YNET(I+1,L1,JW) - YNET(I,L1,JW) )**2
+ ( ZNET(I+1,L1,JW) - ZNET(I,L1,JW) )**2 )
DO 2320 NN=1,2
R(NN)= SORT( ( XP - XNET(I-1+NN,L1,JW) )**2
+ ( YP - YNET(I-1+NN,L1,JW) )**2
+ ( ZP - ZNET(I-1+NN,L1,JW) )**2 )
DO 2334 NN=1,2
IF( EL*R(NN) - EPS ) 2332, 2332, 2330
COSTH(NN) = ( R(1)**2 - R(2)**2 - (-1)**(NN) * EL**21
/ ( 2.*EL*R(NN) )
GO TO 2334
COSTH(NN) = 0.00
2332 WRITE(6,8212) LSYM, IW, II, JJ, JSYM, JW, I, L1, EL, NN, R(NN)
8212 FORMAT( ' COEFF 8212 ELR(NN) < EPS ON CHORD SEGMENT'
, 415, 10X, 415, 10X, E10.2, I5, E10.2 )
2334 CONTINUE
C
IF( 1.-COSTH(2)**2 ) 8230, 8230, 2340
8230 WRITE(6,8232) LSYM, IW, II, JJ, JSYM, JW, I, L1, COSTH(2)
8232 FORMAT( ' COEFF 8232 COSTH(2)**2 > 1 ON CHORD SEGMENT'
, 415, 10X, 415, 10X, E10.6 )
GO TO 2450
C
2340 SMALLR = R(2) * SORT( 1.-COSTH(2)**2 )
IF( SMALLR - EPSR ) 8250, 8250, 2350
8250 WRITE(6,8252) LSYM, IW, II, JJ, JSYM, JW, I, L1, SMALLR
8252 FORMAT( ' COEFF 8252 SMALLR < EPSR ON CHORD SEGMENT'
, 415, 10X, 415, 10X, E10.2 )
GO TO 2450
2350 ASEG(1) = APOKLF( YNET(I+1,L1,JW), YNET(I,L1,JW), ZP,
ZNET(I+1,L1,JW), ZNET(I,L1,JW), YP )
ASEG(2) = APOKLF( XNET(I+1,L1,JW), ZNET(I,L1,JW), ZP,
XNET(I+1,L1,JW), XNET(I,L1,JW), YP )
ASEG(3) = APOKLF( XNET(I+1,L1,JW), XNET(I,L1,JW), YP,
YNET(I+1,L1,JW), YNET(I,L1,JW), XP )
ABSLXR = SORT( ASEG(1)**2 + ASEG(2)**2 + ASEG(3)**2 )
IF( ABSLXR - EPS ) 8270, 8270, 2360
8270 WRITE(6,8272) LSYM, IW, II, JJ, JSYM, JW, I, L1, ABSLXR
8272 FORMAT( ' COEFF 8272 ABSLXR < EPS ON CHORD SEGMENT'
, 415, 10X, 415, 10X, E10.2 )
GO TO 2450
2360 COEFF = ( COSTH(1) - COSTH(2) )
/ ( 4.*3.1416 * SMALLR * ABSLXR )
DO 2420 N=1,3
EIC(I,LR,N) = ASEG(N) * COEFF
GO TO 2500
2420 DO 2460 N=1,3
EIC(I,LR,N) = 0.00
2460 CONTINUE
2500 C
ON 3500 LR=1,2
L1 = L-1+LR
R(1) = SORT( ( XP - XNET(NNC,L1,JW) )**2
+ ( YP - YNET(NNC,L1,JW) )**2
+ ( ZP - ZNET(NNC,L1,JW) )**2 )
IF( R(1)-EPSR ) 3332, 3332, 3330
COSTH(1) = - ( AX(IW) * XP - XNET(NNC,L1,JW) )
+ AY(IW) * YP - YNET(NNC,L1,JW) )

```

```

2          * AZ(JW) * ( ZP - ZNFT(NNC,L1,JW) 1 1 /R(1)
3332      COSTH(1) = 0.00
3332      WRITE(6,8312) LSYM, IW, 11, JJ, JSYM, JW, L1, R(1)
8312      FORMAT( ' ICDEFF 8312 R(1) < FPSR ON TRAIL VORTEX'
1          , 415, ' TRAIL VOR', 215, 5X, 15, 10X, E10.2 )
C
3336      IF( 1.-COSTH(1)**2 ) R330, R330, 3340
8330      WRITE(6,8332) LSYM, IW, 11, JJ, JSYM, JW, L1, COSTH(1)
8332      FORMAT( ' ICDEFF 8332 COSTH(1)**2 > 1 ON TRAIL VORTEX'
1          , 415, ' TRAIL VOR', 215, 5X, 15, 10X, F10.6 )
3340      SMALLR = R(1) * SORT( 1.-COSTH(1)**2 (
1          IF( SMALLR-EPSR ) R350, R350, 3350
8350      WRITE(6,8352) LSYM, IW, 11, JJ, JSYM, JW, L1, SMALLR
8352      FORMAT( ' ICDEFF 8352 SMALLR < EPSR ON TRAIL VORTEX'
1          , 415, ' TRAIL VOR', 215, 5X, 15, 10X, F10.2 )
3350      GO TO 3450
C
3350      DSEG(1) = AY(JW) * ( ZP - ZNET(NNC,L1,JW) (
1          - AZ(JW) * ( YP - YNET(NNC,L1,JW) )
C
1          DSEG(2) = - AZ(JW) * ( XP - XNFT(NNC,L1,JW) (
1          - AX(JW) * ( ZP - ZNET(NNC,L1,JW) )
C
1          DSEG(3) = - AX(JW) * ( YP - YNET(NNC,L1,JW) (
1          - AY(JW) * ( XP - XNFT(NNC,L1,JW) )
8370      ABSLXR = SORT( OSFG(1)**2 + DSEG(2)**2 + DSEG(3)**2 (
8372      WRITE(6,8372) LSYM, IW, 11, JJ, JSYM, JW, L1, ABSLXR
8372      FORMAT( ' ICDEFF 8372 ABSLXR < EPS ON TRAIL VORTEX'
1          , 415, ' TRAIL VOR', 215, 5X, 15, 10X, E10.2 )
3360      GO TO 3450
1          COEFF = / ( 1.- COSTH(1) )
1          / 4.*3.1416 * SMALLR * ABSLXR
3420      DO 3420 N=1,3
1          FIC(LR,N) = DSEG(N) * COEFF
3450      GO TO 3500
3460      DO 3460 N=1,3
1          FIC(LR,N) = 0.00
3500      CONTINUE
C
C
4100      DO 4100 N=1,3
1          DO 4100 LR=1,2
1          SUMEIC(LR,N) = 0.00
C
4210      DO 4300 M=1,NVORC
1          K = NCHORD(JW)-M
4210      IF( JW-1 4210, 4210, 4220
1          JT = (K-1)*NVORS + L
4220      GO TO 4230
1          M = NVOR(JW-1) + (K-1)*NVORS + L
4230      DO 4300 N=1,3
1          DO 4240 LR=1,2
4240      SUMEIC(LR,N) = SUMEIC(LR,N) + EIC(K,LR,N)
1          GIC = OIC(K,N) - SUMEIC(LR,N) + SUMEIC(LR,N)
4250      GO TO( 4250, 4252), NSYM
4252      GO TO( 4252, 4254), JSYM
4254      G(JT,N) = GIC
4300      GO TO 4300
1          G(JT,N) = G(JT,N) - GIC
5000      CONTINUE
C
5090      GO TO( 5090, 5200), NSYM
5090      DO 5100 JW=1,NWP
1          NNC = NCHORD(JW)
1          NNS = NSPAN(JW)
5100      DO 5100 K=1,NNC
1          DO 5100 L=1,NNS
1          YNET(K,L,JW) = - YNET(K,L,JW)
C
5200      CONTINUE
C
C
8512      IF( IPRG ) R550, R550, R512
8514      IF( IT-1 ) R516, R516, R514
8516      IF( IT-NVELT ) R550, R516, R516
8516      DO 8520 N=1,3
1          WRITE(6,8518) ISOLVE, LSYM, IT, N
8518      FORMAT( ' ISOLVE =', 11, ' LSYM=', 11, ' IT=', 13, ' N=',
1          11, 10X, ' G(JT,N) AS ORIGINALLY COMPUTED ' (
8520      WRITE(6,8522) G(JT,N), JT=1,NVORT(
1          16F8.3 )
8530      WRITE(6,8540) IT
8540      FORMAT(75X, ' ICDEFF IT=', 13, ' ENTRY WRITE(24) G '(
1          16F8.3 )
8550      WRITE(24) G
C
5500      GO TO( 5500, 6000, 6000, 6000), ISOLVE
5500      DO 5520 JT=1,NVORT
1          H(JT) = 0.00
5520      DO 5520 N=1,3
1          H(JT) = H(JT) + 80C(IT,N) * G(JT,N)
1          WRITE(23) H
8562      IF( IPRM ) 6000, 6000, R562
8564      IF( IT-1 ) 8570, 8570, R564
8570      IF( IT-NVORT ) 6000, 8570, R570
8572      WRITE(6,8572) LSYM, IT, H(JT), JT=1,NVORT(
1          16F8.3 )
8572      FORMAT( ' H(JT) NON-NORMALIZED AS ORIGINALLY COMPUTED '
1          LSYM=', 11, ' IT=', 13, ' (16F8.3) )
6000      CONTINUE
C
6090      GO TO( 6200, 6090), NSYM
6090      DO 6100 JW=1,NWP
1          NNC = NCHORD(JW)
1          NNS = NSPAN(JW)
6100      DO 6100 K=1,NNC
1          DO 6100 J=1,NNS
1          YNET(1,J,JW) = - YNET(1,J,JW)
C
6200      CONTINUE
C
7000      RETURN
END

```

Output Data

The output data from Program A consists of the $\vec{G}$ and H matrices. The $\vec{G}$ matrices are written on a magnetic tape identified with data set reference number 24, and the H matrix is written on a second magnetic tape having the data set reference number 23. The entire quantity of output data which is generated in the sample run that is used herein as a benchmark is much too voluminous to display in its entirety; hence, only representative results will be shown.

When NSYM = 2 and ISOLVE = 1, the elements of the first row and first column of the x, y, z components of $\vec{G}^{(1)}$ (the first partition of $\vec{G}$, the FORTRAN designation for the first partition being LSYM = 1) are

$$G_{x1,1}^{(1)} = 0, \quad G_{y1,1}^{(1)} = 1.321, \quad \text{and} \quad G_{z1,1}^{(1)} = 3.188$$

The corresponding value of H is

$$H_{1,1}^{(1)} = -3.451$$

The initial elements of the second partition (LSYM = 2) are

$$G_{x1,1}^{(2)} = 0.126, \quad G_{y1,1}^{(2)} = 0.621, \quad \text{and} \quad G_{z1,1}^{(2)} = 0.648;$$

$$\text{and } H_{1,1}^{(2)} = -0.836$$

2.2 USER'S GUIDE TO PROGRAM B

Input Data

Several of the input variables whose values are read from punched card data sets are mentioned above in the description of Program A. The others are defined as follows:

IPRHI = 0	Do not print H^{-1}
= 1	Print H^{-1}
ISEG = 1	Entire Program B is executed in one submittal to the computer
= 2	Program B is segmented into 2 parts, each of which requires a separate submittal

IRUN = 1 This is the first submittal
 = 2 This is the second submittal when ISEG = 2; it is a
 continuation of calculations performed in the first
 submittal

The $\vec{G}$ and H matrices, which were written in Program A as data sets with reference numbers 24 and 23, respectively, are read in Program B.

Dimensions of Arrays

The values of the dimensions for the variables in COMMON/CNET/ are the same as used in Program A. Wherever the number 156 appears as a dimension in the sample program listing, it indicates that the dimensions in those locations are equal to NVORT.

AEDC-TR-72-162

[illegible]

DIV CHK
NEG SORT
RIG EXPN
NEG LGIO

```

3410 H1(IT,JT) = EH1(JT)
      IF( IPRM ) 8530, 8530, 8516
8516 IF( IT-1 ) 8530, 8520, 8518
8518 IF( IT-NVORT ) 8530, 8520, 8520
8520 WRITE(6,8522) IT, H1(IT,JT), JT=1, NVORT)
8522 FORMAT( ' H1(IT,JT) NON-NORMALIZED FROM READ(23) IT=,13
      / (1H 16F8.3) )
8530 CONTINUE
      H0IAG(IT,1) = H1(IT,IT)
      H0IAG(IT,2) = -H0IAG(IT,1)
      ON 3430 JT=1, NVORT
      H1(IT,JT) = H1(IT,JT) / H0IAG(IT,1)
3430 EH1(JT) = H1(IT,JT)
3440 WRITE(31) EH1
      END FILE 31
      REWIND 31

C
      I21 = (2-ISEG)*21 + (ISEG-1)*22
      WRITE(121) H0IAG

C
      DO 8610 IT=1, NVORT
8610 WRITE(6,8612) IT, H0IAG(IT,ISYM), ISYM=1, NSYM1
8612 FORMAT( ' H0IAG(IT,ISYM) IT=,13, 2E15.7 )
      IF( IPRM ) 8626, 8626, 8614
8614 DO 8624 IT=1, NVORT
      IF( IT-1 ) 8620, 8620, 8618
8618 IF( IT-NVORT ) 8624, 8620, 8620
8620 WRITE(6,8622) IT, H1(IT,JT), JT=1, NVORT)
8622 FORMAT( ' H1(IT,JT) NORMALIZED AS ORIGINALLY COMPUTED IT=,13
      / (1H 16F8.3) )
8624 CONTINUE
8626 CONTINUE

C
      CALL MATINVI NVORT, H1INV )

C
      GO TO( 4100, 4200 ), NSYM
4100 DO 4120 IT=1, NVORT
      DO 4110 JT=1, NVORT
4110 EH1INV(JT) = H1INV(IT,JT)
4120 WRITE(21) EH1INV
      ON 4130 IT=1, NVORT
      READ(31) EH1
4130 WRITE(21) EH1
      REWIND 31
      GO TO 6400

C
4200 CONTINUE
      DO 4240 JT=1, NVORT
      ON 4240 IT=1, NVORT
4240 EH1INV(IT) = H1INV(IT,JT)
4250 WRITE(51) EH1INV
      END FILE 51
      REWIND 51

C
      IF( IPRM ) 8656, 8656, 8640
8640 DO 8654 IT=1, NVORT
      IF( IT-1 ) 8650, 8650, 8648
8648 IF( IT-NVORT ) 8654, 8650, 8650
8650 WRITE(6,8652) IT, H1INV(IT,JT), JT=1, NVORT)
8652 FORMAT( ' H1INV(IT,JT) INVERSE OF H1 IT=,13
      / (1H 8E15.7) )
8654 CONTINUE
8656 CONTINUE

C
      DO 4330 IT=1, NVORT
      READ(23) EH2
      IF( IPRM ) 8580, 8580, 8566
      IF( IT-1 ) 8570, 8570, 8568
8568 IF( IT-NVORT ) 8580, 8570, 8570
8570 WRITE(6,8572) IT, EH2(JT), JT=1, NVORT)
8572 FORMAT( ' EH2(JT) NON-NORMALIZED FROM READ(23) IT=,13
      / (1H 16F8.3) )
8580 CONTINUE
      DO 4310 JT=1, NVORT
      EH2(JT) = EH2(JT) / H0IAG(IT,1)
4310 WRITE(32) EH2
      DO 4320 JT=1, NVORT
      H2H(JT) = 0.
      DO 4320 KK=1, NVORT
4320 H2H(JT) = H2H(JT) + EH2(KK) * H1INV(KK,JT)
4330 CONTINUE
      END FILE 32
      END FILE 41
      REWIND 23
      REWIND 32
      REWIND 41

C
      DO 4350 IT=1, NVORT
      READ(32) EH2
      ON 4350 JT=1, NVORT
4350 H2(I,JT) = EH2(JT)
      REWIND 32

C
      IF( IPRM ) 8690, 8690, 8660
8660 DO 8680 IT=1, NVORT
      IF( IT-1 ) 8670, 8670, 8668
8668 IF( IT-NVORT ) 8680, 8670, 8670
8670 WRITE(6,8672) IT, H2(I,JT), JT=1, NVORT)
8672 FORMAT( ' H2(I,JT) NORMALIZED AS ORIGINALLY COMPUTED IT=,13
      / (1H 8F15.7) )
8680 CONTINUE
8690 CONTINUE

C
      I42 = (2-ISEG)*42 + (ISEG-1)*22
      ON 4430 IT=1, NVORT
      READ(41) H2H1
      ON 4410 JT=1, NVORT
      H2H1H2(JT) = 0.
      ON 4410 KK=1, NVORT
4410 H2H1H2(JT) = H2H1H2(JT) + H2H1(KK) * H2(KK,JT)
      READ(31) EH1
      DO 4420 JT=1, NVORT
4420 FH1H2(JT) = EH1(JT) - H2H1H2(JT)
4430 CONTINUE
      REWIND 31
      REWIND 41

C
      I52 = (2-ISEG)*52 + (ISEG-1)*22

```

```

      DO 4470      JT=1,NVORT
      READ(51)     EH1INV
      DO 4480      IT=1,NVORT
      H2H1(IT) = 0.
      DO 4490      KK=1,NVORT
      H2H1(IT) = H2H1(IT) + H2(IT,KK) * EH1INV(KK)
4460      WRITE(152) H2H1
4470      CONTINUE
      REWIND 51
C
      GO TO( 4480, 4500), ISEG
4480      END FILE 42
      END FILE 52
      REWIND 42
      REWIND 52
      GO TO 4700
C
4500      DO 4510      IT=1,NVORT
      READ(31)        EH1
4510      WRITE(22)    EH1
      DO 4520      IT=1,NVORT
      READ(31)        EH2
4520      WRITE(22)    EH2
      END FILE 72
      REWIND 22
      STOP
C
C
4600      READ(22)     HDIAG
      WRITE(21)        HDIAG
C
      I42 = 22
4700      DO 4720      IT=1,NVORT
      READ(142)        EHINV1
      DO 4720      JT=1,NVORT
      HINV1(IT,JT) = EHINV1(JT)
4720      HINV1(IT,JT) = EHINV1(JT)
C
      CALL          MATINV( NVORT, HINV1 )
C
      DO 4750      IT=1,NVORT
      DO 4740      JT=1,NVORT
4740      EHINV1(JT) = HINV1(IT,JT)
      WRITE(21)     EHINV1
      IF( IPRM )   4750, 4750, 8716
8716      IF( IT-1 ) 8720, 8720, 8718
8718      IF( IT-NVORT ) 4750, 8720, 8720
8720      WRITE(6,8722) IT, (HINV1(IT,JT), JT=1,NVORT)
8722      FORMAT( 1, HINV1(IT,JT), INVERSE OF H1-H2*H1INV*H2
1      IT='13 / (1M 8E15.7) )
4750      CONTINUE
C
      I52 = (2-ISEG)*52 + (ISEG-1)*22
      DO 4820      JT=1,NVORT
      READ(152)     H2H1
      DO 4810      IT=1,NVORT
      EHINV2(IT) = 0.
      DO 4810      KK=1,NVORT
      EHINV2(IT) = EHINV2(IT) - HINV1(IT,KK) * H2H1(KK)
4810      WRITE(53)   EHINV2
4820      CONTINUE
      END FILE 53
      REWIND 53
C
      DO 4850      JT=1,NVORT
      READ(53)     EHINV2
      DO 4850      IT=1,NVORT
      HINV2(IT,JT) = EHINV2(IT)
4850      HINV2(IT,JT) = EHINV2(IT)
C
      DO 4860      IT=1,NVORT
      DO 4858      JT=1,NVORT
4858      EHINV2(JT) = HINV2(IT,JT)
      WRITE(21)     EHINV2
      IF( IPRM )   4860, 4860, 8766
8766      IF( IT-1 ) 8770, 8770, 8768
8768      IF( IT-NVORT ) 4860, 8770, 8770
8770      WRITE(6,8772) IT, (HINV2(IT,JT), JT=1,NVORT)
8772      FORMAT( 1, HINV2(IT,JT), INVERSE OF H1-H2*H1INV
1      IT='13 / (1M 8E15.7) )
4860      CONTINUE
C
      I31 = (2-ISEG)*31 + (ISEG-1)*22
      I32 = (2-ISEG)*32 + (ISEG-1)*22
      DO 4910      IT=1,NVORT
      READ(131)     EH1
4910      WRITE(21)   EH1
C
      DO 4922      IT=1,NVORT
      READ(132)     EH2
      WRITE(21)     EH2
      GO TO( 4922, 4920), ISEG
4920      REWIND 22
4922      CONTINUE
C
C
6400      CONTINUE
      DO 6480      ISOLVE=1,4
      IF( IVEL( ISOLVE ) ) 6480, 6480, 6426
6426      NVELT = NUT( ISOLVE )
      DO 6460      ISYM=1,NSYM
      DO 6460      IT=1,NVELT
      READ(24)     G
      WRITE(21)     G
      IF( IPRG )   6460, 6460, 8912
8912      IF( IT-1 ) 8916, 8916, 8914
8914      IF( IT-NVELT ) 6460, 8916, 8916
8916      DO 8920      N=1,3
8920      WRITE(6,8922) ISOLVE, ISYM, IT, N, (G(JT,N), JT=1,NVORT)
8922      FORMAT( 1, ISOLVE='1', 5X, ' ISYM='11, 5X, ' IT='13, 5X,
1      N='11, 15X, ' G(JT,N) READ FROM UNIT(24)', / (1M 16F8.31) )
6460      CONTINUE
6480      CONTINUE
      END FILE 21
      REWIND 21
      REWIND 24
      STOP
      END

```

```

SUBROUTINE MATINV( NVORT, A )
DIMENSION INDEX (156,3), AINVORT,NVORTI
NO/FILE PRECISION , SWAP , PIVOT , T
2 AMAX = NVORT
NIM2 = NVORT

REFERENCE MCCORMICK/SALVADORI P.306, KID P.168

INITIALIZATION
INDEX(J,3) OF M/S = IPVNT(J) OF KUN
ON 52 J=1,NIM2
INDEX(J,3)= 0

52 DO 550 I=1,NIM2
   IROW = I
   ICOLUM = I
   C SINCE THE M-MATRIX IS DIAGONALLY DOMINANT BYPASS PIVOTING
   GO TO 140
   C SEARCH FOR PIVOT ELEMENT
   AMAX = 0.00
   DO 105 J=1,NIM2
     IF( INDEX(J,3) = 1 ) 60, 105, 60
     DO 100 K=1,NIM2
       IF( INDEX(K,3) = 1 ) 80, 100, 715
       IF( AMAX = ABS(A(I,J,K)) ) 85, 100, 100
       IROW = J
       ICOLUM = K
       AMAX = ABS( A(I,J,K) )
     100 CONTINUE
     105 CONTINUE
     140 INDEX(ICOLUM,3) = INDEX(ICOLUM,3) + 1
     INDEX(I, 1) = IROW
     INDEX(I, 2) = ICOLUM
     C INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
     IF( IROW - ICOLUM ) 150, 310, 150
     150 DO 200 L=1,NIM2
       SWAP = A(IROW,L)
       A(IROW,L) = A(ICOLUM,L)
       A(ICOLUM,L) = SWAP
     200
     C DIVIDE PIVOT ROW BY PIVOT ELEMENT
     310 PIVOT = A(ICOLUM,ICOLUM)
     A(ICOLUM,ICOLUM) = 1.00
     DO 350 L=1,NIM2
       A(ICOLUM,L) = A(ICOLUM,L) / PIVOT
     350
     C REDUCE NON-PIVOT ROWS
     DO 550 L1=1,NIM2
       IF( L1 - ICOLUM ) 400, 550, 400
       T = A(L1,ICOLUM)
       A(L1,ICOLUM) = 0.00
       DO 450 L=1,NIM2
         A(L1,L) = A(L1,L) - A(ICOLUM,L1) * T
       450
     550 CONTINUE
     C INTERCHANGE COLUMNS
     DO 710 I=1,NIM2
       L = NIM2 + 1 - I
       IF( INDEX(I,1) = INDEX(L,2) ) 630, 710, 630
     630 JROW = INDEX(I,1)
     JCOLUM = INDEX(L,2)
     C DO 705 K=1,NIM2
     SWAP = A(K,JROW)
     A(K,JROW) = A(K,JCOLUM)
     A(K,JCOLUM) = SWAP
     705 CONTINUE
     710
     C TEST FOR SINGULARITY OF MATRIX
     ON 730 K=1,NIM2
     IF( INDEX(K,3) = 1 ) 715, 730, 715
     715 WRITE(8,100)
     8100 FORMAT(1 SINGULAR MATRIX OCCURED IN SUBROUTINE MATINV)
     STOP
     730 CONTINUE
   C RETURN
   END

```

Output Data

Values of the HDIAG and normalized H matrices may be inferred from the output data of Program A. Representative values for the illustrative case described herein are shown for elements of these matrices in addition to the partitions of the H^{-1} matrix, $A^{(1)}$ and $A^{(2)}$.

<u>First Partition</u>		<u>Second Partition</u>	
HDIAG(1, 1)	= -3.45	HDIAG(1, 2)	= 3.45
$H_{1,1}^{(1)}$	= 1.000	$H_{1,1}^{(2)}$	= 0.242
$A_{1,1}^{(1)}$	= 1.082	$A_{1,1}^{(2)}$	= -0.314

2.3 USER'S GUIDE TO PROGRAM C—FIRST JOB STEP

Input Data

The user must define the following variables by arithmetic expressions:

NSYM	This is defined in Section 2.1 of this Appendix
NVORT	Number of vortices on half of a planform
NUT(1)	Number of points on one side of the x-z symmetry plane at which influence coefficients have been computed in Program A when ISOLVE = 1 (NUT(1) = NVORT)
NUT(2)	Number of influence coefficient points when ISOLVE = 2. The contribution of NUT(2) from each wing part, IW, is NSPAN(IW) x (NCHORD(IW) - 1).
NUT(3)	Number of influence coefficient points when ISOLVE = 3. The contribution to NUT(3) from each wing part, IW, is (NSPAN(IW) - 1) x NCHORD(IW).

FORTTRAN statements used to compute NUT(1), NUT(2), and NUT(3) are to be found in Programs A and B, and in the second job step of Program C.

The matrices of geometric factors which were written in Program B as a data set using reference number 21 are read using reference number 31 in this job step.

Dimensions of Arrays

Wherever the number 156 appears as a dimension in the program listing, it indicates that the dimensions in those locations are equal to NVORT.

```

C      PROGRAM C - 1ST JOB STEP
C      THIS PROGRAM TRANSFERS TAPE(31), HDIAG, MINV, H, G, ONTO DISK(21)
C      DIMENSION
1      HDIAG (156,2), MINV (156), H (156), G (156,3)
C      NSYM = 2
C      FOLLOWING STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
C      NVORT = 156
C      NUT(1) = 156
C      NUT(2) = 168
C      NUT(3) = 195
C      ABOVE STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
C      READ(31) HDIAG
C      WRITE(21) HDIAG
C      DO 6250 ISYM=1,NSYM
C      DO 6250 IT=1,NVORT
C      READ(31) MINV
C 6250 WRITE(21) MINV
C      DO 6350 ISYM=1,NSYM
C      DO 6350 IT=1,NVORT
C      READ(31) H
C 6350 IF IHGAMA=1, WRITE(21) H
C      CONTINUE
C      DO 6450 ISOLVE=1,3
C      NVELT = NUT(ISOLVE)
C      DO 6450 ISYM=1,NSYM
C      DO 6450 IT=1,NVELT
C      READ(31) G
C      IF (ISOLVE=1) REMOVE FOLLOWING 2 CARDS
C 6440 CONTINUE
C      WRITE(21) G
C 6450 CONTINUE
C      END FILE 21
C      REMIND 31
C      REMIND 21
C      STOP
C      END

```

Output Data

A subset of the data set containing HDIAG, H^{-1} , H, and $\vec{G}$, which is read from tape in this job step, is written on magnetic disk, data set reference number 21. See Section 3.3.1 for details.

2.4 USER'S GUIDE TO PROGRAM C—SECOND JOB STEP

Input Data

The input data are similar to those of Program A, except for the additions noted below.

The following data are read from punched cards:

IHGAMA = 0	H is not contained on data set reference number 21
= 1	H is contained on data set reference number 21
ICF = 1	Laminar skin friction assumed
= 2	Turbulent skin friction assumed
NSCORD(IW)	Number of chordwise network points on the geometric surface of wing part IW; force coefficients are summed only over vortex segments corresponding to these points (e.g., not over the segments in the wake of the sample run described herein)

XM, ZM	Body axis coordinates of pitch axis, in.
YL, ZL	Body axis coordinates of roll axis, in.
XN, YN	Body axis coordinates of yaw axis, in.
XCG, YCG, ZCG	Body axis coordinates of center of gravity, in.
IBC = 0	Store is assumed immersed in a uniform flow field; i. e. , no parent aircraft is present
IBC = 1	Store is assumed immersed in a nonuniform flow field
IPRVFS = 0	Do not print the NUFF velocity components in SUB- ROUTINE FRESTR
IPRVFS = 1	Print the NUFF velocity components
IPRVEL = 0	Do not print velocity distributions in SUBROUTINE VELOCY
IPRVEL = 1	Print velocity distributions
IPRGAM = 0	Do not print the strengths of the vortices, Γ_j
IPRGAM = 1	Print Γ_j
IPRCP = 0	Do not print the pressure coefficient, C_p , distribution
IPRCP = 1	Print C_p
IPRCF = 0	Do not print the force coefficient distribution
IPRCF = 1	Print the force coefficient
IX1, IX2	Counters for the initial and final grid points at which NUFF data are input in the x-direction, usually IX1 = 1
IY1, IY2	Counters for initial and final NUFF grid points in the y-direction, usually IY1 = 1
IZ1, IZ2	Counters for initial and final NUFF grid points in the z-direction, usually IZ1 = 1
DOWNV (IX, IY, IZ)	Downwash angles (deg) of the NUFF (positive DOWNV is upwash)
SIDEV (IX, IY, IZ)	Sidewash angles (deg) of the NUFF (positive SIDEV is inwash on negative y-side of the x-z plane of symmetry)
VMAG (IX, IY, IZ)	Ratio of the magnitude of the local velocity vector to the magnitude of the velocity at infinity

The following data are defined by FORTRAN arithmetic expressions:

XP(IX), YP(IY) ZP(IZ)	Grid points at which NUFF data is input These points must be ordered in increasing values of XP, YP, and ZP
S	Reference area used in definition of the force coefficients of a model store, in ²
CBAR	Reference chord length used in definition of pitching- and yawing-moment coefficients of a model store, in.
BSPAN	Reference lateral dimension used in definition of rolling-moment coefficient of a model store, in.
XNET, YNET, ZNET	Coordinates of the vortex network points; these must be the same as the corresponding values in Program A

Dimensions of Arrays

The dimensions in this job step are the same as those in Program A, with the addition of the following. The counters for the number of grid points in the X, Y, and Z directions at which the nonuniform flow field is specified (IX2, IY2, and IZ2) are the dimensions of the coordinates of these points. These counters also constitute the dimensions of the downwash, sidewash, and velocity magnitude variables at these grid points. Thus, the DIMENSION statement is based upon the values XP(IX2), YP(IY2), and ZP(IZ2); and DOWNV(IX2, IY2, IZ2), SIDEV(IX2, IY2, IZ2), and VMAG(IX2, IY2, IZ2).

PROGRAM C
SECOND JOB STEP
INPUT DATA FROM PUNCHED CARDS
SAMPLE RUN

FORCE
COEFF IBC=0 M=0.

AMACH
0.0

INGAMA
0

ICF
1

NMP
3

NCHORD(1)
24

NSPAN(1)
5

NCHORD(2)
9

NSPAN(2)
5

NCHORD(3)
9

NSPAN(3)
5

NSCORD(IN)
20

9

9

MTIP(1)
0

MROOT(1)
0

MTIP(2)
1

MROOT(2)
1

MTIP(3)
1

MROOT(3)
1

XCG
-1.6440

YCG
0.0

ZCG
0.0

XM
-1.6440

ZM
0.0

YL
0.0

ZL
0.0

XN
-1.6440

YN
0.0

S
0.5027

BSPAN
0.8000

CBAR
4.3950

NSYM
2

IBC
0

IVEL(1)
1

IVEL(2)
1

IVEL(3)
1

IVEL(4)
0

IPRG
0

IPRVFS
0

IPRVEL
0

IPRGAM
0

IPRCP
1

IPRCF
1

IX1
1

IX2
10

IY1
1

IY2
8

IZ1
1

IZ2
10

```

C      PROGRAM C - 2ND JOB STEP
C      THIS JOB STEP PERFORMS
C      1) READ(5, ) INPUT DATA AND APPLIES GOETHERT TRANSFORMATION
C      2) COMPUTE COMMON/CNET/ AND COMMON/CCOORD/
C      3) WRITE(51) INPUT DATA, COMMON/CNET/, COMMON/CCOORD/
C      4) PASSES HOIAG, HINIV, G ON DISK UNIT(21) TO THE NEXT JOB STEP
C
C      COMMON/CNET/ NWP
C      1 NCHORD(3) , NSPAN(3) , NSCORD(3) , NVDR(3)
C      2 NUC(3,4) , NUS(3,4) , NVEL(3,4) , NUT(4)
C      3 COMMON/CCOORD/ XNET(24,5,3), YNET(24,5,3), ZNET(24,5,3),
C      4 AKSI(23,4,3), ETA(23,4,3), ZETA(23,4,3),
C      5 XFLOW(1,1), YFLOW(1,1), ZFLOW(1,1),
C      6 RUC(156,3), AX(3), AY(3), AZ(3)
C      7 DIMENSION
C      8 MTP(3) , MRDOT(3) , IVEL(4)
C      9 XP(10) , YP(8) , ZP(10)
C      10 DOWNV(10,8,10), SDEV(10,8,10), VMAG(10,8,10)
C      11 DIMENSION CXD(3)
C
C      XP(1X), YP(1Y), ZP(1Z) NONUNIFORM FLOW FIELD GRID POINTS
C      XDRIG, YDRIG, ZDRIG IN PARENT A/C REF. SYSTEM
C      XN, YN, ZN, XL, XL, XN, YN ROTATIONAL AXES COORDINATES
C      XVP, YVP, ZVP IN PARENT A/C REF. SYSTEM
C      XCG, YCG, ZCG ROTATIONAL AXES COORDINATES
C      IN STORE AXES REF. SYSTEM
C      P(POINTS AT WHICH VELOCITY IS COMPUTED
C      IN PARENT A/C REF. SYSTEM
C      STORE CENTER OF GRAVITY
C      IN STORE AXES REF. SYSTEM
C
C      7100 FORMAT( 1615 )
C      7200 FORMAT( RE10.0 )
C      7250 FORMAT( 10ER.0 )
C      7300 FORMAT( ROM
C
C      1
C      READ(5,7300)
C      READ(5,7200) AMACH
C      READ(5,7100) JHAMA, ICF
C      READ(5,7100) NWP
C      READ(5,7100) (NCHORD(IW), NSPAN(IW), IW=1,NWP)
C      READ(5,7100) (NSCORD(IW), IW=1,NWP)
C      READ(5,7100) (MTP(IW), MRDOT(IW), IW=1,NWP)
C      READ(5,7200) XCG, YCG, ZCG
C      READ(5,7200) XN, YN, ZN, XL, XL, XN, YN
C      READ(5,7100) NSYM, IBC
C      READ(5,7100) IVEL(1), IVEL(2), IVEL(3), IVEL(4)
C      READ(5,7100) IPRG
C      1 IPRVFS, IPRVEL, IPRGAM, IPRCP, IPRCF
C
C      NON-UNIFORM FREE STREAM BOUNDARY CONDITIONS
C      1 READ(5,7100) IX1, IX2, IY1, IY2, IZ1, IZ2
C      FORMULAS FOR XP(1X), YP(1Y), ZP(1Z) ARE PECULIAR TO M-117 BOMB
C
C      DD 110 IX=IX1,IX2
C      1110 XP(1X) = -18. + IX - IX1
C      DD 1120 IY=IY1,IY2
C      1120 YP(1Y) = -7. + IY - IY1
C      DD 1130 IZ=IZ1,IZ2
C      1130 ZP(1Z) = FLOW(IZ)
C      DD 1140 IY=IY1,IY2
C      DD 1140 IZ=IZ1,IZ2
C      1140 READ(5,7250) (DOWNV(IX,IY,IZ), IX=IX1,IX2)
C      DD 1150 IY=IY1,IY2
C      DD 1150 IZ=IZ1,IZ2
C      1150 READ(5,7250) (SDEV(IX,IY,IZ), IX=IX1,IX2)
C      DD 1160 IY=IY1,IY2
C      DD 1160 IZ=IZ1,IZ2
C      1160 READ(5,7250) (VMAG(IX,IY,IZ), IX=IX1,IX2)
C
C      FOLLOWING STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
C      RSPAN = 8
C      S = 3.1416 * (RSPAN/2)**2
C      CRAR = 4.395
C
C      XNET(5,1,1) = 4.395
C      XNET(9,1,1) = 3.435
C      XNET(11,1,1) = 2.935
C      XNET(15,1,1) = 2.035
C      XNET(19,1,1) = 1.035
C      XNET(24,1,1) = 0.
C
C      DD 1254 I=1,R
C      1254 XNET(1,1,1) = XNET(5,1,1) + (1-5) * (XNET(11,1,1)-XNET(5,1,1))
C      1 XNET(10,1,1) = (XNET(9,1,1) + XNET(11,1,1)) / 2.
C
C      DD 1256 I=12,14
C      1256 XNET(1,1,1) = XNET(11,1,1) + (J-11) * (XNET(15,1,1)-XNET(11,1,1))
C      1 (15-11)
C
C      DD 1258 I=16,18
C      1258 XNET(1,1,1) = XNET(15,1,1) + (J-15) * (XNET(19,1,1)-XNET(15,1,1))
C      1 (19-15)
C
C      DD 1260 I=20,23
C      1260 XNET(1,1,1) = XNET(19,1,1) + (J-19) * (XNET(24,1,1)-XNET(19,1,1))
C      1 (24-19)
C
C      1 NNC = NCHORD(1)
C      1 NNS = NSPAN(1)
C      DD 1262 I=1,NNC
C      DD 1262 J=2,NNS
C      1262 XNET(1,J,1) = XNET(1,1,1)
C
C      XO = 1.055
C      ZO = - SORT( 1.6**2 - XO**2 )
C      ZNET(1,1,1) = -.2815
C      ZNET(11,1,1) = -.2815
C      ZNET(19,1,1) = -.4
C      ZNET(24,1,1) = 0.
C
C      DD 1294 I=1,NNC
C      1294 IF( I-11 ) 1282, 1282, 1272
C      1272 IF( I-15 ) 1284, 1284, 1274
C      1274 IF( I-19 ) 1286, 1286, 1276
C      1276 IF( I-24 ) 1288, 1290, 1290
C
C      I=1,11
C      1282 ZCL = - ZNET(1,1,1)

```

```

      GO TO 1292
C
1284 I=12,15
      ZCL = ZNET(11,1,1) * (1-11) * IZNET(15,1,1) - ZNET(11,1,1)
      ZCL = - ZCL
      GO TO 1292
C
1286 I=16,19
      ZCL = - ZNET(15,1,1)
      GO TO 1292
C
1288 I=20,23
      ZCL = + ZN
      GO TO 1292
      + SORT( 1.6**2 - (XNET(1,1,1)-XOI**2 )
C
1290 I=24
      ZCL = 0.
C
1292 YNET(1,1,1) = 0.
      YNET(1,2,1) = ZCL * SIN( 3.1416 / 4. I
      YNET(1,3,1) = ZCL
      YNET(1,4,1) = ZCL * SIN( 3.1416 / 4. I
      YNET(1,5,1) = 0.
      ZNET(1,1,1) = - ZCL
      ZNET(1,2,1) = - ZCL * SIN( 3.1416 / 4. I
      ZNET(1,3,1) = 0.
      ZNET(1,4,1) = ZCL * SIN( 3.1416 / 4. I
      ZNET(1,5,1) = ZCL
1294 CONTINUE
      IF( NWP-I I 1380, 1380, 1300
1300 NNC = NCHORD(2)
      NNCM1 = NCHORD(2)-1
      NNS = NSPAN(2)
      NNSM1 = NSPAN(2)-1
      XNET(1,1,2) = 3.9450
      XNET(1,NNS,2) = 3.4350
      XNET(NNC,1,2) = 4.3950
      XNET(NNC,NNS,2) = 4.3950
      DO 1312 I=1,NNC
      XNET(I,1,2) = XNET(1,1,2) + ( XNET(NNC,1,2) - XNET(1,1,2) I
      * FLOAT(I-1) / (NNC-1)
      XNET(I,NNS,2) = XNET(1,NNS,2)
      + ( XNET(NNC,NNS,2) - XNET(1,NNS,2) I
      * FLOAT(I-1) / (NNS-1)
      DO 1310 J=1,NNS
1310 XNET(I,J,2) = XNET(I,1,2) + I XNET(I,NNS,2) - XNET(I,1,2) I
      * FLOAT(J-1) / (NNS-1)
1312 CONTINUE
C
      YNET(1,1,2) = 0.4020
      YNET(1,NNS,2) = 0.1991
      ZNET(1,1,2) = -YNET(1,1,2)
      ZNET(1,NNS,2) = -YNET(1,NNS,2)
      DO 1350 J=1,NNS
      YNET(I,J,2) = YNET(1,1,2) + ( YNET(1,NNS,2) - YNET(1,1,2) I
      * FLOAT(J-1) / (NNS-1)
1350 ZNET(I,J,2) = ZNET(1,1,2) + ( ZNET(1,NNS,2) - ZNET(1,1,2) I
      * FLOAT(J-1) / (NNS-1)
C
      DO 1360 I=2,NNC
      DO 1360 J=1,NNS
      YNET(I,J,2) = YNET(1,J,2)
      ZNET(I,J,2) = ZNET(1,J,2)
C
      DO 1370 I=1,NNC
      DO 1370 J=1,NNS
      XNET(I,J,3) = XNET(I,J,2)
      YNET(I,J,3) = YNET(I,J,2)
      ZNET(I,J,3) = -ZNET(I,J,2)
1370 CONTINUE
C
      DO 1420 IW=1,NWP
      NNC = NCHORD(IW)
      NNS = NSPAN(IW)
      DO 1420 I=1,NNC
      DO 1420 J=1,NNS
      XNET(I,J,IW) = - XNET(I,J,IW)
      YNET(I,J,IW) = - YNET(I,J,IW)
1420 ABOVE STATEMENTS ARE PECULIAR TO THE MODELING OF THE M-117 BOMB
C
C
C
      BETA = SORT( 1. - AMACH**2 )
      S = BETA**2 * 5
      RSPAN = BETA * RSPAN
      YCG = BETA * YCG
      ZCG = BETA * ZCG
      YL = BETA * YL
      YN = BETA * YN
      ZL = BETA * ZL
      ZM = BETA * ZM
      DO 1500 IW=1,NWP
      NNC = NCHORD(IW)
      NNS = NSPAN(IW)
      DO 1500 I=1,NNC
      DO 1500 J=1,NNS
      YNET(I,J,IW) = BETA * YNET(I,J,IW)
      ZNET(I,J,IW) = BETA * ZNET(I,J,IW)
1500 CONTINUE
C
      DO 1520 IY=1,IY2
      YP(IY) = BETA * YP(IY)
      DO 1530 IZ=1,IZ2
      ZP(IZ) = BETA * ZP(IZ)
1530 CONTINUE
C
      DO 1550 IX=1,IX2
      DO 1550 IY=1,IY2
      DO 1550 IZ=1,IZ2
      DDWNV(IX,IY,IZ) = BETA * DDWNV(IX,IY,IZ)
      SIOFV(IX,IY,IZ) = BETA * SIOFV(IX,IY,IZ)
1550 CONTINUE
C
      WRITE(6,7300)
      WRITE(6,8004)
      R004 FORMAT( /
      WRITE(6,8006)
      R006 FORMAT( /
      WRITE(6,8010)
      AMACH
      AMACH
      /6F20.2I
      IHGAMA,
      IHGAMA
      ICF /6I20 I
      NWP

```

```

R010 FORMAT(/,
1 WRIT(16,R020) ,
NCHORD(1), NSPAN(1),
R020 FORMAT(/,
1 NCHORD(2) ,
NSPAN(2) , NCHORD(3) , NSPAN(3) ,
2 NSPAN(4) ,
R030 FORMAT(/,
1 WRIT(16,R030) ,
NCHORD(1), NSPAN(1),
R040 FORMAT(/,
1 WRIT(16,R040) ,
NCHORD(1), NSPAN(1),
R040 FORMAT(/,
1 MTIP(2) ,
MRONT(2) , MTIP(3) ,
2 MRONT(3) ,
1 WRIT(16,R040) ,
XCG, YCG,
R060 FORMAT(/,
1 ZCG ,
XCG, YCG,
1 WRIT(16,R050) ,
XM, ZM, YL, ZL, XN, YN,
R050 FORMAT(/,
1 YL ,
ZL, XN,
2 YN ,
WRIT(16,R070) ,
S,
R070 FORMAT(/,
1 CBAR ,
S,
WRIT(16,R080) ,
NSYM,
R080 FORMAT(/,
1 WRIT(16,R090) ,
ISOLVE1, ISOLVE2,
R090 FORMAT(/,
1 IVEL13 ,
IVEL14 , IVEL11 ,
2 WRIT(16,R100) ,
IPRG,
R100 FORMAT(/,
1 IPRVFS, IPRVEL, IPRGAM, IPRCP, IPRCF,
2 IPRVFS, IPRVEL, IPRGAM, IPRCP, IPRCF,
1 WRIT(16,R110) ,
IX1, IX2, IY1, IY2, IZ1, IZ2,
R110 FORMAT(/,
1 IY1 ,
IY2, IZ1, IZ2,
2 IZ2 ,
DN 2130 IW=1,NWP
NUC(IW,1) = NCHORD(IW)-1
NUS(IW,1) = NSPAN(IW)-1
NUC(IW,2) = NCHORD(IW)
NUS(IW,2) = NSPAN(IW)
NUC(IW,3) = NCHORD(IW)-1
NUS(IW,3) = NSPAN(IW)
C
2130 NVOR(1) = (NCHORD(11-1) * (NSPAN(11-1)
ON 2140 ISOLVE=1,3
NVEL(IW,NWP-1) = NUC(1,ISOLVE) * NUS(1,ISOLVE)
2140 NVOR(1) = 2130, 2140
ON 2150 IW=2,NWP
NVOR(IW) = NVOR(IW-1) + (NCHORD(IW-1) * (NSPAN(IW-1)
2150 NVEL(IW,ISOLVE) = NVEL(IW-1,ISOLVE)
2150 NVEL(IW,ISOLVE) = NVEL(IW,NWP,ISOLVE) * NUS(IW,ISOLVE)
ON 2160 ISOLVE=1,3
NUT(IW,ISOLVE) = NUC(IW,ISOLVE) * NUS(IW,ISOLVE)
2160 NVORT = NVOR(NWP)
C
ON 2670 IW=1,NWP
NVORC = NCHORD(IW)-1
NVORC = NSPAN(IW)-1
DO 2650 IP=1,NVORC
I = IP
DO 2650 IO=1,NVORS
J = IO
IF (IW-1) 2610, 2610, 2620
2610 IT = (IP-1)*NVORS + IO
2620 IT = NVOR(IW-1) + (IP-1)*NVORS + IO
2630 CONTINUE
AKSII(IP,IO,IW) = ( XNET(I,J,IW) + XNET(I,J+1,IW)
1 ETA(IP,IO,IW) = ( XNET(I+1,J,IW) + XNET(I+1,J+1,IW) I / 4.
1 ZETA(IP,IO,IW) = ( XNET(I+1,J,IW) + XNET(I+1,J+1,IW) I / 4.
1 ZETA(IP,IO,IW) = ( XNET(I+1,J,IW) + XNET(I+1,J+1,IW) I / 4.
CX = XNET(I,J+1,IW) - XNET(I+1,J,IW)
CY = XNET(I,J+1,IW) - XNET(I+1,J,IW)
CZ = XNET(I,J+1,IW) - XNET(I+1,J,IW)
OX = XNET(I,J,IW) - XNET(I+1,J+1,IW)
DY = XNET(I,J,IW) - XNET(I+1,J+1,IW)
DZ = XNET(I,J,IW) - XNET(I+1,J+1,IW)
CXD(1) = CY * DZ - OY * CZ
CXD(2) = - CX * DZ + OX * CZ
CXD(3) = CX * DY - OX * CY
ARSCD = SQRT( CXD(1)**2 + CXD(2)**2 + CXD(3)**2 )
2650 RDC(I1,N) = CXD(N) / ARSCD
C
2670 CONTINUE
C
WRITE(6,R506)
R506 FORMAT(1, 2X ,
1W FIELD ,
CONTROL VOLUME GRID POINTS. NONUNIFORM FLO
R510 WRITE(6,R510) ,
(XP(IX), IX=IX1,IX2)
R510 FORMAT( 3X ,
XP(IX)=, 10F10.5 )
R520 WRITE(6,R520) ,
(YP(IY), IY=IY1,IY2)
R520 FORMAT( 3X ,
YP(IY)=, 10F10.5 )
R530 WRITE(6,R530) ,
(ZP(IZ), IZ=IZ1,IZ2)
R530 FORMAT( 3X ,
ZP(IZ)=, 10F10.5 )
DO 8616 IX=IX1,IX2
WRITE(6,R610) ,
XP(IX)
R610 FORMAT(/ 3X ,
DOWNWASH (DEG) AT PARENT AIRCRAFT X-STA
1 IX=, IZ ,
XP(IX)=, F10.5 I
DO 8612 IZ=IZ1,IZ2
WRITE(6,R614) ,
ZP(IZ), (DOWNWV(IX,IY,IZI, IY=IY1,IY2I
R614 FORMAT( , IZ=, IZ, 2X, , ZP(IZ)=, F10.5 ,
1 IZ ,
(DOWNWV, IY=IY1,IY2I =, 8F7.2 I
R616 CONTINUE
C
DO 8626 IX=IX1,IX2
WRITE(6,R620) ,
XP(IX)
R620 FORMAT( 3X ,
SIDEWASH (DEG) AT PARENT AIRCRAFT X-STA
1 IX=, IZ ,
XP(IX)=, F10.5 I
DO 8622 IZ=IZ1,IZ2
WRITE(6,R624) ,
ZP(IZ), (SIDEV(IX,IY,IZI, IY=IY1,IY2I

```

```

      A624 FORMAT( ' IZ=',I2, 2X, ' ZP(I2)=',F10.5, ' 15X,
      A626 CONTINUE ' (SIDEV, IY=IY1,IY2) =',8F7.2 )
C
      DO A636 IX=IX1,IX2
      WRITE(6,A630) IX, XP(IX)
      A630 FORMAT(// 3X, ' VELOCITY MAGNITUDE AT PARENT AIRCRAFT X-STA
      1 IX=',I2, IZ=IZ1, IZ2 ZP(IZ), IVMAG [(IX,IY,IZ), IY=IY1,IY2)
      A632 WRITE(6,A634) IZ, ZP(IZ), F10.5, 15X,
      A634 FORMAT( ' IZ=',I2, 2X, ' (VMAG, IY=IY1,IY2) =',8F7.4 )
      A636 CONTINUE
C
      DO A290 IW=1,NWP
      NNC = NCHORD(IW)
      NNS = NSPAN(IW)
      GO TO( A210, A220, A230), IW
      A210 WRITE(6,A212) IW
      A212 FORMAT(// ' NETWORK POINTS ON BODY J=1 IS TOP (-Z) CL J=NJ IS BOTTOM (+Z) CL 'I
      1 GO TO A250
      A220 WRITE(6,A222) IW
      A222 FORMAT(// ' NETWORK POINTS ON UPPER FIN J=1 IS TIP J=NJ IS ROOT 'I
      1 GO TO A250
      A230 WRITE(6,A232) IW
      A232 FORMAT(// ' NETWORK POINTS ON LOWER FIN J=1 IS TIP J=NJ IS ROOT 'I
      1
      A250 WRITE(6,A260) XNET(I,J,IW)
      A260 FORMAT(// ' XNET(I,J,IW) 'I
      DO A262 I=1,NNC
      A262 WRITE(6,A264) I, (XNET(I,J,IW), J=1,NNS)
      A264 FORMAT(// ' I=',I2, 9F10.4 )
      WRITE(6,A270) YNET(I,J,IW)
      A270 FORMAT(// ' YNET(I,J,IW) 'I
      DO A272 I=1,NNC
      A272 WRITE(6,A274) I, (YNET(I,J,IW), J=1,NNS)
      A274 FORMAT(// ' I=',I2, 9F10.4 )
      WRITE(6,A280) ZNET(I,J,IW)
      A280 FORMAT(// ' ZNET(I,J,IW) 'I
      DO A282 I=1,NNC
      A282 WRITE(6,A284) I, (ZNET(I,J,IW), J=1,NNS)
      A284 FORMAT(// ' I=',I2, 9F10.4 )
      A290 CONTINUE
C
      DO A388 IW=1,NWP
      NVORC = NCHORD(IW)-1
      NVORS = NSPAN(IW)-1
      WRITE(6,A310) IW
      A310 FORMAT(// ' BOUNDARY POINTS IW=',I1 )
      A360 WRITE(6,A360) AKSI(IP,IO,IW)
      A360 FORMAT(// ' AKSI(IP,IO,IW) 'I
      DO A362 IP=1,NVORC
      A362 WRITE(6,A364) IP, (AKSI(IP,IO,IW), IO=1,NVORS)
      A364 FORMAT(// ' IP=',I2, 8F10.5 )
      WRITE(6,A370) ETA(IP,IO,IW)
      A370 FORMAT(// ' ETA(IP,IO,IW) 'I
      DO A372 IP=1,NVORC
      A372 WRITE(6,A374) IP, (ETA(IP,IO,IW), IO=1,NVORS)
      A374 FORMAT(// ' IP=',I2, 8F10.5 )
      WRITE(6,A380) ZETA(IP,IO,IW)
      A380 FORMAT(// ' ZETA(IP,IO,IW) 'I
      DO A382 IP=1,NVORC
      A382 WRITE(6,A384) IP, (ZETA(IP,IO,IW), IO=1,NVORS)
      A384 FORMAT(// ' IP=',I2, 8F10.5 )
      A388 CONTINUE
      A410 WRITE(6,A410)
      A410 FORMAT(// ' ADC(IT,N) DIRECTION COSINES AT BOUNDARY POINTS
      1 N=1 IS X-COMPONENT N=2 IS Y N=3 IS Z 'I
      DO A420 IW=1,NWP
      NVORC = NCHORD(IW)-1
      NVORS = NSPAN(IW)-1
      DO A420 N=1,3
      DO A420 IP=1,NVORC
      IF( IW-1 ) 8412, 8412, 8414
      8412 IT1 = (IP-1)*NVORS + 1
      IT2 = IP * NVORS
      GO TO 8416
      8414 IT1 = NVOR(IW-1) + (IP-1)*NVORS + 1
      IT2 = NVOR(IW-1) + IP * NVORS
      A416 WRITE(6,A418) IW, N, IP, (DOC(IT,N), IT=IT1,IT2)
      A418 FORMAT(// ' IW=',I1, ' N=',I1, ' IP=',I2, 8F10.5 )
      A420 CONTINUE
C
      WRITE(51)
      1 AMACH ' ICF '
      2 IMGAMA ' NSPAN '
      3 NWP ' WROOT '
      4 NCHORD ' YCG '
      5 NSCORD ' ZL '
      6 RTIP ' CBAR '
      7 XCG ' ZM '
      8 XM ' YL '
      9 S ' XN '
      A NSYM ' YN '
      A IVEL ' IRC '
      B IPRG ' IPRVFS, IPRVEL, IPRGAM, IPRCP, IPRCF '
      B IX1 ' IY1 '
      B IX2 ' IY2 '
      F XP ' IZ1, IZ2 '
      F NNWNV ' YP '
      F ' SIDEV '
      F ' VMAG '
      WRITE(51)
      1 NUS ' NVOR '
      2 XNET ' NUT '
      3 AKSI ' ZNET '
      4 ' ZETA '
      END FILE 51
      REWIND 51
C
      STOP
      END

```

NETWORK POINTS ON BODY

IW=1 J=1 IS TOP 1-21 CL J=NJ IS BOTTOM 1-21 CL												7NAT11,J,IW1					
XNFT11,J,IW1						YNFT11,J,IW1											
I= 1	-5.3683	-5.3683	-5.3683	-5.3683	-5.3683	I= 1	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 1	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 2	-5.1250	-5.1250	-5.1250	-5.1250	-5.1250	I= 2	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 2	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 3	-4.8817	-4.8817	-4.8817	-4.8817	-4.8817	I= 3	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 3	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 4	-4.6383	-4.6383	-4.6383	-4.6383	-4.6383	I= 4	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 4	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 5	-4.3950	-4.3950	-4.3950	-4.3950	-4.3950	I= 5	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 5	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 6	-4.1517	-4.1517	-4.1517	-4.1517	-4.1517	I= 6	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 6	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 7	-3.9083	-3.9083	-3.9083	-3.9083	-3.9083	I= 7	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 7	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 8	-3.6650	-3.6650	-3.6650	-3.6650	-3.6650	I= 8	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 8	-0.2815	-0.1991	0.0	0.1991	0.2815
I= 9	-3.4217	-3.4217	-3.4217	-3.4217	-3.4217	I= 9	0.0	-0.1991	-0.2815	-0.1991	0.0	I= 9	-0.2815	-0.1991	0.0	0.1991	0.2815
I=10	-3.1783	-3.1783	-3.1783	-3.1783	-3.1783	I=10	0.0	-0.1991	-0.2815	-0.1991	0.0	I=10	-0.2815	-0.1991	0.0	0.1991	0.2815
I=11	-2.9350	-2.9350	-2.9350	-2.9350	-2.9350	I=11	0.0	-0.1991	-0.2815	-0.1991	0.0	I=11	-0.2815	-0.1991	0.0	0.1991	0.2815
I=12	-2.6917	-2.6917	-2.6917	-2.6917	-2.6917	I=12	0.0	-0.2200	-0.3111	-0.2200	0.0	I=12	-0.3111	-0.2200	0.0	0.2200	0.3111
I=13	-2.4483	-2.4483	-2.4483	-2.4483	-2.4483	I=13	7.0	-0.2409	-0.3407	-0.2409	0.0	I=13	-0.3407	-0.2409	0.0	0.2409	0.3407
I=14	-2.2050	-2.2050	-2.2050	-2.2050	-2.2050	I=14	0.0	-0.2819	-0.3704	-0.2819	0.0	I=14	-0.3704	-0.2819	0.0	0.2819	0.3704
I=15	-2.0350	-2.0350	-2.0350	-2.0350	-2.0350	I=15	0.0	-0.2828	-0.4000	-0.2828	0.0	I=15	-0.4000	-0.2828	0.0	0.2828	0.4000
I=16	-1.7917	-1.7917	-1.7917	-1.7917	-1.7917	I=16	0.0	-0.2828	-0.4000	-0.2828	0.0	I=16	-0.4000	-0.2828	0.0	0.2828	0.4000
I=17	-1.5483	-1.5483	-1.5483	-1.5483	-1.5483	I=17	0.0	-0.2828	-0.4000	-0.2828	0.0	I=17	-0.4000	-0.2828	0.0	0.2828	0.4000
I=18	-1.3050	-1.3050	-1.3050	-1.3050	-1.3050	I=18	0.0	-0.2828	-0.4000	-0.2828	0.0	I=18	-0.4000	-0.2828	0.0	0.2828	0.4000
I=19	-1.0617	-1.0617	-1.0617	-1.0617	-1.0617	I=19	0.0	-0.2828	-0.4000	-0.2828	0.0	I=19	-0.4000	-0.2828	0.0	0.2828	0.4000
I=20	-0.8183	-0.8183	-0.8183	-0.8183	-0.8183	I=20	0.0	-0.2711	-0.3834	-0.2711	0.0	I=20	-0.3834	-0.2711	0.0	0.2711	0.3834
I=21	-0.6348	-0.6348	-0.6348	-0.6348	-0.6348	I=21	0.0	-0.2411	-0.3409	-0.2411	0.0	I=21	-0.3409	-0.2411	0.0	0.2411	0.3409
I=22	-0.4232	-0.4232	-0.4232	-0.4232	-0.4232	I=22	0.0	-0.1888	-0.2671	-0.1888	0.0	I=22	-0.2671	-0.1888	0.0	0.1888	0.2671
I=23	-0.2116	-0.2116	-0.2116	-0.2116	-0.2116	I=23	0.0	-0.1108	-0.1568	-0.1108	0.0	I=23	-0.1568	-0.1108	0.0	0.1108	0.1568
I=24	0.0	0.0	0.0	0.0	0.0	I=24	0.0	0.0	0.0	0.0	0.0	I=24	0.0	0.0	0.0	0.0	0.0

NETWORK POINTS ON UPPER FIN

IW=2 J=1 IS TIP J=NJ IS ROOT					
XNFT11,J,IW1					
I= 1	-3.9450	-3.9450	-3.9450	-3.9450	-3.9450
I= 2	-4.0012	-4.0012	-4.0012	-4.0012	-4.0012
I= 3	-4.0575	-4.0575	-4.0575	-4.0575	-4.0575
I= 4	-4.1137	-4.1137	-4.1137	-4.1137	-4.1137
I= 5	-4.1700	-4.1700	-4.1700	-4.1700	-4.1700
I= 6	-4.2262	-4.2262	-4.2262	-4.2262	-4.2262
I= 7	-4.2825	-4.2825	-4.2825	-4.2825	-4.2825
I= 8	-4.3387	-4.3387	-4.3387	-4.3387	-4.3387
I= 9	-4.3950	-4.3950	-4.3950	-4.3950	-4.3950
YNFT11,J,IW1					
I= 1	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 2	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 3	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 4	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 5	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 6	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 7	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 8	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 9	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
7NFT11,J,IW1					
I= 1	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 2	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 3	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 4	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 5	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 6	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 7	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 8	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 9	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991

NETWORK POINTS ON LOWER FIN

IW=3 J=1 IS TIP J=NJ IS ROOT					
XNFT11,J,IW1					
I= 1	-3.9450	-3.9450	-3.9450	-3.9450	-3.9450
I= 2	-4.0012	-4.0012	-4.0012	-4.0012	-4.0012
I= 3	-4.0575	-4.0575	-4.0575	-4.0575	-4.0575
I= 4	-4.1137	-4.1137	-4.1137	-4.1137	-4.1137
I= 5	-4.1700	-4.1700	-4.1700	-4.1700	-4.1700
I= 6	-4.2262	-4.2262	-4.2262	-4.2262	-4.2262
I= 7	-4.2825	-4.2825	-4.2825	-4.2825	-4.2825
I= 8	-4.3387	-4.3387	-4.3387	-4.3387	-4.3387
I= 9	-4.3950	-4.3950	-4.3950	-4.3950	-4.3950
YNFT11,J,IW1					
I= 1	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 2	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 3	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 4	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 5	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 6	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 7	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 8	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 9	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
7NFT11,J,IW1					
I= 1	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 2	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 3	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 4	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 5	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 6	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 7	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 8	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991
I= 9	-0.4020	-0.3513	-0.3005	-0.2498	-0.1991

Output Data

The output data, which is passed to the following job step using data set reference number 51, consists of the inputs to this step plus arrays that are functions of the geometry and vortex modeling.

2.5 USER'S GUIDE TO PROGRAM C—THIRD JOB STEP

Input Data

The following are input data for the optional version of the third job step of Program C, in which trajectories are computed.

LM	Object-time dimension of the dependent variables of the equations of motion
AMACH	Free-stream Mach number
DT(1)	Time interval of half of a numerical integration step, sec
NCASE	Number of vortex-lattice cases to be computed with each transfer of data from data set reference number 21 into the computer core
NK	Number of time steps plus 1 in the trajectory; NK-1 must be a multiple of NCASE-1
SCALE	Ratio of full-scale store dimensions to store model dimensions
WGTTA	Components of full-scale store weight along X, Y, and Z wind axes, lb
S	Full-scale store reference area, ft ²
BSPAN	Full-scale store lateral reference length, ft
CBAR	Full-scale store longitudinal reference length, ft
AIX, AIY, AIZ	Full-scale store mass moments of inertia about body axes, slug-ft ²
AIXZ	Full-scale store cross product of mass moment of inertia in the body x-z plane, slug-ft ²
RHO	Free-stream density, slug/ft ³
ASOUND	Speed of sound, ft/sec

FX1, FX2	Thrust forces acting along full-scale store body x-axis (see Fig. 12, Ref. 4), lb
FZ1, FZ2	Ejector forces acting on full-scale store (see Fig. 12, Ref. 4), lb
ALX1, ALX2	Body x-axis coordinates of full-scale store ejection stations (see Fig. 12, Ref. 4), ft
CLP	Roll-damping coefficient (see p. viii of Ref. 4)
CMQ	Pitch-damping coefficient (see p. viii of Ref. 4)
CNR	Yaw-damping coefficient (see p. viii of Ref. 4)
U(1), V(1), W(1)	Initial values of full-scale store translational velocity components along body x axis, y axis, and z axis, respectively, ft/sec
P(1), Q(1), R(1)	Initial values of full-scale store rotational velocity components about body x axis, y axis, and z axis, respectively, radians/sec
X(1), Y(1), Z(1)	Initial values of full-scale store center of gravity coordinates in wind axes, ft
ANU(1), PSI(1), OMEGA(1)	Initial values of angular displacement about the store body axes in pitch, yaw, and roll, respectively, radians/sec
CF(1, 1), ..., CF(1, 6)	Initial values of aerodynamic force coefficients in the sequence axial force, side force, normal force, pitching moment, yawing moment, and rolling moment
CA1	Coefficient used to correct the axial-force coefficient in the predictor-corrector numerical integration scheme
CLA	Coefficient used to correct the normal- and side-force coefficients in the predictor-corrector numerical integration scheme
CMA	Coefficient used to correct the pitching- and yawing-moment coefficients in the predictor-corrector numerical integration scheme

The following are input data for the optional version of the third job step of Program C, in which force coefficients, but not trajectories, are computed.

NCASE	Number of vortex-lattice cases to be computed with each transfer of data from data set reference Number 21 into the core
NREAD	Number of times data set reference Number 21 is read
X, Y, Z	Coordinates of the point which the store body axes are rotated about with respect to the wind axes, in the parent aircraft reference system, ft
ANU, PSI, OMEGA	Angular displacement about the store body axes in pitch, yaw, and roll, respectively, radians/sec

The following input data are defined by arithmetic expressions:

ICF = 1	Laminar skin friction assumed
= 2	Turbulent skin friction assumed
BSPAN	Full-scale store lateral reference length, ft
CBAR	Full-scale store longitudinal reference length, ft

Dimensions of Arrays

Whenever the number 195 appears as a dimension in the program listing, this is the number of points at which velocity is calculated when ISOLVE = 3 (NUT(3)). The arithmetic expression for NUT(3) is given in the second job step of Program C.

NCASE (which has the value 5 in the sample program listing) is a dimension for the following variables: CDTOT, CYTOT, CLTOT, CMTOT, CNTOT, CRTOT, BDOTV, GAMMAV, HGAMMA, VX, VY, and VZ.

In SUBROUTINE ACOEFF the elements of the vorticity distribution in the GAMMAV array are reordered to form the GAMMA array. The maximum values of the three subscripts of GAMMA are, respectively, (1) the maximum value of NCHORD(IW), (2) the maximum value of NSPAN(IW) + 1, and (3) NWP. In this same subroutine the dimensions of the variables which are the spanwise distributions of force coefficients on vortex segments are, respectively, (1) the maximum value of NCHORD(IW) and (2) the maximum value of NSPAN(IW)-1. The dimensions of the variables which represent chordwise distributions are, respectively, (1) the maximum value of NCHORD(IW)-1 and (2) the maximum value of NSPAN(IW).

```

C PROGRAM C - 3RD JOM STEP
C THIS PROGRAM INTEGRATES THE SIX-DEGREE-OF-FREEDOM
C EQUATIONS OF MOTION USING A 4TH ORDER RUNGE-KUTTA METHOD
C DIMENSION
1 L(4) , DT (33) ,
2 A(3,3) , WGT(13) ,
3 WGT(13) , WGT(13) ,
4 CF(13,4) , DCF(16,2) , PITCH(33) , YAMP (33) ,
4 AINCP(33) , AINCP (33) ,
5 CA(33) , CY(33) , CN(33) , CPITCH(33) , CYAW(33) , CRD(13,33) ,
6 CROT(15) , CYROT(15) , CLROT(15) , CROT(15) , CROT(15) ,
8 RDOT(4) , VDOT(4) , WDOT(4) , RDOT(4) , RDOT(4) ,
9 XDOT(4) , YDOT(4) , ZDOT(4) , ANUDOT(4) , PSIDOT(4) , OMEGADOT(4) ,
A U(33) , V(33) , W(33) , P(33) , Q(33) , R(33) ,
R X(33) , Y(33) , Z(33) , ANU(33) , PSI(33) , OMEGA(33) ,
CALL ERKSET (209,10,5,2)
CALL ERKSET (251,10,5,2)
CALL ERKSET (252,10,5,2)
CALL ERKSET (253,10,5,2)
C ALTITUDE = 5000.
C TEMP = 41.2 DEG F
C ASOUND = 1097.5
C EJECTOR FORCE = 1000.
C EJECTOR STROKE = .2552
C XCG FULL SCALE = 2.74
C INPUTS SUBJECT TO FREQUENT CHANGE
C U(1),...,OMEGA(11), CF(1,JF)
C LM = 33
C NK = 17
C NM = 7*NM - 1
C NCASE = 5
C INCRFM = (NCASE-1) / 2
C ANACH = 0
C DT(1) = .1 / 4.
C DO 500 M=1,NM
500 DT(M) = DT(1)
C THE FOLLOWING STATEMENTS ARE PECULIAR TO THE M=117 ROMB
C SCALF = 20.
C WGT(11)= 0.
C WGT(12)= 0.
C WGT(13)= 250.
C AMRAR = WGT(13) / 32.2
C S = 1.395
C RSPAN = 1.333
C CHA4 = 1.325
C AIX = 4.
C AIY = 30.
C AIZ = 30.
C AIRZ = 0.
C RHO = .001978
C ASOUND = 1097.5
C U1 = ANACH * ASDUNO
C THRUST AND EJECTOR FORCES
C FX1 = 0.
C FX2 = 0.
C FZ1 = 0.
C FZ2 = 0.
C ALX1 = 0.
C ALX2 = 0.
C DAMPING COEFFICIENTS
C CLP = 0.
C CMQ = -2.319
C CNR = -2.319
C XCARIG = SCALE * (-12.99-1.644) / 12.
C YCARIG = SCALE * (-4.66) / 12.
C ZCARIG = SCALE * 1.1 / 12.
C GO TO 600
C 600 CONTINUE
C INITIAL CONDITIONS FOR ROMB RELEASE
C P(1) = 0.
C Q(1) = 0.
C R(1) = 0.
C IF OMEGA(11)=3.1416/4. BOTH ANU(1) AND PSI(1) .NE. 0.
C ANU(1) = -.7 * 3.1416 / 180.
C PSI(1) = 0.
C OMEGA(11) = 3.1416 / 4.
C OMEGA(11) = 0.
C GO TO 620
C VERTICAL EJECTION OF ROMB
C U(1) = 0.
C V(1) = 0.
C W(1) = 0.
C X(1) = XCARIG
C Y(1) = YCARIG
C Z(1) = ZCARIG + .2552
C GO TO 650
C 620 CONTINUE
C 45 DEGREE EJECTION OF ROMB
C U(1) = 0.
C V(1) = -8.79 / SORT( 2. )
C W(1) = 8.79 / SORT( 2. )
C X(1) = XCARIG
C Y(1) = YCARIG
C Z(1) = ZCARIG + .2552 / SORT( 2. )
C CF(1,1) = .1346
C CF(1,2) = .0485
C CF(1,3) = .0032
C CF(1,4) = -.0008
C CF(1,5) = -.0128
C CF(1,6) = 0.
C GO TO 650
C INITIAL CONDITIONS FOR CONTINUATION OF TRAJECTORY CALCULATIONS
C U(1),...,OMEGA(11), CF(1,JF)
C 650 CONTINUE
C STATIC STABILITY DERIVATIVES
C CLA = D(CL)/D(ANU)

```

DIV CMK
NEG SORT
RIG EXPN
NEG LG10

```

CMA = D1CM/D1ANU)
CVP = D1CV/D1PSI)
CNP = D1CN/D1PSI)
R10 IF(ARS(AMACH-0.5) - .1) R20, 820, 810
R20 IF(ARS(AMACH-0.85) - .1) R30, 830, R40
CMA = -.0636 * 180. / 3.1416
CA1 = -.0169 * 180. / 3.1416
CA1 = (.134-.064)/10.**2) * (180./3.1416)**2
GO TO 850

C 830 CLA = (.39/5.7) * 180. / 3.1416
CMA = -.1/5.71 * 180. / 3.1416
CA1 = (.089-.066)/5.7**2) * (180./3.1416)**2
GO TO 850

C R40 WRITE(6,R122)
R122 FORMAT(1X, MACH NO. NOT PROPERLY INPUT CAUSING PROGRAMMED STOP)
STOP
THE ABOVE STATEMENTS ARE PECULIAR TO THE M-117 BOMB

C 850 CVP = CLA
CNP = CMA
SIDE FOURCF
OCFOA(2,1) = 0.
OCFOA(2,2) = CVP
LIFT
OCFOA(3,1) = CLA
OCFOA(3,2) = 0.
PITCH
OCFOA(4,1) = CMA
OCFOA(4,2) = 0.
YAW
OCFOA(5,1) = 0.
OCFOA(5,2) = CNP
ROLL
OCFOA(6,1) = 0.
OCFOA(6,2) = 0.

C IKJ = 0
L(1) = 0
L(2) = 1
L(3) = 1
L(4) = 2

C DO 6300 KM1=1,MK, INCREM
M1 = 2 * KM1
M2 = 2 * (KM1+1-INCREM) - 1
DO 6200 ITCF=1,2
DO 6100 KM2=1, INCREM
K = KM1 + KM2 - 1
M = 2 * K - 1
GO TO( 2500, 2210), ITCF
IF( KM2-1 ) 2220, 2220, 2500

C 2210 CALL VORLAT( LM, K, M,
1 NCASE
1 IKJ
1 IPI
1 A
1 B
1 C
CYTOT, CYTOT, CLTOT, ANU, PSI, OMEGA, CRTOT)

C DO 2410 ICASE=1,NCASE
MM = M-1+ICASE
CF(MM,1) = CLTOT(ICASE)
CF(MM,2) = CYTOT(ICASE)
CF(MM,3) = CLTOT(ICASE)
CF(MM,4) = CMTOT(ICASE)
CF(MM,5) = CMTOT(ICASE)
CF(MM,6) = CRTOT(ICASE)
2410 CONTINUE

C WRITE(6,R220)
R220 FORMAT(1X, '2ND ESTIMATE OF AERODYNAMIC COEFFICIENTS FROM',
1 ' VORTEX-LATTICE CALCULATIONS',
2 ' CAXIAL CSIOE CNDRM CPITCH CYAW CROLL')

C DO R230 MM=M1,M2
R230 WRITE(6,R232) MM, (CF(MM,JF), JF=1,6)
R232 FORMAT(1X, 'M=',12, '2K, 6F10.4')
2500 CONTINUE

C KKK = 0
3000 KKK = KKK + 1
M = 2 * K - 1 + L(KKK)
IF( K-KKK ) 3110, 6500, 6500
IF( KKK-1 ) 3500, 3500, 3120
3110 CONTINUE
U(M) = U(2*K-1) + L(KKK) * UDOT(KKK-1) * DT(M-1)
V(M) = V(2*K-1) + L(KKK) * VDOT(KKK-1) * DT(M-1)
W(M) = W(2*K-1) + L(KKK) * WDOT(KKK-1) * DT(M-1)
P(M) = P(2*K-1) + L(KKK) * PDOT(KKK-1) * DT(M-1)
Q(M) = Q(2*K-1) + L(KKK) * QDOT(KKK-1) * DT(M-1)
R(M) = R(2*K-1) + L(KKK) * RDOT(KKK-1) * DT(M-1)
X(M) = X(2*K-1) + L(KKK) * XDOT(KKK-1) * DT(M-1)
Y(M) = Y(2*K-1) + L(KKK) * YDOT(KKK-1) * DT(M-1)
Z(M) = Z(2*K-1) + L(KKK) * ZDOT(KKK-1) * DT(M-1)
ANU(M) = ANU(2*K-1) + L(KKK) * ANUDOT(KKK-1) * DT(M-1)
PSI(M) = PSI(2*K-1) + L(KKK) * PSIDOT(KKK-1) * DT(M-1)
OMEGA(M) = OMEGA(2*K-1) + L(KKK) * OMEGADOT(KKK-1) * DT(M-1)
3500 CONTINUE

C PITCH(M) = ANU(M) + ATAN2( W(M),U(M)+U(M) )
YAW(M) = PSI(M) - ATAN2( V(M),U(M)+U(M) )
A1NC10(M) = ANCOS( COS(PITCH(M)) * COS(YAW(M)) )
GO TO( 4100, 4200), ITCF
ITCF = 1 * KM1 - 1
DO 4100 JF=2,6
DO 4150 JF=2,6
CF(M,JF) = CF(M1,JF) + OCFOA(JF,1) * (PITCH(M)-PITCH(M1))
+ OCFOA(JF,2) * (YAW(M)-YAW(M1))
GO TO 4300

C 4200 ITCF = 2
CF(M,1) = CF(M,1) - CA1 * (A1NC10(M)**2 - A1NCP(M)**2)
DO 4250 JF=2,6

```

```

4250 CF(M,JF) = CF(M,JF) + DCFDAIJF,1) * (PITCH(M)-PITCHP(M))
4300 CONTINUE
      CA(M) = CF(M,1)
      CY(M) = CF(M,2)
      CN(M) = CF(M,3)
      CPITCH(M) = CF(M,4)
      CYAW(M) = CF(M,5)
      CROLL(M) = CF(M,6)

C
      CALL AXES (LM, M, A, ANU, PSI, OMEGA)
      DO 4500 N=1,3
      WGTTRA(N) = A(N,1)*WGTTA(1) + A(N,2)*WGTTA(2) + A(N,3)*WGTTA(3)
4500 THE FORMULAS FOR THE DERIVATIVES XDOT(KRK),...,RDOT(KRK)
      ARE GIVEN IN AFDC-TR-69-200, BY CHRISTOPHER AND CARLTON
      XDOT(KRK) = A(1,1) * U(M) + A(2,1) * V(M) + A(3,1) * W(M)
      YDOT(KRK) = A(1,2) * U(M) + A(2,2) * V(M) + A(3,2) * W(M)
      ZDOT(KRK) = A(1,3) * U(M) + A(2,3) * V(M) + A(3,3) * W(M)
      COSPSI = COS(PSI(M))
      SINPSI = SIN(PSI(M))
      COSOMG = COS(OMEGA(M))
      SINOMG = SIN(OMEGA(M))
      ANUDOT(KRK) = (Q(M) * COSOMG - R(M) * SINOMG) / COSPSI
      PSIDOT(KRK) = R(M) * COSOMG + Q(M) * SINOMG
      OMGDOT(KRK) = P(M) - ANUDOT(KRK) * SINPSI

C
      ITOTAL = SQRT( (U1+XDOT(KRK))**2 + YDOT(KRK)**2
      + ZDOT(KRK)**2 )
      QS = RHO * UTOTAL**2 / 2.
      FX = WGTTRA(1) - QS * S * CA(M) - FX1 + FX2
      FY = WGTTRA(2) - QS * S * CY(M)
      FZ = WGTTRA(3) - QS * S * CN(M) + FZ1 + FZ2
      TL = CROLL(M) + CLP * RSPAN * P(M) / (2.*UTOTAL)
      TM = CPITCH(M) + CHO * CRAR * Q(M) / (2.*UTOTAL)
      TN = CYAW(M) + CNR * CRAR * Q(M) / (2.*UTOTAL)
      TH = QS * S * RSPAN * TL
      TI = QS * S * CRAR * TM - FZ1 * ALX1 - FZ2 * ALX2
      TJ = QS * S * CRAR * TN
      UDOT(KRK) = FX / ANRAR + R(M) * V(M) - Q(M) * W(M)
      VDOT(KRK) = FY / ANRAR + P(M) * W(M) - R(M) * U(M)
      WDOT(KRK) = FZ / ANRAR + Q(M) * U(M) - P(M) * V(M)
      PNDOT = TL - Q(M)*R(M)*(AIZ-AIY) + P(M)*Q(M)*AIXZ
      PNODT / AIX
      RNDOT = TM - Q(M)*R(M)*AIXZ - P(M)*Q(M)*(AIY-AIX)
      RNDOT / AIZ
      PMDOT(KRK) = (PNDOT + RNDOT * AIXZ / AIX)
      / (1. - A(XZ**2 / (AIX**AIZ))
      QDOT(KRK) = TM - (P(M)**2-R(M)**2) * AIXZ
      - P(M) * R(M) * (AIX-AIZ)
      ODOT(KRK) = ODOT(KRK) / AIY
      RDOT(KRK) = TM + (PNDOT(KRK)-Q(M)*R(M)) * AIXZ
      - P(M) * Q(M) * (AIY-AIX)
      RDOT(KRK) = RDOT(KRK) / AIZ
      GO TO( 3000, 3000, 3000, 5100), KRK
5100 CONTINUE
      U(M) = U(2*K-1) + (UDOT(1)+2.*UDOT(2)+2.*UDOT(3)+UDOT(4))
      * DT(M-1) / 3.
      V(M) = V(2*K-1) + (VDOT(1)+2.*VDOT(2)+2.*VDOT(3)+VDOT(4))
      * DT(M-1) / 3.
      W(M) = W(2*K-1) + (WDOT(1)+2.*WDOT(2)+2.*WDOT(3)+WDOT(4))
      * DT(M-1) / 3.
      P(M) = P(2*K-1) + (PNDOT(1)+2.*PNDOT(2)+2.*PNDOT(3)+PNDOT(4))
      * DT(M-1) / 3.
      Q(M) = Q(2*K-1) + (QDOT(1)+2.*QDOT(2)+2.*QDOT(3)+QDOT(4))
      * DT(M-1) / 3.
      R(M) = R(2*K-1) + (RDOT(1)+2.*RDOT(2)+2.*RDOT(3)+RDOT(4))
      * DT(M-1) / 3.
      X(M) = X(2*K-1) + (XDOT(1)+2.*XDOT(2)+2.*XDOT(3)+XDOT(4))
      * DT(M-1) / 3.
      Y(M) = Y(2*K-1) + (YDOT(1)+2.*YDOT(2)+2.*YDOT(3)+YDOT(4))
      * DT(M-1) / 3.
      Z(M) = Z(2*K-1) + (ZDOT(1)+2.*ZDOT(2)+2.*ZDOT(3)+ZDOT(4))
      * DT(M-1) / 3.
      ANU(M) = ANU(2*K-1) + (ANUDOT(1)+2.*ANUDOT(2)
      +2.*ANUDOT(3)+ANUDOT(4)) * DT(M-1)/3.
      PSI(M) = PSI(2*K-1) + (PSIDOT(1)+2.*PSIDOT(2)
      +2.*PSIDOT(3)+PSIDOT(4)) * DT(M-1)/3.
      OMEGA(M) = OMEGA(2*K-1) + (OMGDOT(1)+2.*OMGDOT(2)
      +2.*OMGDOT(3)+OMGDOT(4)) * DT(M-1)/3.

C
      IF( ITCF-1 ) 5510, 5510, 6100
      IF( M-M2 ) 6100, 5520, 5520
      DO 5550 M=M1,M2
      PITCHP(M) = PITCH(M)
      YAWP(M) = YAW(M)
      AINC(P(M)) = AINCIO(M)
5550 CONTINUE
6100 CONTINUE
      IF( KM)-(INX-INCREM) ) 8622, 8624, 6200
      M1 = M1
      GO TO 8630
8624 M1 = 1
8630 WRITE(6,8632)
8632 FORMAT(// ' AERO DYNAMIC COEFFICIENTS FROM TAYLOR SERIES' / 3X,
      1, ITCF M CAXIAL CSIOE CNORM CPITCH CYAW CROLL
      2LL ')
      DO 8640 M=M1,M2
8640 WRITE(6,8642) ITCF, M, CA(M), CY(M), CN(M),
      CPITCH(M), CYAW(M), CROLL(M)
8642 FORMAT( 2I5, 6F10.4 )
      WRITE(6,8646)
8646 FORMAT(// ' U,V,W,P,Q,R,ANU,PSI,OMEGA ARE REFERRED TO BODY AXES.
      1 UNITS ARE FEET, SECONDS, DEGREES. ')
      WRITE(6,8652)
8652 FORMAT( 3X, 1 ITCF M ',
      1 3X, U(M) V(M) W(M) P(M) Q(M) R(M) ',
      2 4X, X(M) Y(M) Z(M) ANU(M) PSI(M) OMEGA(M) ',
      3 )
      DO 8660 M=M1,M2
      POEG = P(M) * 180. / 3.1416
      QOEG = Q(M) * 180. / 3.1416
      ROEG = R(M) * 180. / 3.1416
      ANUOEG = ANU(M) * 180. / 3.1416
      PSIOEG = PSI(M) * 180. / 3.1416
      OMGOEG = OMEGA(M) * 180. / 3.1416
8660 WRITE(6,8662) ITCF, M, U(M), V(M), W(M), POEG, QOEG, ROEG,
      1 X(M), Y(M), Z(M), ANUOEG, PSIOEG, OMGOEG

```

```

8662 FDMAT(215, 12F10.3 )
C
6200 CONTINUE
6300 CONTINUE
6500 CONTINUE
C
      WRITE(6,8720)
8720 FORMAT( // , 'TRAJECTORY VELOCITY AND DISPLACEMENT IN WIND AXES '
1         / , 'UNITS ARE FEET, SECONDS, DEG '
2         / , 'M, 3X, U(FT/SEC) V(FT/SEC) W(FT/SEC) P(DEG/S)
3         / , 'Q(DEG/S) R(DEG/S) X(FT) Y(FT) Z(FT) ANU(DEG) PSI(
4         / , 'DEG) OMEGA(DEG)' )
      DD 6600 M=1,NM
      CALL AXES ( LM, M, A, ANU, PSI, OMEGA )
      UWIND = A(1,1) * U(M) + A(2,1) * V(M) + A(3,1) * W(M)
      VWIND = A(1,2) * U(M) + A(2,2) * V(M) + A(3,2) * W(M)
      WWIND = A(1,3) * U(M) + A(2,3) * V(M) + A(3,3) * W(M)
C
      PWIND = A(1,1) * P(M) + A(2,1) * Q(M) + A(3,1) * R(M)
      QWIND = A(1,2) * P(M) + A(2,2) * Q(M) + A(3,2) * R(M)
      RWIND = A(1,3) * P(M) + A(2,3) * Q(M) + A(3,3) * R(M)
C
      OMEGAW = A(1,1) * OMEGA(M) + A(2,1) * ANU(M) + A(3,1) * PSI(M)
      ANUW = A(1,2) * OMEGA(M) + A(2,2) * ANU(M) + A(3,2) * PSI(M)
      PSIW = A(1,3) * OMEGA(M) + A(2,3) * ANU(M) + A(3,3) * PSI(M)
C
      PWIND = PWIND * 180. / 3.1416
      QWIND = QWIND * 180. / 3.1416
      RWIND = RWIND * 180. / 3.1416
      ANUW = ANUW * 180. / 3.1416
      PSIW = PSIW * 180. / 3.1416
      OMEGAW = OMEGAW * 180. / 3.1416
      WRITE(6,8730) M, UWIND, VWIND, WWIND, PWIND, QWIND, RWIND,
1         X(M), Y(M), Z(M), ANUW, PSIW, OMEGAW
8730 FORMAT( 15, 12F10.3 )
6600 CONTINUE
      STOP
      END

```

```

PROGRAM C - 3RD JDR STFP
THIS IS AN ALTERNATE VERSION OF THE MAIN PROGRAM OF THIS STFP IN
WHICH TRAJECTORIES ARE NOT COMPUTED.
THIS PROGRAM COMPUTES AERODYNAMIC FORCES ON A STATIONARY PLANFORM
WHOSE ORIENTATION IN SPACE IS SPECIFIED
DIMENSION
1 U (5), V (5), W (5), P (5), O (5), R (5),
2 X (5), Y (5), Z (5), ANU (5), PSI (5), OMEGA (5),
3 COTOT (5), CYTOT (5), CLTOT (5), CMTOT (5), CNTOT (5), CRTOT (5)
DIMENSION A13(3)
7100 FORMAT(16I5)
7200 FORMAT(16I0.0)
LM = 5
KM = 1
NM = 1
C INPUT DUMMY VALUES FOR U1 AND SCALE
U1 = 1.
SCALE = 1.
READ(5,7100) NCASE, NREAD
DO 100 MM=1,NCASE
U(MM) = 0.
V(MM) = 0.
W(MM) = 0.
P(MM) = 0.
O(MM) = 0.
R(MM) = 0.
100 CONTINUE
1 KJ = 0
DO 200 I=1,NREAD
READ(5,7200) (X(MM), MM=1,NCASE)
READ(5,7200) (V(MM), MM=1,NCASE)
READ(5,7200) (Z(MM), MM=1,NCASE)
READ(5,7200) (ANU(MM), MM=1,NCASE)
READ(5,7200) (PSI(MM), MM=1,NCASE)
READ(5,7200) (OMEGA(MM), MM=1,NCASE)
CALL VORLAT(LM, KM, NM, I)
1 KJ = 1
U1 = 1.
SCALE = 1.
C U1, V1, W1, P1, O1, R1
C COTOT, CYTOT, CLTOT, CMTOT, CNTOT, CRTOT
ICL = 1
RSPAN = 1.
CRAR = 4.395
DO 800 MM=1,NCASE
XORIG = X(MM) * 12.
YORIG = Y(MM) * 12.
ZORIG = Z(MM) * 12.
PITCH = ANU(MM) * 180. / 3.1416
YAW = PSI(MM) * 180. / 3.1416
ROLL = OMEGA(MM) * 180. / 3.1416
CALL AXES(LM, MM, NM, ANU)
CO = A1(1,1) * COTOT(MM) - A1(2,1) * CYTOT(MM)
CY = - A1(1,2) * COTOT(MM) + A1(2,2) * CYTOT(MM)
CL = A1(1,3) * COTOT(MM) - A1(2,3) * CYTOT(MM)
CR = A1(1,1) * RSPAN * CMTOT(MM) + A1(2,1) * CRAR * CMTOT(MM)
CM = A1(1,2) * RSPAN * CMTOT(MM) + A1(2,2) * CRAR * CMTOT(MM)
CN = A1(1,3) * RSPAN * CMTOT(MM) + A1(2,3) * CRAR * CMTOT(MM)
CR = CR / RSPAN
CM = CM / RSPAN
CN = CN / RSPAN
C WRITE(6,8720) MM, XORIG, YORIG, ZORIG, PITCH, YAW, ROLL
A720 FORMAT(// 'IRUN=', I2, 'X, ' XORIG(1)=', F6.2, 'X, '
1 ' YORIG(1)=', F6.2, 'X, ' ZORIG(1)=', F6.2, 'X, '
2 ' PITCH(1)=', F6.1, 'X, ' YAW(1)=', F6.1, 'X, ' ROLL(1)=',
3 F6.1, 'X)
DO 100 I=1,NCASE
A720 WRITE(6,8720) I, ICF
RR22 FORMAT(15X) ' FORCE COEFFICIENTS FROM LAMINAR SKIN FRICTION
1 ASSUMPTION
GO TO 8400
A730 WRITE(6,8720) I, ICF
A732 FORMAT(15X) ' FORCE COEFFICIENTS FROM TURBULENT SKIN FRICTION
1 ASSUMPTION
GO TO 8400
A740 WRITE(6,8720) I, ICF
A742 FORMAT(15X) ' FORCE COEFFICIENTS IN WIND AXES SYSTEM
1 / 15X, ' COIRAG) ' CYSINE FORCE) CL(LIFT) CR
2 ROLL) CM(PITCH) CN(YAW) / ' IRUN=', I2, ' F615.5
1 WRITE(6,8752) MM, COTOT(MM), CYTOT(MM), CLTOT(MM),
1 CRTOT(MM), CMTOT(MM), CNTOT(MM), CRTOT(MM)
A752 FORMAT(15X) ' FORCE COEFFICIENTS IN BODY AXES SYSTEM
1 / 15X, ' CAXIAL CSIDE CNORM CRD
2 ROLL CPITCH CYAW CSIDE / ' IRUN=', I2, ' F615.5
1
A900 CONTINUE
700 CONTINUE
STOP
END

```

```

SUBROUTINE VORLAT( LM, K, M,
1 NCASE
1 IKJ
1 NWP
1 ICF
1 X
1 CDTOT
COMMON/CNET/
1 NUC
2 COMMON/CCORR/
1 XNET
1 AKS1
1 RDC
COMMON/CVELOC/
1 COMMON/CPRINT/
C 1
1 DIMENSION OF GAMMAV IN ACDEFF MUST AGREE WITH GAMMAV BELOW
1 MOIAG(156,2), MINV (156), H (156), G (156,3)
1 DIMENSION
1 RQNTV(156,2,5),
2 GAMMAV(156,2,5),
3 HGAMMA(156,2,5),
4 VX(195,2,5), VY (195,2,5), VZ (195,2,5)
1 DIMENSION
1 MTIP(3), MRQNT(3)
2 IVEL(4), YP(8), ZP(10)
3 DOWNV(10,8,10), SIOEV(10,8,10), VMAG(10,8,10)
4 A13
5 UCF(LM), VCG(LM), WCG(LM), P(LM), D(LM), R(LM)
6 X(LM), Y(LM), Z(LM), ANU(LM), PSI(LM), OMEGA(LM)
7 CDTOT(5), CYTOT(5), CLTOT(5), CMTOT(5), CNTOT(5), CRTOT(5)
EQUIVALENCE ( RQNTV(1,1,1), HGAMMA(1,1,1) )
7100 FORMAT(15)
7200 FORMAT(15)
1 FT M-1 4000, 4000, 4150
4000 READ(51)
1 AMACH
2 IMGAMA
3 NWP
4 NCHORD
5 NSCORR
6 MTIP
7 XCG
8 XM
9 NSYM
A IVEL
D IPRG
E IX1
E IX2
E DOWNV
E READ(61)
1 NUC
2 XNET
3 AKS1
4 RDC
E REWIND 51
E RET
NVRT = SORT(1, -AMACH**2)
4150 CONTINUE
READ(21) MOIAG
C
IKJ = IKJ + 1
NWP = NWP
DO 6900 ISOLVE=1,4
GD TO( 5100, 5110, 5110, 5114), ISOLVE
5110 IF( IVEL(ISOLVE) ) 6900, 6900, 5180
5114 IF( IVEL(4) ) 6900, 6900, 5120
5120 CONTINUE
C THE FOLLOWING STATEMENTS, THROUGH 5172, ARE USED TO COMPUTE VELOC
C TY AT POINTS OFF THE SURFACE OF THE AERODYNAMIC PLANFORM
READ(5,7100) NWP
DO 5150 IW=1,NWP
READ(5,7100) NUC(IW,4), NUS(IW,4)
NVELC = NUC(IW,4)
NVELS = NUS(IW,4)
DO 7710 II=1,NVELC
READ(5,7200) (XFLOWF(II,JJ), JJ=1,NVELS)
DO 7720 II=1,NVELC
READ(5,7200) (YFLOWF(II,JJ), JJ=1,NVELS)
DO 7730 II=1,NVELC
READ(5,7200) (ZFLOWF(II,JJ), JJ=1,NVELS)
5150 CONTINUE
C
5160 NVEL(I,4) = NUC(I,4) * NUS(I,4)
IF( NWP=1 ) 5172, 5172, 5164
5164 DO 5170 IW=2,NWP
5170 NVEL(IW,4) = NVEL(IW-1,4) + NUC(IW,4) * NUS(IW,4)
5172 NUT(4) = NVEL(NWP,4)
C
5180 NVELT = NUT(ISOLVE)
DO 5700 ICASE=1,NCASE
MM = 2**(-2/ICASE)
ANU(MM) = ANU(MM) * BETA
PSI(MM) = PSI(MM) * BETA
CALL AXES ( LM, MM, A, ANU, PSI, OMEGA )
ANU(MM) = ANU(MM) / BETA
PSI(MM) = PSI(MM) / BETA
IF( IFC ) 5300, 5300, 5400
5300 DO 5310 ISYM=1,NSYM
DO 5310 IT=1,NVELT
VXNORM(IT,ISYM) = - A(1,1)
VYNORM(IT,ISYM) = - A(2,1)
VZNORM(IT,ISYM) = - A(3,1)
GO TO 5500
5400 CONTINUE
XORIG = X(MM) * BETA * 12. / SCALE
YORIG = Y(MM) * BETA * 12. / SCALE
ZORIG = Z(MM) * BETA * 12. / SCALE
CALL FRESTRI XM, ZM, YL, ZL, XN, YN,
LM
1 IX1, IX2, IY1, IY2, IZ1, IZ2,
2 XP, YP, ZP
3 DOWNV, SIOEV, VMAG
4 ISOLVE, ICASE, NSYM

```

```

5 XORTG , YORIG , ZORIG ,
6 A ,
7 MM , U1 , SCALE ,
8 UCG , VCG , WCG , P , Q , R )
5500 DO 5510 ISYM=1, NSYM
      DO 5510 IT=1, NVELT
      VX(IT, ISYM, ICASE) = VYNORM(IT, ISYM)
      VY(IT, ISYM, ICASE) = VYNORM(IT, ISYM)
      VZ(IT, ISYM, ICASE) = VZNORM(IT, ISYM)
5510 CONTINUE
C
IF( ISOLVE=1 ) 6100, 6100, 6400
6100 DO 6150 ICASE=1, NCASE
      DO 6150 ISYM=1, NSYM
      DO 6150 IT=1, NVORT
      ROOTV(IT, ISYM, ICASE) = ( - RDC(IT, 1) * VX(IT, ISYM, ICASE)
                                - RDC(IT, 2) * VY(IT, ISYM, ICASE)
                                - RDC(IT, 3) * VZ(IT, ISYM, ICASE) )
                                / HDIAG(IT, ISYM)
      GO TO( 6150, 6140 ), NSYM
6140 RDC(IT, 2) = - RDC(IT, 2)
6150 CONTINUE
C
DO 6210 ICASE=1, NCASE
      DO 6210 ISYM=1, NSYM
      DO 6210 IT=1, NVORT
6210 GAMMAV(IT, ISYM, ICASE) = 0.
C
DO 6250 ISYM=1, NSYM
      DO 6250 IT=1, NVORT
      READ(21) H
      DO 6240 ICASE=1, NCASE
      DO 6240 JT=1, NVORT
      GAMMAV(IT, 1, ICASE) = GAMMAV(IT, 1, ICASE)
                                + H*INV(JT) * ROOTV(JT, ISYM, ICASE)
      GO TO( 6230, 6232 ), NSYM
6230 GAMMAV(IT, 2, ICASE) = - GAMMAV(IT, 1, ICASE)
      GO TO 6240
6232 GAMMAV(IT, 2, ICASE) = GAMMAV(IT, 2, ICASE)
                                + H*INV(JT) * ROOTV(JT, 3-ISM, ICASE)
6240 CONTINUE
6250 CONTINUE
C
IF( IPRGAM ) 8838, 8838, 8828
8828 WRITE(6, 8830) (ICASE, ICASE=1, NCASE)
8830 FORMAT('1', 21X, ' GAMMAV(IT, ISYM, ICASE) / 16X, 5(6X, ' ICASE=', I1) )
      DO 8834 ISYM=1, NSYM
      DO 8834 IT=1, NVORT
8834 WRITE(6, 8836) ISYM, IT, (GAMMAV(IT, ISYM, ICASE), ICASE=1, NCASE)
8836 FORMAT('1', ISYM=', I1, ' IT=', I3, ' RFI4.5' )
8838 CONTINUE
C
IF( IHGAMA ) 6390, 6390, 6300
6300 CONTINUE
      WRITE(6, 8810) (ICASE, ICASE=1, NCASE)
8810 FORMAT('1', 21X, ' ROOTV(IT, ISYM, ICASE) / 16X, 5(6X, ' ICASE=', I1) )
      DO 8814 ISYM=1, NSYM
      DO 8814 IT=1, NVORT
8814 WRITE(6, 8816) ISYM, IT, (ROOTV(IT, ISYM, ICASE), ICASE=1, NCASE)
8816 FORMAT('1', ISYM=', I1, ' IT=', I3, ' RFI4.5' )
C
DO 6310 ICASE=1, NCASE
      DO 6310 ISYM=1, NSYM
      DO 6310 IT=1, NVORT
6310 HGAMMA(IT, ISYM, ICASE) = 0.
C
DO 6350 ISYM=1, NSYM
      DO 6350 IT=1, NVORT
      READ(21) H
      DO 6340 ICASE=1, NCASE
      DO 6340 JT=1, NVORT
      HGAMMA(IT, 1, ICASE) = HGAMMA(IT, 1, ICASE)
                                + H*INV(JT) * GAMMAV(JT, ISYM, ICASE)
      GO TO( 6340, 6332 ), NSYM
6332 HGAMMA(IT, 2, ICASE) = HGAMMA(IT, 2, ICASE)
                                + H*INV(JT) * GAMMAV(JT, 3-ISM, ICASE)
6340 CONTINUE
6350 CONTINUE
C
WRITE(6, 8820) (ICASE, ICASE=1, NCASE)
8820 FORMAT('1', 21X, ' HGAMMA(IT, ISYM, ICASE) / 16X, 5(6X, ' ICASE=', I1) )
      DO 8824 ISYM=1, NSYM
      DO 8824 IT=1, NVORT
8824 WRITE(6, 8826) ISYM, IT, (HGAMMA(IT, ISYM, ICASE), ICASE=1, NCASE)
8826 FORMAT('1', ISYM=', I1, ' IT=', I3, ' RFI4.5' )
6390 IF( IVEL(1) ) 6900, 6900, 6400
C
6400 CONTINUE
      DO 6450 ISYM=1, NSYM
      DO 6450 IT=1, NVELT
      READ(21) G
      IF( IPRG ) 8930, 8930, 8912
      IF( IT=1 ) 8916, 8916, 8914
      IF( IT=NVELT ) 8930, 8916, 8916
      DO 8920 N=1, 3
8920 WRITE(6, 8922) ISOLVE, ISYM, IT, N, (G(JT, N), JT=1, NVORT)
8922 FORMAT('1', ISOLVE=', I1, ' ISYM=', I1, ' IT=', I3, ' 5X, '
            , N=', I1, ' ISX=', G(JT, N) READ FROM UNIT(21) / (1H 16F8.3))
8930 CONTINUE
      DO 6440 ICASE=1, NCASE
      DO 6440 JT=1, NVORT
      VX(IT, 1, ICASE) = VX(IT, 1, ICASE)
                                + G(JT, 1) * GAMMAV(JT, ISYM, ICASE)
      VY(IT, 1, ICASE) = VY(IT, 1, ICASE)
                                + G(JT, 2) * GAMMAV(JT, ISYM, ICASE)
      VZ(IT, 1, ICASE) = VZ(IT, 1, ICASE)
                                + G(JT, 3) * GAMMAV(JT, ISYM, ICASE)
      GO TO( 6440, 6432 ), NSYM
6432 VX(IT, 2, ICASE) = - VX(IT, 2, ICASE)
                                + G(JT, 1) * GAMMAV(JT, 3-ISM, ICASE)
      VY(IT, 2, ICASE) = VY(IT, 2, ICASE)
                                + G(JT, 2) * GAMMAV(JT, 3-ISM, ICASE)
      VZ(IT, 2, ICASE) = VZ(IT, 2, ICASE)
                                + G(JT, 3) * GAMMAV(JT, 3-ISM, ICASE)
6440 CONTINUE
6450 CONTINUE
C
DO 6900 ICASE=1, NCASE
      DO 6900 ISYM=1, NSYM

```

```

      DO 6580      IW=1,NWP
      NVFLC      = NHC(IW,ISOLVE)
      NVFLS      = NUS(IW,ISOLVE)
      DO 6580      II=1,NVFLC
      DO 6580      JJ=1,NVFLS
      IF (IW-1) 6562, 6562, 6564
      6562      IT      =      (II-1)*NVFLS + JJ
      GO TO 6572
      6564      IT      = NVFL(IW-1,ISOLVE) + (II-1)*NVFLS + JJ
      6572      U(II,JJ,IW) = VX(IT,ISYM,CASE)
      V(II,JJ,IW) = VY(IT,ISYM,CASE)
      W(II,JJ,IW) = VZ(IT,ISYM,CASE)
      6580      CONTINUE
C
      IF IPRVFL) 6694, 6694, 6692
      6692      CALL VELOCY( ISOLVE, ICASE, NSYM, ISYM )
      6694      MM      = 200-2*ICASE
      ALPHA      = ANU(MM) * 180. / 3.1416
      PSIDFG      = PSI(MM) * 180. / 3.1416
      PMI      = OMEGA(MM) * 180. / 3.1416
      CALL ACNEFF( ISOLVE, NCASE, ICASE, NSYM, ISYM,
      1      LM, ICF,
      1      MKJ, MRONT,
      1      MM,
      2      RETA, ANU, PSI,
      2      XCG, YCG, ZCG,
      3      S, RSPAN, CRAR,
      4      GAMMAV,
      5      CNTOT, CYTOT, CLTOT, CMTOT, CNTOT, CRTOT )
C
      GO TO( 6800, 6710), NSYM
      DO 6750      IW=1,NWP
      NNC      = NCHORD(IW)
      NNS      = NSPAN(IW)
      NVORC      = NCHORD(IW)-1
      NVORS      = NSPAN(IW)-1
      DO 6720      I=1,NNC
      DO 6720      J=1,NNS
      6720      YNET(I,J,IW) = - YNET(I,J,IW)
      GO TO( 6730, 6750, 6750, 6750), ISOLVE
      6730      DO 6740      IP=1,NVORC
      DO 6740      IO=1,NVORS
      ETA(IP,IO,IW) = - ETA(IP,IO,IW)
      IF (IW-1) 6732, 6732, 6734
      6732      IT      =      (IP-1)*NVORS + IO
      GO TO 6736
      6734      IT      = NVOR(IW-1) + (IP-1)*NVORS + IO
      6736      RDC(IT,2) = - RDC(IT,2)
      6740      CONTINUE
      6750      CONTINUE
      6800      CONTINUE
C
      6900      CONTINUE
      REMIND 21
      RETURN
      END

```

```

SHARDTIME FRESTR XM, 7L, YL, 7L, XM, YN.
1 1X1, 1X2, 1Y1, 1Y2, 1Z1, 1Z2, 7P
2 X2 Y2
3 00WNV : S1NFV : VMAG :
4 1SOLVF : ICASE : 1SYM :
5 X1X1G : Y1X1G : 7NR1G :
6 A :
7 WM :
8 UCG : VCG : III : SCALE : P : Q : R :
9 COMMON/CNFT / NWP : N1WP : N1P :
10 NCN13N(3) : NSPAN(3) : NSCNRN(3) : NVNM(3) :
11 MIC(3,4) : NUS(3,4) : NVFL(3,4) : NIT(4) :
12 COMMON/CCNCRN/ XNFT(124,5,3) : YNFT(124,5,3) : 7NFT(124,5,3) :
13 AKSI(123,4,3) : FTA(123,4,3) : 7FTA(123,4,3) :
14 XFLNM(1,1) : YFLNM(1,1) : ZFLNM(1,1) :
15 9NC(15A,3) : AX(3) : AY(3) : AZ(3) :
16 COMMON/CVFLC/ V(124,5,3) : V(124,5,3) : W(124,5,3) :
17 VXNCRN(195,2) : VYNCRN(195,2) : V7NCRN(195,2) :
18 COMMON/CPRINT/ PRG : PRVFL : PR1AM : PRCP : PRFC :
19 DIMENSION YPIR : 7P(10) :
20 00WNV(10,R,10) : S1DEV(10,R,10) : VMAG(10,R,10) :
21 A(3,3) :
22 1CGL1M : VCG1M : WCG1M : P(1M) : O(1M) : Q(1M) :
23 F00WNV(2) : FSIDEV(2) : FVMAG(2) :
24 VPNC(124,5,3) :
NO 6000 1SYM=1,1SYM
NN 5950 1W=1,N1WP
NVLC = N1C(1W,1SOLVF)
NVLS = N1S(1W,1SOLVF)
NN 5950 1I=1,NVLC
NN 5950 1J=1,NVLS
IF(1W-1) 5502, 5507, 5504
5502 IT = (1I-1)*NVLS + JJ
GN 10 5506 = NVLC(1W-1,1SOLVF) + (1I-1)*NVLS + JJ
5504 IT 5520, 5530, 5540, 1SOLVF
5510 IP = 1I
IO = JJ
PX = XNET(1I,J,W) + XNET(1I,J+1,W) / 2.
PY = YNET(1I,J,W) + YNET(1I,J+1,W) / 2.
PZ = ZNET(1I,J,W) + ZNET(1I,J+1,W) / 2.
GN 10 5542
C
5570 J = 1I
J = JJ
PX = XNET(1I,J,W) + XNET(1I,J+1,W) / 2.
PY = YNET(1I,J,W) + YNET(1I,J+1,W) / 2.
PZ = ZNET(1I,J,W) + ZNET(1I,J+1,W) / 2.
GN 10 5542
C
5530 J = 1I
J = JJ
PX = XNET(1I,J,W) + XNET(1I+1,J,W) / 2.
PY = YNET(1I,J,W) + YNET(1I+1,J,W) / 2.
PZ = ZNET(1I,J,W) + ZNET(1I+1,J,W) / 2.
GN 10 5542
C
5540 PX = XLOWF(1I,JJ)
PY = YLOWF(1I,JJ)
PZ = ZLOWF(1I,JJ)
C
5542 PX = XM
PY = YL
PZ = ZL
C
GN 10( 554A, 5546), 1SYM
5544 PY = -PY
5546 CONTINUE
= A(1,1) * PX + A(2,1) * PY + A(3,1) * PZ
= A(1,2) * PX + A(2,2) * PY + A(3,2) * PZ
= A(1,3) * PX + A(2,3) * PY + A(3,3) * PZ
XVP = XVP + XN1G
YVP = YVP + YN1G
ZVP = ZVP + ZN1G
IF( XVP - XP(1X) ) 5930, 5A10, 5620
XMT = ( XVP - XP(1X) ) / ( XP(1X+1) - XP(1X) )
IX = IX+1
GN 10 5700
5620 DP 5632 IX=X1,1X2
IF( XVP - XP(1X) ) 5430, 5A30, 5632
5630 XMT = ( XVP - XP(1X) ) / ( XP(1X) - XP(1X-1) )
GN 10 5700
5632 CONTINUE
GN 10 5430
C
5700 IF( YVP - YP(1Y) ) 5930, 5710, 5720
5710 YMT = ( YVP - YP(1Y) ) / ( YP(1Y+1) - YP(1Y) )
1Y = 1Y+1
GN 10 5400
5720 NN 5732 1Y=1Y,1Y2
IF( YVP - YP(1Y) ) 5730, 5730, 5732
5730 YMT = ( YVP - YP(1Y) ) / ( YP(1Y) - YP(1Y-1) )
GN 10 5400
5732 CONTINUE
GN 10 5430
C
5A00 IF( ZVP - ZP(1Z) ) 5930, 5A10, 5A20
5A10 ZMT = ( ZVP - ZP(1Z) ) / ( ZP(1Z+1) - ZP(1Z) )
1Z = 1Z+1
GN 10 5400
5A20 NN 5A32 1Z=1Z,1Z2
IF( ZVP - ZP(1Z) ) 5A30, 5A30, 5A32
5A30 ZMT = ( ZVP - ZP(1Z) ) / ( ZP(1Z) - ZP(1Z-1) )
GN 10 5900
5A32 CONTINUE
GN 10 5430
C
5400 REF. NRS AMS 55, P.447, 25.2.64
FINWNV(X1)= (1.-YMT) * (1.-ZMT) * 00WNV(1X-2+NX,1Y-1,1Z-1)
+ (1.-YMT) * (1.-ZMT) * 00WNV(1X-2+NX,1Y-1,1Z-1)
+ (1.-YMT) * ZMT * 00WNV(1X-2+NX,1Y-1,1Z-1)
+ (1.-YMT) * (1.-ZMT) * S1DEV(1X-2+NX,1Y-1,1Z-1)

```

```

1      + YWT * (1.-ZWT) * S1DEV(IX-2+NX,1Y,1Z-1)
2      + (1.-YWT) * ZWT * S1DEV(IX-2+NX,1Y-1,1Z)
3      + YWT * ZWT * S1DEV(IX-2+NX,1Y,1Z-1)
1      FVMAG(NX) = (1.-YWT) * (1.-ZWT) * VMAG(IX-2+NX,1Y-1,1Z-1)
2      + (1.-YWT) * ZWT * VMAG(IX-2+NX,1Y,1Z-1)
3      + YWT * ZWT * VMAG(IX-2+NX,1Y,1Z)
5910  CONTINUE
      DOWNVP = (1.-XWT) * FDOWNV(1) + XWT * FDOWNV(2)
      S1DEVP = (1.-XWT) * FS1DEV(1) + XWT * FS1DEV(2)
      VPMAG = (1.-XWT) * FVMAG(1) + XWT * FVMAG(2)
C     POSITIVE DOWNWASH ANGLE (EPS) LON = UPWASH
C     POSITIVE SIDEWASH ANGLE (SIGMA) = INWASH ON THE PORTSIDE
      TANEPS = TAN( DOWNVP * 3.1416/180. )
      TANSIG = TAN( S1DEVP * 3.1416/180. )
      UP = -1. / SORT( 1. + TANEPS**2 + TANSIG**2 )
      VP = - TANSIG * UP
      WP = - TANEPS * UP
      UP = UP * VPMAG
      VP = VP * VPMAG
      WP = WP * VPMAG
      GN TO 5940
5930  UP = -1.
      VP = 0.
      WP = 0.
5940  CONTINUE
      VXNORM(11,1SYM) = A(1,1) * UP + A(1,2) * VP + A(1,3) * WP
      VYNORM(11,1SYM) = A(2,1) * UP + A(2,2) * VP + A(2,3) * WP
      VZNORM(11,1SYM) = A(3,1) * UP + A(3,2) * VP + A(3,3) * WP
      UFS = UCG(MM) + Q(MM) * PZ * SCALE / 12.
      VFS = VCG(MM) + R(MM) * PY * SCALE / 12.
      WFS = WCG(MM) + Q(MM) * PX * SCALE / 12.
      UFS = UCG(MM) - Q(MM) * PZ * SCALE / 12.
      VFS = VCG(MM) - R(MM) * PY * SCALE / 12.
      WFS = WCG(MM) - Q(MM) * PX * SCALE / 12.
      VXNORM(11,1SYM) = VXNORM(11,1SYM) - UFS / 11
      VYNORM(11,1SYM) = VYNORM(11,1SYM) - VFS / 11
      VZNORM(11,1SYM) = VZNORM(11,1SYM) - WFS / 11
      VPNORM(11,JJ,1W) = SORT( VXNORM(11,1SYM)**2 + VYNORM(11,1SYM)**2 + VZNORM(11,1SYM)**2 )
5950  CONTINUE
C
      IF( 1PRVFS ) R750, R750, R700
      R700 WRITE(6,R704) ' ' , ISOLVE, ICASE, NSYM, 1SYM
      R704 FORMAT(1//,ROX ' ' , ISOLVE='11', ICASE='12', NSYM='11', 1SYM='11' )
      WRITE(6,R705) ' '
      R705 FORMAT(1//, ' ' , MAGNITUDE OF FREFSTRAM VELOCITY VECTOR.
      RESULANT OF VXNORM, VYNORM, VZNORM.
      ' ' , VPNORM(11,JJ,1W) )
      DO 8740 1W=1,N1WP
      WRITE(6,R708) ' ' , 1W
      R708 FORMAT(1//, ' ' , 1W = WING PART = '11' )
      NVELC = NUC(1W,ISOLVE)
      NVELS = NUS(1W,ISOLVE)
      DO 8730 11=1,NVELC
      R730 WRITE(6,R732) ' ' , 11, (VPNORM(11,JJ,1W), JJ=1,NVELS)
      R732 FORMAT(1//, ' ' , 11='12, 10F10.5 )
      R740 CONTINUE
      R750 CONTINUE
C
      6000 CONTINUE
C
      IF( 1PRVFS ) R850, R850, R800
      R800 GO TO R840 1SYM=1,NSYM
      WRITE(6,R804) ' ' , ISOLVE, ICASE
      R804 FORMAT(1//,ROX ' ' , ISOLVE='11', ICASE='12' )
      WRITE(6,R806) ' '
      R806 FORMAT(1//, ' ' , NORMALIZED NONUNIFORM FREESTREAM VELOCITY F
      IELD INDUCED BY PARENT AIRCRAFT AT POINTS ON STORE AT WHICH VELOC
      ITY IS COMPUTED. ' ' , VXNORM, VYNORM, VZNORM ARE GIVEN IN THE ROM
      ' ' )
      49 AXES REFERENCE SYSTEM.
      DO 8840 1W=1,N1WP
      NVELC = NUC(1W,ISOLVE)
      NVELS = NUS(1W,ISOLVE)
      WRITE(6,R808) ' ' , 1W
      R808 FORMAT(1//, ' ' , 1W = WING PART = '11' )
      DO 8840 N=1,3
      WRITE(6,R812) ' ' , N
      R812 FORMAT(1//, ' ' , N=1,NVELC
      IF( 1W-1 ) 8818, 8818, R820
      R818 1T1 = (11-1)*NVELS + 1
      1T2 = 11 *NVELS
      GO TO 8826
      R820 1T1 = NVEL(1W-1,ISOLVE) + (11-1)*NVELS + 1
      1T2 = NVEL(1W-1,ISOLVE) + 11 *NVELS
      R826 GO TO 8828, R832, 8836, N
      R828 WRITE(6,R830) ' ' , ISYM, 1W, 11, (VXNORM(11,1SYM, 1T=1T1,1T2)
      R830 FORMAT(1//, ' ' , VXNORM(11,1SYM, 1T=1T1,1T2)
      ' ' , 11='12, 5F11.3 )
      GO TO 8840
      R832 WRITE(6,R834) ' ' , ISYM, 1W, 11, (VYNORM(11,1SYM, 1T=1T1,1T2)
      R834 FORMAT(1//, ' ' , VYNORM(11,1SYM, 1T=1T1,1T2)
      ' ' , 11='12, 5F11.3 )
      GO TO 8840
      R836 WRITE(6,R838) ' ' , ISYM, 1W, 11, (VZNORM(11,1SYM, 1T=1T1,1T2)
      R838 FORMAT(1//, ' ' , VZNORM(11,1SYM, 1T=1T1,1T2)
      ' ' , 11='12, 5F11.3 )
      GO TO 8840
      R840 CONTINUE
      R850 CONTINUE
      RETURN
      END

```

```

SUBROUTINE VELOCITY SOLVE, ICASE, NSYM, ISYM, I
COMMON/CNET / NWP, NSPAN(3), NSCORD(3), NVOR(3),
1 NUC(1,4), NUS(1,4), NVEL(1,4), NUT(1,4),
2 COMMON/CVELAC/ U(2,5,3), V(2,5,3), W(2,5,3),
3 VXNDRM(195,2), VYNDRM(195,2), VZNDRM(195,2)
COMMON/CPRINT/ IPRG,
1 IPRVFS, IPRVEL, IPRGAM, IPRCP, IPRCF
2 DIMENSION UMIOS(24,5,3), VMIOS(24,5,3), WMIOS(24,5,3),
3 UMIOC(24,5,3), VMIOC(24,5,3), WMIOC(24,5,3),
4 U(1,1,1), VMIOC(1,1,1), WMIOS(1,1,1,1),
5 W(1,1,1), WMIOC(1,1,1), WMIOS(1,1,1,1)
6 GO TO( R100, 2500, 3500, 8100), ISOLVE
R100 IF( IPRVEL ) 1900, 1900, 8102
R102 CONTINUE
WRITE(6,8104) ISOLVE, ICASE, NSYM, ISYM
R104 FORMAT('1',ROX, ISOLVE='1', ICASE='12',
1 NSYM='11', ISYM='11')
WRITE(6,8106)
R106 FORMAT( ' NORMALIZED VELOCITY AT BOUNDARY POINTS =
1 IFREESTREAM + G*GAMMA )
DO 8150 IW=1,NWP
NVELC = NUC(IW,SOLVE)
NVELS = NUS(IW,SOLVE)
WRITE(6,8108) IW, WING PART = '11'
R108 FORMAT(// IW= WING PART = '11' )
R110 FORMAT(// UM(IW,IQ,IW) )
DO 8112 IP=1,NVELC
R112 WRITE(6,8114) IP, (U(IP,IQ,IW), IQ=1,NVELS)
R114 FORMAT(// IP='12', RF15.4 )
C
R120 WRITE(6,8120) V(IP,IQ,IW)
R120 FORMAT(// V(IP,IQ,IW) )
DO 8122 IP=1,NVELC
R122 WRITE(6,8124) IP, (V(IP,IQ,IW), IQ=1,NVELS)
R124 FORMAT(// IP='12', RF15.4 )
C
R130 WRITE(6,8130) W(IP,IQ,IW)
R130 FORMAT(// W(IP,IQ,IW) )
DO 8132 IP=1,NVELC
R132 WRITE(6,8134) IP, (W(IP,IQ,IW), IQ=1,NVELS)
R134 FORMAT(// IP='12', RF15.4 )
C
R150 CONTINUE
1900 RETURN
C
2500 CONTINUE
DO 2600 IW=1,NWP
NVELC = NUC(IW,2)
NVELS = NUS(IW,2)
DO 2600 IQ=1,NVELC
DO 2600 IO=1,NVELS
UMIOS(IQ,IW) = U(IQ,IO,IW)
VMIOS(IQ,IW) = V(IQ,IO,IW)
WMIOS(IQ,IW) = W(IQ,IO,IW)
2600 WMIOS(IQ,IW) = W(IQ,IO,IW)
C
IF( IPRVEL ) 2900, 2900, 8202
R202 CONTINUE
WRITE(6,8204) ISOLVE, ICASE, NSYM, ISYM
R204 FORMAT('1',ROX, ISOLVE='1', ICASE='12',
1 NSYM='11', ISYM='11')
WRITE(6,8206)
R206 FORMAT( ' NORMALIZED VELOCITY AT SPANWISE SEGMENT M10
1 IPOINTS = FREESTREAM + G*GAMMA )
DO 8250 IW=1,NWP
NVELC = NUC(IW,2)
NVELS = NUS(IW,2)
WRITE(6,8208) IW, WING PART = '11'
R208 FORMAT(// IW= WING PART = '11' )
R210 FORMAT(// UM(IQ,IW) )
DO 8212 IP=1,NVELC
R212 WRITE(6,8214) IP, (UM(IQ,IW), IQ=1,NVELS)
R214 FORMAT(// IP='12', RF15.4 )
R220 WRITE(6,8220) VMIOS(IQ,IW)
R220 FORMAT(// VMIOS(IQ,IW) )
DO 8222 IP=1,NVELC
R222 WRITE(6,8224) IP, (VMIOS(IQ,IW), IQ=1,NVELS)
R224 FORMAT(// IP='12', RF15.4 )
R230 WRITE(6,8230) WMIOS(IQ,IW)
R230 FORMAT(// WMIOS(IQ,IW) )
DO 8232 IP=1,NVELC
R232 WRITE(6,8234) IP, (WMIOS(IQ,IW), IQ=1,NVELS)
R234 FORMAT(// IP='12', RF15.4 )
R250 CONTINUE
2900 RETURN
C
3500 CONTINUE
DO 3600 IW=1,NWP
NVELC = NUC(IW,3)
NVELS = NUS(IW,3)
DO 3600 IP=1,NVELC
DO 3600 J=1,NVELS
UMIOC(IP,J,IW) = U(IP,J,IW)
VMIOC(IP,J,IW) = V(IP,J,IW)
WMIOC(IP,J,IW) = W(IP,J,IW)
3600 WMIOC(IP,J,IW) = W(IP,J,IW)
C
IF( IPRVEL ) 3900, 3900, 8302
R302 CONTINUE
WRITE(6,8304) ISOLVE, ICASE, NSYM, ISYM
R304 FORMAT('1',ROX, ISOLVE='1', ICASE='12',
1 NSYM='11', ISYM='11')
WRITE(6,8306)
R306 FORMAT( ' NORMALIZED VELOCITY AT CHORDWISE SEGMENT M1
1 IPOINTS = FREESTREAM + G*GAMMA )
DO 8350 IW=1,NWP
NVELC = NUC(IW,3)
NVELS = NUS(IW,3)
WRITE(6,8308) IW, WING PART = '11'
R308 FORMAT(// IW= WING PART = '11' )
R310 FORMAT(// UMIOC(IP,J,IW) )
DO 8312 IP=1,NVELC
R312 WRITE(6,8314) IP, (UMIOC(IP,J,IW), J=1,NVELS)
R314 FORMAT(// IP='12', RF15.4 )
R320 WRITE(6,8320) VMIOC(IP,J,IW)
R320 FORMAT(// VMIOC(IP,J,IW) )
DO 8322 IP=1,NVELC

```

```

R322 WRITE(A,R324)      IP, (VMINDC(IP,J,IW), J=1,NVFLS)
R324 FORMAT(            * IP=*,I2,  RF15.4      )
      WRITE(A,R33D)      *
R330 FORMAT(            * VMINDC(IP,J,IW)        * )
      DO R332 IP=1,NVFLC
R332 WRITE(A,R334)      IP, (VMINDC(IP,J,IW), J=1,NVFLS)
R334 FORMAT(            * IP=*,I2,  RF15.4      )
R350 CONTINUE
R900 RETURN
      END

```

```

SURROUTINE ACOEFFI ISOLVE, NCASE, ICASE, NSYM, ISYM,
1  IKJ, ICF,
1  MTIP, MROOT,
1  MM,
2  BFTA, ANU, PSI,
2  XCG, YCG, ZCG,
3  S, CBAR,
4  GAMMAV, CYTOT, CLTOT, CMTOT, CNTOT, CRTOT, I
C  COMMON/CNET/ NWP, NSPAN(3), NSCORD(3), NVOR(13),
1  NCHORD(3), NUS(3,4), NVEL(3,4), NUT(4),
2  NUC(3,4), XNET(24,5,3), YNET(24,5,3), ZNET(24,5,3),
COMMON/CCOORD/ AKSI(23,4,3), ETA(23,4,3), ZETA(23,4,3),
1  XFLOW(1,1), YFLOW(1,1), ZFLOW(1,1),
2  BOC(156,3), AX(3), AY(3), AZ(3),
3  COMMON/CVELOC/ U(24,5,3), V(24,5,3), W(24,5,3),
1  VXNORM(195,2), VYNORM(195,2), VZNORM(195,2)
COMMON/CPRINT/ IPRG, IPRVEL, IPRGAM, IPRCP, IPRCF
1  DIMENSION
1  MTIP(3), MROOT(3), ANU(LM), PSI(LM),
2  GAMMAV(156,2,5),
4  CYTOT(5), CLTOT(5), CMTOT(5), CNTOT(5), CRTOT(5)
1  DIMENSION
1  DELV1(3), DELV2(3),
2  DGLSP(24,4), DGLCM(23,5),
3  URPU(23,4), VBPUPP(23,4), WBPUPP(23,4),
4  URPLW(23,4), VBPLW(23,4), WBPLW(23,4),
5  CRPP(23,4), CPLW(23,4),
2  DIMENSION UMIDS(24,5,3), VMIDS(24,5,3), WMIDS(24,5,3),
1  UMIOC(24,5,3), VMIOC(24,5,3), WMIOC(24,5,3)
1  DIMENSION GAMMA(24,6,3)
1  DIMENSION
1  CMSP(24,4), CVSP(24,4), CLSP(24,4),
2  CMSP(24,4), CNSP(24,4), CRSP(24,4),
3  CDCH(23,5), CVCH(23,5), CLCH(23,5),
4  CMCH(23,5), CNCH(23,5), CRCH(23,5),
1  DIMENSION DELX(195,2), DELY(195,2), DELZ(195,2),
1  A3(195,2,6)
1  EQUIVALENCE ( U(1,1,1), UMIOC(1,1,1), UMIDS(1,1,1) ),
2  ( V(1,1,1), VMIOC(1,1,1), VMIDS(1,1,1) ),
3  ( W(1,1,1), WMIOC(1,1,1), WMIDS(1,1,1) )
1  DEFINE GAMMA(K, LG, IW)
1  LG = 1
1  MTIP(IW) = 0 CONTIGUOUS TIPS ( CLOSED BODY )
1  MROOT(IW) = 0 CONTIGUOUS ROOTS ( CLOSED BODY, WING, HORIZ TAIL )
1  = 1 TERMINAL TIPS ( WING, TAIL, FIN )
1  = 1 TERMINAL ROOTS ( VERT TAIL, FIN )
C
DO 1400 IW=1,NWP
NMC = NCHORD(IW)
NNS = NSPAN(IW)
NNSP1 = NNSP(IW) + 1
NVORC = NCHORD(IW) - 1
NVORS = NSPAN(IW) - 1
DO 1110 K=1,NVORC
JT = (K-1)*NVORS + 1
GAMMA(K, 1, IW) = ( 1-MTIP(IW) )
1  JT = K * NVORS * ( 1-MROOT(IW) )
1110 1  GAMMA(K, NVORS+2, IW) = ( 1-MROOT(IW) )
C  * ( -GAMMAV(JT,3-ISYM,ICASE) )
DO 1250 LG=2,NNS
DO 1250 K=1,NVORC
(F(IW-1)) 1210, 1210, 1220
1  JT = (K-1)*NVORS + (LG-1)
1220 JT TO 1250
1250 GAMMA(K, LG, IW) = NVOR(IW-1) + (K-1)*NVORS + (LG-1)
C  = GAMMAV(JT, ISYM, ICASE)
DO 1350 LG=1,NNSP1
1350 GAMMA(NNC, LG, IW) = 0.
1400 CONTINUE
C
IF( IPRGAM ) 8150, 8150, 8100
R100 CONTINUE
WRITE(6,8104) ISOLVE, ICASE, NSYM, ISYM
8104 FORMAT('11,BOX', ISOLVE='11', ICASE='12',
1  NSYM='11', ISYM='11')
1  WRITE(6,8110) GAMMA(K, LG, IW)
8110 FORMAT('11,BOX', GAMMA(K, LG, IW)
DO 8130 IW=1,NWP
NMC = NCHORD(IW)
NNSP1 = NSPAN(IW) + 1
WRITE(6,8120) IW, WING PART = '11'
8120 FORMAT('11,BOX', IW, WING PART = '11')
DO 8130 K=1,NMC
8130 WRITE(6,8132) K, (GAMMA(K, LG, IW), LG=1,NNSP1)
8132 FORMAT('11,BOX', K, (GAMMA(K, LG, IW), LG=1,NNSP1)
8150 CONTINUE
C
GO TO( 1500, 2000, 3000), ISOLVE
1500 CONTINUE
DO 1900 IW=1,NWP
NMC = NCHORD(IW)
NNS = NSPAN(IW)
NVORC = NCHORD(IW) - 1
NVORS = NSPAN(IW) - 1
C
COMPUTE OGOL AT MIDPOINTS OF SPAN SEGMENTS
DO 1550 I=1,NMC
IP = 1
DO 1550 IO=1,NVORS
J = IO
LG = IO+1
IF( I-1 ) 1510, 1510, 1508
1508 IF( I=NMC ) 1520, 1530, 1530
1510 XLE = ( XNET(1,J,IW) + XNET(1,J+1,IW) ) / 2.
1510 YLE = ( YNET(1,J,IW) + YNET(1,J+1,IW) ) / 2.
1510 ZLE = ( ZNET(1,J,IW) + ZNET(1,J+1,IW) ) / 2.
ELAS/OC = 2. * SORT( ( AKSI(1,IO,IW)-XLE
1  + ( ETA(1,IO,IW)-YLE

```

```

      2      GN TD 1540      + I ZETA( 1,IO,IW)-ZLE      )**2 1
C
1520 ELASOC = SORT( ( AKSI( IP,IQ,IW)-AKSI( IP-1,IQ,IW) ) **2
      1      + ( ETA( IP,IQ,IW)-ETA( IP-1,IQ,IW) ) **2
      2      + ( ZETA( IP,IQ,IW)-ZETA( IP-1,IQ,IW) ) **2 )
C
      2      GO TD 1540
C
1530 XTE = ( XNET(NMC,J,IW) + XNET(NMC,J+1,IW) ) / 2.
      1      YTE = ( YNET(NMC,J,IW) + YNET(NMC,J+1,IW) ) / 2.
      2      ZTE = ( ZNET(NMC,J,IW) + ZNET(NMC,J+1,IW) ) / 2.
      1      ELASOC = 2.*SORT( ( AKSI(NVORC,IQ,IW)-XTE ) **2
      2      + ( ETA(NVORC,IQ,IW)-YTE ) **2
      3      + ( ZETA(NVORC,IQ,IW)-ZTE ) **2 )
C
1540 DGOLSP(I,IO) = GAMMA(I,IO+1,IW) / ELASOC
1550 CONTINUE
C
      C      COMPUTE OGDL AT MIDPOINTS OF CHORD SEGMENTS
      DO 1650 IP=1,NVORC
      1      IO = IP
      DO 1650 J=1,NNS
      1      IQ = J
      1      LG = IO+1
      IF( J-1 ) 1610,1610,1604
      IF( J-NNS ) 1620,1630,1630
C
1610 IF( MTIP(IW) ) 1612,1612,1614
1612 ELASOC = ABS( 2.*ETA( IP,1,IW) )
      GO TO 1640
1614 XTIP = ( XNET(1,1,IW) + XNET(1+1,1,IW) ) / 2.
      1      YTIP = ( YNET(1,1,IW) + YNET(1+1,1,IW) ) / 2.
      2      ZTIP = ( ZNET(1,1,IW) + ZNET(1+1,1,IW) ) / 2.
      1      ELASOC = 2.*SORT( ( AKSI( IP,1,IW)-XTIP ) **2
      2      + ( ETA( IP,1,IW)-YTIP ) **2
      3      + ( ZETA( IP,1,IW)-ZTIP ) **2 )
C
      2      GO TD 1640
C
1620 ELASOC = SORT( ( AKSI( IP,IQ,IW)-AKSI( IP,IQ-1,IW) ) **2
      1      + ( ETA( IP,IQ,IW)-ETA( IP,IQ-1,IW) ) **2
      2      + ( ZETA( IP,IQ,IW)-ZETA( IP,IQ-1,IW) ) **2 )
      GO TO 1640
C
1630 IF( MRONT(IW) ) 1632,1632,1634
1632 ELASOC = ABS( 2.*ETA( IP,NNS-1,IW) )
      GO TO 1640
1634 KROOT = ( XNET(1,NNS,IW) + XNET(1+1,NNS,IW) ) / 2.
      1      YROOT = ( YNET(1,NNS,IW) + YNET(1+1,NNS,IW) ) / 2.
      2      ZROOT = ( ZNET(1,NNS,IW) + ZNET(1+1,NNS,IW) ) / 2.
      1      ELASOC = 2.*SORT( ( XROOT -AKSI( IP,NNS-1,IW) ) **2
      2      + ( YROOT -ETA( IP,NNS-1,IW) ) **2
      3      + ( ZROOT -ZETA( IP,NNS-1,IW) ) **2 )
C
1640 SUMGAM = 0.
      DO 1642 K=1,I
1642 SUMGAM = SUMGAM + GAMMA(K,LG-1,IW) - GAMMA(K,LG,IW)
1650 DGOLCH(IP,J) = SUMGAM / ELASOC
C
      DO 1700 IP=1,NVORC
      1      I = IP
      DO 1700 IQ=1,NVORS
      1      J = IQ
      DGOLSP(IP,IQ) = 0.5 * ( DGOLSP(IP,IQ) + DGOLSP(IP+1,IQ) )
      1700 DGOLCH(IP,IQ) = 0.5 * ( DGOLCH(IP,J) + DGOLCH(IP,J+1) )
C
      DO 1850 IP=1,NVORC
      1      I = IP
      DO 1850 IO=1,NVORS
      1      J = IO
      IF( IW-1 ) 1810,1810,1820
      1810 IT = ( IP-1)*NVORS + IO
      GO TO 1830
1820 IT = NVOR(IW-1) + IP-1)*NVORS + IO
1830 CONTINUE
      DELXSP = ( XNET(I,J+1,IW) + XNET(I+1,J+1,IW) ) / 2.
      1      DELYSP = ( YNET(I,J+1,IW) + YNET(I+1,J+1,IW) ) / 2.
      2      DELZSP = ( ZNET(I,J+1,IW) + ZNET(I+1,J+1,IW) ) / 2.
      1      EL = SORT( DELXSP**2 + DELYSP**2 + DELZSP**2 )
      COEFF = 0.5 * DGOLSP(IT,IQ) / EL
      DELV1(1) = COEFF * ( DELYSP * ROC(IT,3)
      1      - DELZSP * ROC(IT,2) )
      1      DELV1(2) = COEFF * ( DELZSP * ROC(IT,1)
      2      - DELXSP * ROC(IT,3) )
      1      DELV1(3) = COEFF * ( DELXSP * ROC(IT,2)
      3      - DELYSP * ROC(IT,1) )
C
      DELXCH = ( XNET(I+1,J,IW) + XNET(I+1,J+1,IW) ) / 2.
      1      DELYCH = ( YNET(I+1,J,IW) + YNET(I+1,J+1,IW) ) / 2.
      2      DELZCH = ( ZNET(I+1,J,IW) + ZNET(I+1,J+1,IW) ) / 2.
      1      EL = SORT( DELXCH**2 + DELYCH**2 + DELZCH**2 )
      COEFF = 0.5 * DGOLCH(IP,IQ) / EL
      DELV2(1) = COEFF * ( DELYCH * ROC(IT,3)
      1      - DELZCH * ROC(IT,2) )
      1      DELV2(2) = COEFF * ( DELZCH * ROC(IT,1)
      2      - DELXCH * ROC(IT,3) )
      1      DELV2(3) = COEFF * ( DELXCH * ROC(IT,2)
      3      - DELYCH * ROC(IT,1) )
      UIPP(IP,IQ) = UIP(IQ,IW) + DELV1(1) + DELV2(1)
      VIPP(IP,IQ) = VIP(IQ,IW) + DELV1(2) + DELV2(2)
      WIPP(IP,IQ) = WIP(IQ,IW) + DELV1(3) + DELV2(3)
      UPLW(IP,IQ) = UIP(IQ,IW) - DELV1(1) - DELV2(1)
      VPLW(IP,IQ) = VIP(IQ,IW) - DELV1(2) - DELV2(2)
      WPLW(IP,IQ) = WIP(IQ,IW) - DELV1(3) - DELV2(3)
      CUPP(IP,IQ) = 1. - ( UIP(IQ,IW) + DELV1(1) + DELV2(1) ) **2
      1      - ( VIP(IQ,IW) + DELV1(2) + DELV2(2) ) **2
      2      - ( WIP(IQ,IW) + DELV1(3) + DELV2(3) ) **2
      CPLW(IP,IQ) = 1. - ( UIP(IQ,IW) - DELV1(1) - DELV2(1) ) **2
      1      - ( VIP(IQ,IW) - DELV1(2) - DELV2(2) ) **2
      2      - ( WIP(IQ,IW) - DELV1(3) - DELV2(3) ) **2
      CUPP(IP,IQ) = CUPP(IP,IQ) / BET**2

```

[illegible]

```

2434 IF( ISYM -1 ) 2436, 2436, 2460
2436 DELX(I,T,1) = XNET(I,J,1,1W) - XNET(I,J,1W)
      DELZ(I,T,1) = YNET(I,J,1,1W) - YNET(I,J,1W)
      ZNET(I,J,1,1W) = ZNET(I,J,1,1W) - ZNET(I,J,1W)
      XMIDS = ( XNET(I,J,1W) + XNET(I,J,1,1W) ) / 2.
      YMIDS = ( YNET(I,J,1W) + YNET(I,J,1,1W) ) / 2.
      ZMIDS = ( ZNET(I,J,1W) + ZNET(I,J,1,1W) ) / 2.
      A3(I,T,1,2) = (2./IS*CHAR) * (XMIDS-ZCG)
      A3(I,T,1,3) = (2./IS*CHAR) * (YMIDS-YCG)
      A3(I,T,1,4) = (2./IS*CHAR) * (ZMIDS-XCG)
      A3(I,T,1,5) = (2./IS*SPAN) * (YMIDS-YCG)
      A3(I,T,1,6) = (2./IS*SPAN) * (ZMIDS-ZCG)
2460 CONTINUE
C
      PX = - GAMMA(I,LC,1W)
      PY = ( VMIDS(I,1Q,1W)*DELX(I,T,1) - VMIDS(I,1Q,1W)*DELZ(I,T,1) )
      PZ = ( VMIDS(I,1Q,1W)*DELZ(I,T,1) - VMIDS(I,1Q,1W)*DELX(I,T,1) )
      COSP(I,1Q) = - (2./S) * PX
      CYSPI(I,1Q) = (2./S) * PY
      CLSPI(I,1Q) = - (2./S) * PZ
      CMSP(I,1Q) = - PX * A3(I,T,1,2) + PY * A3(I,T,1,4)
      CRSP(I,1Q) = - PX * A3(I,T,1,3) + PY * A3(I,T,1,6)
      CRSP(I,1Q) = PZ * A3(I,T,1,5) - PY * A3(I,T,1,6)
      GO TO 2472, 2470, NSYM
2470 DELX(I,T,1) = - DELX(I,T,1)
      A3(I,T,1,3) = - A3(I,T,1,3)
      A3(I,T,1,5) = - A3(I,T,1,5)
2472 CONTINUE
C
      COTOT(ICASE) = COTOT(ICASE) + COSP(I,1Q)
      CYTOT(ICASE) = CYTOT(ICASE) + CYSPI(I,1Q)
      CLTOT(ICASE) = CLTOT(ICASE) + CLSPI(I,1Q)
      CMTOT(ICASE) = CMTOT(ICASE) + CMSP(I,1Q)
      CRTOT(ICASE) = CRTOT(ICASE) + CRSP(I,1Q)
      CRTOT(ICASE) = CRTOT(ICASE) + CRSP(I,1Q)
2500 CONTINUE
C
      IF( IPRCF ) 8670, 8670, 8600
8600 CONTINUE
      WRITE(6,8604) ISOLVE, ICASE, NSYM, ISYM
8604 FORMAT(// ROX, ISOLVE, ICASE, NSYM, ISYM)
      WRITE(6,8608) 1W WING PART, 11
8608 FORMAT(// 1W WING PART, 11)
      WRITE(6,8610) CLSPI(I,1Q)
8610 FORMAT(// CLSPI(I,1Q))
      DO 8612 I=1, NNC
8612 WRITE(6,8614) CLSPI(I,1Q), IQ=1, NVORS
8614 FORMAT(// CLSPI(I,1Q), IQ=1, NVORS)
      WRITE(6,8620) COSP(I,1Q)
8620 FORMAT(// COSP(I,1Q))
      DO 8622 I=1, NNC
8622 WRITE(6,8624) COSP(I,1Q), IQ=1, NVORS
8624 FORMAT(// COSP(I,1Q), IQ=1, NVORS)
      WRITE(6,8630) CYSPI(I,1Q)
8630 FORMAT(// CYSPI(I,1Q))
      DO 8632 I=1, NNC
8632 WRITE(6,8634) CYSPI(I,1Q), IQ=1, NVORS
8634 FORMAT(// CYSPI(I,1Q), IQ=1, NVORS)
      WRITE(6,8640) CMSP(I,1Q)
8640 FORMAT(// CMSP(I,1Q))
      DO 8642 I=1, NNC
8642 WRITE(6,8644) CMSP(I,1Q), IQ=1, NVORS
8644 FORMAT(// CMSP(I,1Q), IQ=1, NVORS)
      WRITE(6,8650) CRSP(I,1Q)
8650 FORMAT(// CRSP(I,1Q))
      DO 8652 I=1, NNC
8652 WRITE(6,8654) CRSP(I,1Q), IQ=1, NVORS
8654 FORMAT(// CRSP(I,1Q), IQ=1, NVORS)
      WRITE(6,8660) CNSPI(I,1Q)
8660 FORMAT(// CNSPI(I,1Q))
      DO 8662 I=1, NNC
8662 WRITE(6,8664) CNSPI(I,1Q), IQ=1, NVORS
8664 FORMAT(// CNSPI(I,1Q), IQ=1, NVORS)
8670 CONTINUE
2600 CONTINUE
      RETURN
C
3000 CONTINUE
C
      COMPUTE FORCES AND MOMENTS AT CHORDWISE MIDPOINTS
      DO 3600 IW=1, NWP
      NNS = NSPAN(IW)
      GO TO 3100, 3200, ISYM
3100 ISYM = 1
      J1 = 1
      J2 = NNS
      GO TO 3400
C
3200 ISYM = 2
      IF( MTIP(IW) ) 3210, 3210, 3220
3210 J1 = 2
      GO TO 3300
3220 J1 = 1
C
3300 IF( MROOT(IW) ) 3310, 3310, 3320
3310 J2 = NNS-1
      GO TO 3400
3320 J2 = NNS
C
3400 IPLE = 1 + NCHORD(IW) - NSCORD(IW)
      NVORC = NCHORD(IW) - I
      DO 3500 IP=IPLE, NVORC
      I = IP
      DO 3500 J=J1, J2
      LG = J+I
C
      IF( IW-1 ) 3420, 3420, 3422
      GO TO 3430
      IT = (IP-1)*NSPAN(IW) + J
3422 IT = NVEL(IW-1,3) + (IP-1)*NSPAN(IW) + J
3430 IF( IKJ -1 ) 3432, 3432, 3460
3432 IF( ICASE -1 ) 3434, 3434, 3460
3434 IF( ISYM -1 ) 3436, 3436, 3460
3436 DELX(I,T,2) = XNET(I+1,J,IW) - XNET(I,J,IW)
      DELZ(I,T,2) = YNET(I+1,J,IW) - YNET(I,J,IW)

```

```

DELT(1,2) = ZNET(1+1,J,IW) - ZNET(1,J,IW)
XMIIC = (XNET(1,J,IW) + XNET(1+1,J,IW)) / 2.
YMIIC = (YNET(1,J,IW) + YNET(1+1,J,IW)) / 2.
ZMIIC = (ZNET(1,J,IW) + ZNET(1+1,J,IW)) / 2.
A3(1,2,1) = (2./S*CRAR) * (XMIIC-XCG)
A3(1,2,2) = (2./S*CRAR) * (ZMIIC-ZCG)
A3(1,2,3) = (2./S*CRAR) * (YMIIC-YCG)
A3(1,2,4) = (2./S*CRAR) * (XMIIC-XCG)
A3(1,2,5) = (2./S*CRSPAN) * (YMIIC-YCG)
A3(1,2,6) = (2./S*CRSPAN) * (ZMIIC-ZCG)
3460 CONTINUE
C
SIJGAM = 0.
DO 3490 K=1,1
3490 SIJGAM = SIJGAM + GAMMA(K,LG-),)W) - GAMMA(K,LG,IW)
C
PX = - SIJGAM
PY = - SIJGAM
PZ = - SIJGAM
C
CCH(IP,J) = - (2./S) * PX
CYCH(IP,J) = - (2./S) * PY
CLCH(IP,J) = - (2./S) * PZ
CMCH(IP,J) = - PZ * A3(1,2,1) + PX * A3(1,2,2)
CMCH(IP,J) = - PX * A3(1,2,3) + PY * A3(1,2,4)
CMCH(IP,J) = - PZ * A3(1,2,5) - PY * A3(1,2,6)
3470 GO TO( 3472, 3470), NSYM
DELY(1,2) = - DELY(1,2)
A3(1,2,3) = - A3(1,2,3)
A3(1,2,5) = - A3(1,2,5)
3472 CONTINUE
C
COTOT(ICASE) = COTOT(ICASE) + CCH(IP,J)
CYTOT(ICASE) = CYTOT(ICASE) + CYCH(IP,J)
CLTOT(ICASE) = CLTOT(ICASE) + CLCH(IP,J)
CMTOT(ICASE) = CMTOT(ICASE) + CMCH(IP,J)
CMTOT(ICASE) = CMTOT(ICASE) + CMCH(IP,J)
CMTOT(ICASE) = CMTOT(ICASE) + CMCH(IP,J)
3500 CONTINUE
C
IF( IPRCF ) A770, A770, A700
CONTINUE
WRITE(6,8704) ISOLVE, ICASE, NSYM, ISYM
8704 FORMAT(// ROX : ISOLVE=,I1, ICASE=,I2, ISYM=,I1,
1 NSYM=,I1,
)
8705 WRITE(6,8708) W= WING PART =,I1 )
WRITE(6,8710)
8710 FORMAT(// CLCH(IP,J)
DO 8712 IP=1,PLF,NVORC
8712 WRITE(6,8714) IP,CLCH(IP,J), J=J1,J2
8714 FORMAT(// IP=,I2, 5F10.2
)
8720 WRITE(6,8720) CCH(IP,J)
DO 8722 IP=1,PLF,NVORC
8722 WRITE(6,8724) IP,CCH(IP,J), J=J1,J2
8724 FORMAT(// IP=,I2, 5F10.2
)
8730 WRITE(6,8730) CYCH(IP,J)
DO 8732 IP=1,PLF,NVORC
8732 WRITE(6,8734) IP,CYCH(IP,J), J=J1,J2
8734 FORMAT(// IP=,I2, 5F10.2
)
8740 WRITE(6,8740) CMCH(IP,J)
DO 8742 IP=1,PLF,NVORC
8742 WRITE(6,8744) IP,CMCH(IP,J), J=J1,J2
8744 FORMAT(// IP=,I2, 5F10.2
)
8750 WRITE(6,8750) CRCH(IP,J)
DO 8752 IP=1,PLF,NVORC
8752 WRITE(6,8754) IP,CRCH(IP,J), J=J1,J2
8754 FORMAT(// IP=,I2, 5F10.2
)
8760 WRITE(6,8760) CMCH(IP,J)
DO 8762 IP=1,PLF,NVORC
8762 WRITE(6,8764) IP,CMCH(IP,J), J=J1,J2
8764 FORMAT(// IP=,I2, 5F10.2
)
8770 CONTINUE
3600 CONTINUE
C
GO TO( 4050, 4020), NSYM
GO TO( 7000, 4200), ISYM
4050 COTOT(ICASE) = 2. * COTOT(ICASE)
CYTOT(ICASE) = 0.
CLTOT(ICASE) = 2. * CLTOT(ICASE)
CMTOT(ICASE) = 2. * CMTOT(ICASE)
CMTOT(ICASE) = 0.
CMTOT(ICASE) = 0.
4200 CONTINUE
AINCIN = ARCOS( COS(ETA*ANUIMM) ) * COS(RFTA*PSIIMM) )
1 INCIN = 180. / 3.1416
C
THE FOLLOWING 2 STATEMENTS ARE PECULIAR TO THE M-117 BOMB. THEY
PRESERVE AXIAL FORCE CORRECTIONS FOR INTERNAL BASE PRESSURE OF A
BODY OF REVOLUTION, AND SKIN FRICTION.
C
NELCAR = .4455 * 1.992 - .00018*AINCIN**21
COTOT(ICASE) = ( COTOT(ICASE) + NELCAR ) / RETA**2
1 COTOT(ICASE) = ( COTOT(ICASE) + .0935*ICF-1 )
CYTOT(ICASE) = CYTOT(ICASE) / RETA
CLTOT(ICASE) = CLTOT(ICASE) / RETA
CMTOT(ICASE) = CMTOT(ICASE) / RETA
CMTOT(ICASE) = CMTOT(ICASE) / RETA
CMTOT(ICASE) = CMTOT(ICASE) / RETA
7000 RETURN
END

```

```

C      SUBROUTINE AXES ( LM, MM, A, ANU, PSI, OMEGA )
C      THIS SUBROUTINE ASSUMES PITCH-YAW-ROLL ROTATION SEQUENCE
C      REF. THE PLANNER FOL-TDR-64-70 AD617354 P.23
C      DIMENSION A(3,3), ANU(LM), PSI(LM), OMEGA(LM)
C      COSNU = COS( ANU(MM) )
C      SINNU = SIN( ANU(MM) )
C      COSPSI = COS( PSI(MM) )
C      SINPSI = SIN( PSI(MM) )
C      COSOMG = COS( OMEGA(MM) )
C      SINOMG = SIN( OMEGA(MM) )
C      A(1,1) = COSNU * COSPSI
C      A(1,2) = - SINNU * SINPSI
C      A(1,3) = - SINNU * COSPSI
C      A(2,1) = SINNU * COSPSI
C      A(2,2) = COSNU * SINPSI
C      A(2,3) = COSNU * COSPSI
C      A(3,1) = SINNU * SINOMG + COSNU * SINPSI * COSOMG
C      A(3,2) = - SINNU * SINOMG + COSNU * SINPSI * SINOMG
C      A(3,3) = COSNU * SINOMG - SINNU * SINPSI * SINOMG
C      RETURN
C      END

```

Output Data

All of the output data from this step is written on a printer. Distributions of NUFF data, velocity, vorticity, pressure coefficient, and force coefficients can be optionally printed (these options are controlled by input data to the previous step.) When force coefficients are determined without performing trajectory calculations, the output data includes total force and moment coefficients and the orientation of the store with respect to its parent aircraft. When trajectories are calculated, total force and moment coefficients and solutions to the equations of motion are printed.

In performing the calculations for the sample run used to illustrate program operation, it was assumed that the downwash and side-wash vanished identically ($DOWNV = 0$ and $SIDEV = 0$) and the distribution of velocity magnitude was constant ($VMAG = 1.0$).

Representative results for this run (pressure coefficient distribution and total force and moment coefficients) are shown on the following page.

IW= WING PART =1 CPIPP(IIP,IQ)				IW= WING PART =2 CPIPP(IIP,IQ)				IW= WING PART =3 CPIPP(IIP,IQ)			
CSURP UPPER SURFACE				CSURP UPPER SURFACE				CSURP UPPER SURFACE			
IP= 5	-0.0709	-0.0840	-0.0942	IP= 1	-0.1434	-0.1648	-0.2494	IP= 1	-0.2205	-0.1740	-0.2404
IP= 6	-0.0969	-0.0595	-0.0998	IP= 2	-0.0712	-0.0927	-0.0119	IP= 2	-0.0964	-0.0852	0.0040
IP= 7	-0.1334	-0.0342	-0.1254	IP= 3	-0.0353	-0.0584	-0.0439	IP= 3	-0.0642	-0.0535	-0.0597
IP= 8	-0.1124	-0.0106	-0.1225	IP= 4	-0.0253	-0.0218	-0.0443	IP= 4	-0.0626	-0.0703	-0.0410
IP= 9	-0.0351	-0.0581	-0.0834	IP= 5	-0.0243	-0.0254	0.0303	IP= 5	-0.0723	-0.0290	0.0498
IP=10	0.0707	-0.0027	-0.0130	IP= 6	-0.0250	-0.0164	-0.1010	IP= 6	-0.0840	-0.0305	-0.0917
IP=11	0.1209	0.0273	-0.0074	IP= 7	-0.0295	0.0058	0.0542	IP= 7	-0.0972	-0.0148	0.0485
IP=12	0.0648	-0.0327	-0.0789	IP= 8	-0.0319	-0.0035	-0.0171	IP= 8	-0.1081	-0.0294	-0.0165
IP=13	0.0220	-0.0722	-0.1144								
IP=14	-0.0708	-0.1564	-0.1942	CPLDW(IIP,IQ)				CPLDW(IIP,IQ)			
IP=15	-0.1179	-0.1861	-0.2004	CSURP LOWER SURFACE				CSURP LOWER SURFACE			
IP=16	-0.0763	-0.1350	-0.1352	IP= 1	0.0933	0.1358	0.0821	IP= 1	0.1861	0.2154	0.1360
IP=17	-0.0904	-0.1420	-0.1330	IP= 2	0.0315	0.0579	0.1525	IP= 2	0.1077	0.1313	0.2058
IP=18	-0.1754	-0.2137	-0.1886	IP= 3	-0.0040	0.0649	0.0182	IP= 3	0.0714	0.1019	0.0831
IP=19	-0.2902	-0.3018	-0.2427	IP= 4	-0.0318	0.0380	0.0440	IP= 4	0.0464	0.0913	0.0869
IP=20	-0.3503	-0.3258	-0.2197	IP= 5	-0.0534	0.0029	0.0786	IP= 5	0.0284	0.0597	0.1273
IP=21	-0.2995	-0.2372	-0.0821	IP= 6	-0.0710	-0.0053	-0.0629	IP= 6	0.0146	0.0430	-0.0219
IP=22	-0.1908	-0.0903	0.1138	IP= 7	-0.0995	-0.0028	0.0748	IP= 7	-0.0007	0.0444	0.1041
IP=23	0.3894	0.4334	0.5740	IP= 8	-0.1056	-0.0314	-0.0374	IP= 8	-0.0166	0.0137	-0.0008
			0.7287				0.1419				0.1484

IPUN= 1 XORIGIN= 0.0 YORIGIN= 0.0 ZORIGIN= 0.0 PITCH(DEG)= 10.0 YAW(DEG)= 0.0 ROLL(DEG)= 0.0
 FORCE COEFFICIENTS FROM LAMINAR SKIN FRICTION ASSUMPTION
 FORCE COEFFICIENTS IN WIND AXES SYSTEM
 COINRA(1) CVISIO(1) FORCE CL(LIFT) CR(ROLL) CM(PITCH) CNIYAW
 IRUN= 1 0.12732 0.00009 0.40243 -0.00001 -0.15739 -0.00000
 FORCE COEFFICIENTS IN BODY AXES SYSTEM
 CAXIAL CSIDE CNORM CR(ROLL) CM(PITCH) CNIYAW
 IRUN= 1 0.02074 0.00009 0.61558 -0.00000 -0.15739 -0.00000

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1 ORIGINATING ACTIVITY (Corporate author) Arnold Engineering Development Center Arnold Air Force Station, Tennessee 37389		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A
3 REPORT TITLE AERODYNAMIC FORCES AND TRAJECTORIES OF SEPARATED STORES IN DISTURBED FLOW FIELDS		
4 DESCRIPTIVE NOTES (Type of report and inclusive dates) August 16, 1971, through June 30, 1972 -- Final Report		
5 AUTHOR(S) (First name, middle initial, last name) W. N. MacDermott and P. W. Johnson, ARO, Inc.		
6 REPORT DATE March 1973	7a. TOTAL NO OF PAGES 103	7b. NO OF REFS 6
8a. CONTRACT OR GRANT NO b. PROJECT NO c. Program Element 62602F d.	9a. ORIGINATOR'S REPORT NUMBER(S) AEDC-TR-72-162 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) ARO-PWT-TR-72-148	
10 DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.		
11 SUPPLEMENTARY NOTES Available in DDC.	12 SPONSORING MILITARY ACTIVITY Air Force Armament Laboratory, Air Force Systems Command, Eglin AFB, FL 32542	
13 ABSTRACT A vortex-lattice potential flow computer program capable of accepting nonuniform flow boundary conditions but previously restricted to incompressible flows with symmetry was modified to eliminate these restrictions. The program was structured in such a way that, after preliminary calculations of a purely geometric nature were performed one time for a given body, potential flow solutions for any set of boundary conditions on that body could be obtained in computer times measured in seconds rather than minutes. The aerodynamic characteristics of an M-117 bomb, represented by a network of 312 vortices, were calculated for uniform flow at a Mach number of 0.5 and were found to agree with wind tunnel measurements to within 10 percent, except for drag. The program was also used to compute forces on an M-117 bomb at a number of different locations in the disturbed flow field generated by an F-4C parent aircraft. In this case, absolute values of the force coefficients were generally in poor agreement with wind tunnel values, but the incremental variations of the calculated coefficients through the nonuniform flow field were within the range from 5 to 10 percent of wind tunnel measurements. A store separation routine was added to the potential flow program, and several representative store separation trajectories were calculated.		

DD FORM 1473
1 NOV 65

UNCLASSIFIED
Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
external stores vortex flow M-117 bomb F-4C aircraft wind tunnels uniform flow nonuniform flow grids potential flow computer programs mathematical models separation						

AFSC
Arnold AFB Texas

UNCLASSIFIED

Security Classification