AFFDL-TR-73-146-VOL. II

METHODS FOR PREDICTING THE AERODYNAMIC AND STABILITY AND CONTROL CHARACTERISTICS OF STOL AIRCRAFT

VOLUME II ★ STOL AERODYNAMIC METHODS COMPUTER PROGRAM

M. I. Goldhammer
N. F. Wasson

Douglas Aircraft Company
McDonnell Douglas Corporation

Approved for public release; distribution unlimited.

FOREWORD

This report summarizes work accomplished by the Douglas Aircraft Company, McDonnell Douglas Corporation, Long Beach, California, for the Air Force Flight Dynamics Laboratory, AFSC, Wright-Patterson Air Force Base, Ohio, under USAF Contract F33615-71-C-1861 (Project 643A). This document, presented in three volumes, constitutes the final report under the contract.

This work was accomplished during the period 1 July 1971 to 31 November 1973, and this report was released by the authors in December 1973. The Air Force Project Engineer was Mr. Henry W. Woolard of the Control Criteria Branch, Flight Control Division, AFFDL. His assistance in monitoring the work and providing necessary data is greatly appreciated.

The development of several computer programs later incorporated into the STOL Aerodynamic Methods Program was initiated under the sponsorship of the McDonnell Douglas Independent Research and Development Program (IRAD). Included among these is the Mark I version of the EVD Jet-Wing Computer Program which was developed in 1970. The EVD program was extensively rewritten and its capabilities expanded under Office of Naval Research Contract N00014-71-C-0250 during the period April 1971 to April 1972, resulting in the Mark II version of the EVD computer program. It is this program, further improved and expanded, that with several other programs have been integrated to provide a unified STOL aerodynamic methods computer package, designated as STAMP.

The authors gratefully acknowledge the contributions to the work presented in this volume made by Mr. N. D. Halsey (momentum induced drag method); Messrs. D. M. Friedman and A. L. Stelmak (Wooler Jet Program); Mr. W. V. Whitman (jet-wing geometry package); and Messrs. J. L. Hess, T. M. Ridell, and D. N. Smyth (matrix solution). The contributions of Dr. C. C. Shen to the development of the Mark I and II versions of the EVD Jet-Wing Computer Programs, which were written by Mr. N. F. Wasson, are especially appreciated. This work was performed under the technical direction of Mr. M. L. Lopez.

This report contains no classified information.

This technical report has been reviewed and is approved.

C. B. Westbrook
Chief, Control Criteria Branch
Flight Control Division
Air Force Flight Dynamics Laboratory

ABSTRACT

This report describes the STOL Aerodynamic Methods Program, which is intended to aid the engineer in the design and analysis of STOL aircraft employing internally ducted jet flaps, externally blown jet flaps, and mechanical flap systems with vectored thrust. The program provides capabilities to predict either the overall aerodynamic characteristics of a configuration or the aerodynamics of the following:

1. Jet-wing (in or out of ground effect)
2. Wing and vectored jets
3. Fuselage in the flow field of the jet-wing and/or vectored jets.
4. Empennage in the flow field of the jet-wing and/or vectored jets.

The program includes the capabilities for investigating the effects of arbitrary wing planforms with arbitrary high lift systems, including partial span flaps, slats, and jets; arbitrary camber, twist, and jet deflection; fuselages with arbitrary cross-sections and upsweep; arbitrary empennage arrangements, including conventional tails, mid-tails, and T-tails; and a capability to calculate the off-body flow field induced by the jet-wing and/or vectored jets at arbitrary points.

CONTENTS

LIST OF ILLUSTRATIONS

NOMENCLATURE

NAME	SYMBOL	DEFINITION
A	a	Fundamental case scale factor
A		Fuselage cross-sectional area
AC	ε_c	Local incidence due to camber
ACTE	ε_{ct}	Local incidence due to camber at trailing edge
ALFIN	$\alpha_{i\infty}$	Downwash angle or jet angle at infinity downstream
ALPHA	α	Angle of attack
AMAJ		Elliptical fuselage section major axis (Z-axis)
AMIN		Elliptical fuselage section minor axis (Y-axis)
AP		Derivative of fuselage cross-sectional area with respect to X
ARATIO	AR	Wing aspect ratio
AREA	S	Wing planform reference area
ARH	AR	Horizontal tail aspect ratio
AR	AR	Vertical tail aspect ratio
BETA	β	Deflection angle (δ_f, δ_J, or δ_S)
CAMBER		Horizontal or vertical tail camber angle
CCD	$(C_{D_i})_p$	Total induced drag coefficient calculated by pressure integration
CCJ	C_J	Total jet sheet momentum coefficient
CCL	C_L	Total lift coefficient
CCM	C_m	Total pitching moment coefficient (about wing apex)
CCS	C_S	Total leading edge suction coefficient
CCT	C_T	Total net thrust coefficient
CCY	C_Y	Total side force coefficient
CD	cd_i	Wing sectional induced drag coefficient
CDH	$(cd_i)_H$	Horizontal tail sectional induced drag coefficient

NOMENCLATURE

NAME	SYMBOL	DEFINITION
CDITZ	$(c_{d_i})_M$	Wing total induced drag coefficient, calculated by the momentum method
CDV	$(c_{d_i})_V$	Vertical tail sectional induced drag coefficient
CHORD	c	Wing sectional chord
CJ	C_J	Vectored jet thrust coefficient
CL	c_ℓ	Wing sectional lift coefficient
CLH	$(c_\ell)_H$	Horizontal tail sectional lift coefficient
CLL	C_l	Wing rolling moment coefficient
CLLH	$(C_l)_H$	Horizontal tail rolling moment coefficient
CLLP	C_{l_p}	Rolling moment coefficient derivative due to rolling
CLLR	C_{l_r}	Rolling moment coefficient derivative due to yawing
CLLV	$(C_l)_V$	Vertical tail rolling moment coefficient
CLQ	C_{L_q}	Lift coefficient derivative due to pitching
CM	c_m	Sectional pitching moment coefficient (about the local leading edge)
CM	$(C_m)_F$	Fuselage pitching moment (about XMC)
CMAC	$\bar{\bar{c}}$	Mean aerodynamic chord (MAC)
CMH	$(c_m)_H$	Horizontal tail sectional pitching moment coefficient (about local leading edge)
CMQ	C_{m_q}	Pitching moment coefficient derivative due to pitching about center of gravity
CMU	c_μ	Sectional jet sheet momentum coefficient
CN	C_n	Wing or fuselage yawing moment coefficient
CNI	C_{n_i}	Total yawing moment coefficient due to induced drag (including leading edge suction)
CN(P)	C_{n_p}	Yawing moment coefficient derivative due to rolling, dependent on rolling rate
CN(R)	C_{n_r}	Yawing moment coefficient derivative due to yawing, dependent on yawing rate

NOMENCLATURE

NAME	SYMBOL	DEFINITION
CNV	$(c_n)_v$	Vertical tail sectional yawing moment coefficient (about local leading edge)
CP	Δc_p	Pressure coefficient, $\Delta c_p = c_{p_{\text{lower surface}}} - c_{p_{\text{upper surface}}}$
CREF	$\tilde{c}$	Wing reference chord
CS	c_s	Sectional leading edge suction coefficient
CTO	c_t	1. Sectional thrust coefficient 2. Tip chord of wing
CXCL	$\frac{x_{C.L.}}{\tilde{c}}$	x-coordinate of center of lift, in terms of reference chord CSCL = - (CCMG + CCML)/CCL labeled (X/CREF)
CXCLB	$\frac{x_{C.L.}}{b/2}$	x-coordinate of center of lift, in terms of half span CXCLB = - (CCMG + CCML)/CCL *CREF labeled (X/B/2)
CXCP	$\frac{x_{C.P.}}{\tilde{c}}$	x-coordinate of center of pressure, in terms of reference chord
CXCPB	$\frac{x_{C.P.}}{b/2}$	x-coordinate of center of pressure, in terms of half span
CY(P)	C_{Y_p}	Side force coefficient derivative due to rolling, dependent on rolling rate
CY(R)	C_{Y_r}	Side force coefficient derivative due to yawing, dependent on yawing rate
CYV	$(c_y)_v$	Vertical tail sectional side force coefficient
D	d	Chordwise distance from trailing edge to infinity EVD vortex point
DO	D_o	Vectored jet equivalent exit diameter
DEL	$\bar{\delta}$	Chordwise length of an EVD element in terms of local chord
DELTA	$\bar{\Delta}$	Half of the spanwise width of an EVD element or a spanwise division, normalized by b/2
DELZ	Δz	Vectored jet incremental spacing size
DIHED	Ω	Vectored jet dihedral angle (measured from the x-z plane)

NOMENCLATURE

NAME	SYMBOL	DEFINITION
DJ	δ_J	Jet deflection angle relative to the mean camberline at the trailing edge
EPS	ε_i	Total local incidence angle at a downwash control point on the wing
GAMMA	γ	Vorticity intensity
GHITE	h	Height of wing above the ground
H	Δz	Vectored jet incremental spacing size
HL	X_H	Chordwise distance of horizontal tail apex from origin
HL	h_ℓ	Height of wing leading edge above the x-y plane
HH	Z_H	Vertical distance of horizontal tail above wing at zero angle of attack
I	i	EVD element sequence number
ICAMF		Fuselage camber input flag
ICAMH		Horizontal tail camber input flag
ICAMV		Vertical tail camber input flag
ICT		Camber type flag for each wing section
ICTYPE		Wing chordwise division type flag for each section
IDERIV		Stability derivative flag
IFFLD		Flow field control flag
IGRND		Ground effect control flag
IGTYPE		Wing planform geometry type flag
IHC		Horizontal tail camber section type flag
IHINGE		Hinge EVD control flap
IHT		Hinge type flag for each wing section
IHX		Horizontal tail chordwise division section type flag
IJTYPE		Jet sheet chordwise division type flag for each section

NOMENCLATURE

NAME	SYMBOL	DEFINITION
ILT		Flap type flag
INBETA		Hinge angle input flag
INCAMB		Camber angle input flag
INDELJ		Jet angle input flag
INHITE		Leading edge height input flag
INTWST		Twist angle input flag
IPRINT		Printed output control flag
ISECT		Fuselage section type flag
ISY		Vectored jet symmetry flag
ISYMM		Jet-wing symmetry flag
IVC		Vertical tail camber section type flag
IVX		Vertical tail chordwise division section type flag
J	J	Sectional jet sheet momentum
JETFLG		Jet control flag
JETREP		Vectored jet parameter repeat flag
K	k	Sequence number of spanwise divisions or sections
LAST		Last vectored jet input flag
MAC	$\bar{\bar{c}}$	Mean aerodynamic chord of horizontal or vertical tail
N		Fundamental case sequence number
NC		Number of camber angles on a horizontral or vertical tail section
NCASES	K	Number of fundamental cases
NCT		Number of wing camber types
NCTYPE		Number of horizontal and vertical tail camber section types
NF		Number of elements in a fuselage segment

NOMENCLATURE

NAME	SYMBOL	DEFINITION
NHT		Number of hinge types
NI		Number of chordwise divisions of a particular chordwise division type (wing or jet)
NJ		Number of chordwise divisions of jet section
NJTYPE		Number of jet chordwise division types
NPOINT		Number of off-body flow field points input
NROWH		Number of horizontal tail sections
NROWS		Number of jet-wing sections
NROWSJ		Number of jet-wing sections having a trailing jet sheet
NROWV		Number of vertical tail sections
NSEG		Number of fuselage segments
NW		Number of chordwise division of each wing section
NWTYPE		Number of wing chordwise division types
NX		Number of chordwise divisions of each horizontal or vertical tail section
NXTYPE		Number of horizontal and vertical chordwise division types
R		Fuselage sectional radius
RADIUS		Fuselage sectional radius
RP		Derivative of fuselage sectional radius with respect to X
SEG		Fuselage segment number
SPAN	b	Wing span
SPANH	b_H	Horizontal tail span
SPANV	b_V	Vertical tail span
SWEEP	$\Lambda_{c/4}$	Wing quarter-chord sweep angle

NOMENCLATURE

NAME	SYMBOL	DEFINITION
SWH	$(\Lambda_{c/4})_H$	Horizontal tail quarter-chord sweep angle
SWV	$(\Lambda_{c/4})_V$	Vertical tail quarter-chord sweep angle
SSLIP	β	Aircraft sideslip angle (used only in fuselage analysis)
TANLE	$\tan \Lambda_\ell$	Tangent of wing local leading edge sweep angle
TCR	t/c	Wing thickness/chord ratio
THETA	Θ	Total jet sheet deflection angle relative to the free stream
THETA	Θ	Local incidence of vectored jet centerline relative to the freestream
THETAJ	Θ_J	Vectored jet exit angle relative to the freestream
THO	Θ_J	Vectored jet exit angle relative to the freestream
TITLE		Run title
TR	λ	Wing taper ratio
TRH	λ_H	Horizontal tail taper ratio
TRV	λ_V	Vertical tail taper ratio
TWIST	α_0	Twist angle for each wing section
U	U	Free stream velocity
U	u	Onset perturbation velocity
UINF	U_∞	Free stream velocity
UJ	V_J	Vectored jet local velocity
UJO	V_{J_0}	Vectored jet exit velocity
UR UX	U_r U_x	Fuselage cylindrical coordinate system perturbation velocities
V	v	Sidewash perturbation velocity
VH	Z_V	Vertical distance from vertical tail apex to origin, at zero angle of attack

NOMENCLATURE

NAME	SYMBOL	DEFINITION
VL	X_v	Chordwise distance from vertical tail apex to origin, at zero angle of attack
VOLUME	$\overline{V}_H$, $\overline{V}_v$	Horizontal or vertical tail volume
VX VY VZ	u v w	Flow field perturbation velocities induced by vectored jets
W	w	Downwash perturbation velocity
X	x	Chordwise coordinate of the Cartesian coordinate system
XB	$\overline{x}$	Chordwise distance of an EVD element from the leading edge, in terms of local chord
XBH	x_h	Chordwise distance of a hinge point from the leading edge, in terms of local chord
XBT	$\overline{x}$	Chordwise distance of a vertical or horizontal tail element from the leading edge, in terms of local chord
XCL/C	$\overline{x}_{c.l.}$	Chordwise location of center of lift at a section measured from local leading edge, in terms of local chord
XCP/C	$x_{c.p.}$	Chordwise location of center of pressure at a section measured from local leading edge, in terms of local chord
XCG	$x_{c.g.}$	x-coordinate of the center of gravity about which pitching rate derivatives are taken
XF		Chordwise location of fuselage crossectional cut
XI	x_i	x-coordinate of an EVD element
XLEAD	x_ℓ	x-coordinate of the leading edge
XMC	$X_{m.c.}$	x-coordinate of the moment center about which pitching moments are taken
XTRAIL	x_t	x-coordinate of the trailing edge
XO	x_o	Chordwise coordinate of vectored jet exit
Y	y	Spanwise coordinate of the Cartesian coordinate system

NOMENCLATURE

NAME	SYMBOL	DEFINITION
YH	y_H	Spanwise coordinate of the Cartesian coordinate system
YMC	y_{mac}	Spanwise location of horizontal tail mean aerodynamic chord
YO	y_o	Spanwise coordinate of vectored jet exit
Z	z	Vertical coordinate of the Cartesian coordinate system
ZCL		Vertical coordinate of fuselage segment centerline
ZLEAD	z	z-coordinate of vertical tail sectional leading edge
ZV	z_v	Spanwise location of a vertical tail section centerline
ZO	z_o	Vertical coordinate of vectored jet exit

Subscripts

SUFFIX	SUBSCRIPT	DEFINITION
A	α	Indicates a linear variation with angle of attack
A2	α^2	Indicates a quadratic variation with angle of attack
	e	Indicates elevator quantity
G	Γ	Indicates a contribution due to pressure (circulation)
J	J	Indicates a contribution due to jet reaction
MC	m.c.	Indicates pitching moments taken about moment center
MU	μ	Indicates a contribution due to jet reaction at a wing section
	r	Indicates rudder quantity
R2	r^2	Indicates a quadratic variation with yawing rate
T	t	Indicates a contribution due to the thrust component of the jet reaction at a section
X	α	Indicates a contribution from mutual interference between the basic configuration and the angle of attack

NOMENCLATURE

SUFFIX	SUBSCRIPT	DEFINITION
0	o	Indicates $\alpha = 0$
THETA	Θ	Indicates non-symmetric fuselage cross-flow velocity

1.0 INTRODUCTION

This report describes the Mark I version of the STOL Aerodynamic Methods Program (STAMP) which is intended to aid in the preliminary aerodynamic design and analysis of STOL transport aircraft employing internally ducted jet flaps, externally blown jet flaps, or mechanical flap systems with vectored thrust. The computer program package is based on theoretical methods which are disucssed in detail in Volume I of this report. The program is written in the Fortran IV language and can be readily adapted for use on many large computer systems. Versions are currently in operation on the CDC 6000 series and the IBM 370 series computers.

The STAMP computer program package is basically a collection of computer programs, each of which is designed to analyze a particular aircraft component, plus interference methods to account for the influence of one component on another. Although intended for the analysis of complete aircraft configurations, the program has been structured so that components can be analyzed separately or in any desired combination. Within the limitations of each theoretical method, the program calculates aerodynamic and stability and control data for arbitrary jet-wings (internally ducted and externally blown), for vectored jets, and for arbitrary fuselage and empennage arrangements. These data can be calculated both in free air and in proximity to the ground.

The STAMP computer program is a powerful analytical tool for the preliminary aerodynamic design and analysis of STOL transport aircraft. However, efficient and effective use of the program can only be achieved by personnel thoroughly familiar with the range of applicability of the various theoretical methods, with the limitations of the methods, and with the techniques necessary to translate a complex aircraft geometry into numerical data to be read into the computer. The user must be willing to apply his own engineering knowledge and judgment in a methodical manner to each component of a configuration to be analyzed. It is suggested that Volume I of this report be read to gain an understanding of the contents of the program; and that Volume III be read to gain familiarity with aircraft geometry preparation and output data interpretation before running the program; and that a few simple problems be tried in order to develop understanding and familiarity with the program before large complicated problems are attempted. Proceeding in this manner may prevent the waste of large amounts of computer time or, even worse, the generation and use of erroneous aerodynamic data.

2.0 PROGRAM STRUCTURE

The STAMP computer program package is composed of several major subprograms, each of which is intended to analyze a particular STOL aircraft component, plus subprograms designed to calculate the aerodynamic interferences between the various components. The primary components of STAMP are:

- Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Method (including ground effect)
- Jet-Wing Flow Field Method
- Modified Slender Body Fuselage Method
- Empennage Method
- Vectored Jet Flow Field Method

Except for the sharing of common geometric data, each of these major subprograms has been written so that it can be used independently of the STAMP package. Thus, maximum flexibility has been built into the program to satisfy the needs of various users.

The structure of the program is formed in the executive control subroutine (STAMP) which determines which subprograms will be executed and in which order, based on the user inputs. Understanding the logic of the executive control subroutine is the key to understanding the program structure. Figure (2.1) gives a pictorial view of the logical flow during execution of the program. The main control program can access any of the four individual aerodynamic methods and execute that portion of the program independently of the other methods. In addition, interference methods can be accessed to calculate the influence of the jet-wing on the empennage and fuselage or the influence of vectored jets on the wing, empennage, and fuselage. The order of program execution will vary with the particular desires of the user, but the blocks of figure (2.1) have been numbered to indicate the order of execution for the most general case.

GENERAL PROGRAM STRUCTURE

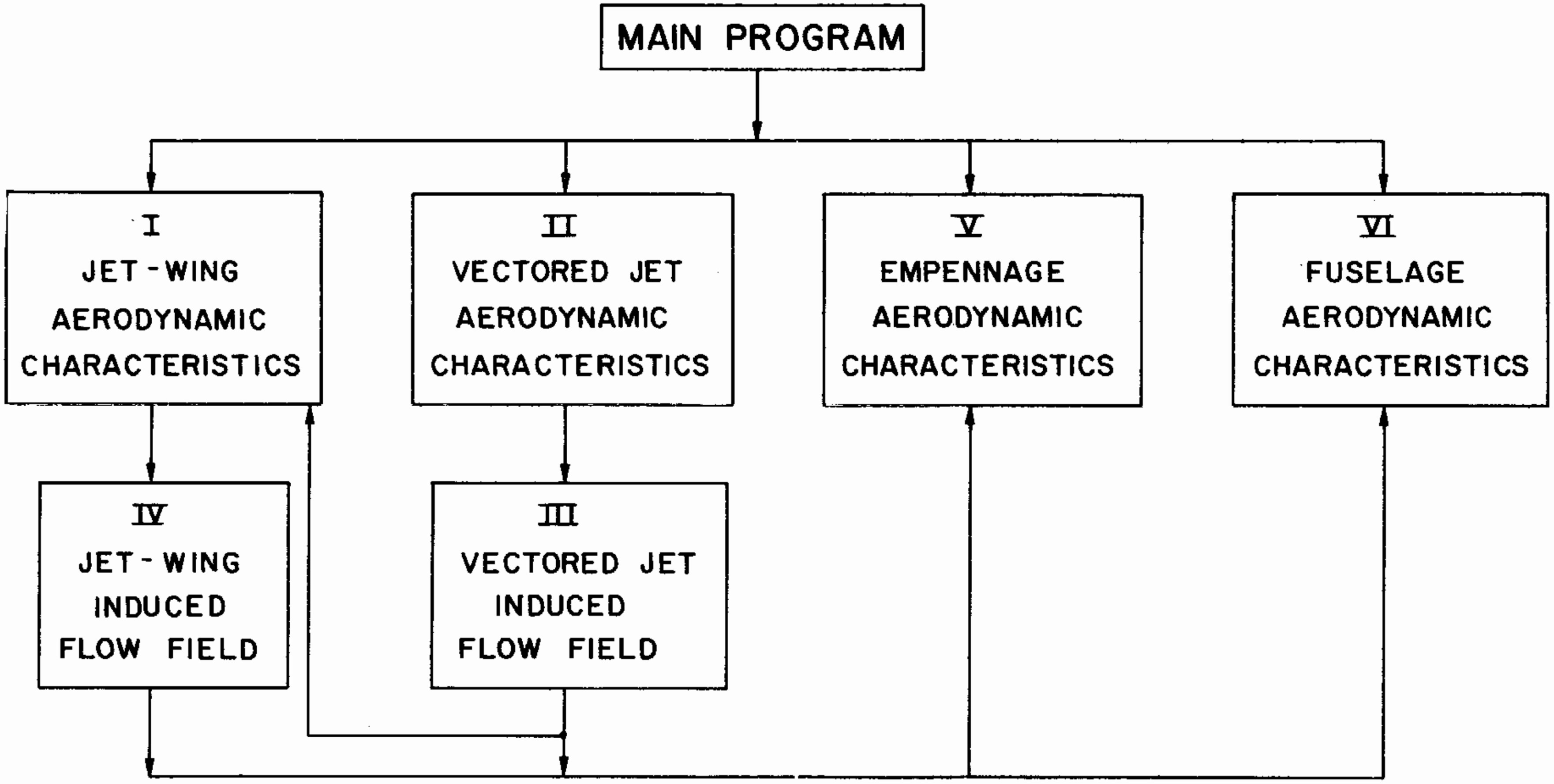


Figure (2.1)

3.0 METHODS CAPABILITIES AND LIMITATIONS

Familiarization with the capabilities and limitations of the various methods in the STAMP package is a necessity for effective use of the program. Volume I of this report presents the theoretical aspects of the various methods and Volume III discusses the engineering applications and extensions of the methods. Here the basic assumptions and restrictions of the theoretical methods are briefly reviewed for the convenience of the reader who has already read Volumes I and III.

3.1 EVD Jet-Wing Lifting Surface Method

The EVD Jet-Wing Lifting Surface Method is a linearized lifting surface theory developed to analyze the aerodynamic characteristics of arbitrary planform wings with an arbitrary distribution of jet momentum issuing from the trailing edge. The method provides a capability to calculate the chordwise and spanwise loading distributions and the total aerodynamic characteristics of a jet-wing and provides the capability to investigate the effects of:

- a. Part-span flaps
- b. Part-span blowing
- c. Arbitrary camber, twist, and jet deflection
- d. Arbitrary jet momentum
- e. Pitching, rolling, and yawing rates, and sideslip
- f. Ground proximity

As discussed below, the method and computer program are structured to take maximum advantage of the linearized nature of the approach.

3.1.1 Brief Review of Linearized Approach

The basic assumptions adopted in the EVD jet-wing lifting surface theory are summarized below:

General Flow Field:

- a. Inviscid, incompressible, and irrotational.
- b. No mixing between the external flow and the jet (i.e., no entrainment into the jet).

c. All streamlines are parallel to the x-z plane (no spanwise components of velocity or jet momentum).

d. Roll up of the jet sheet and wing trailing vortex system is neglected.

Wing:

a. The wing is assumed to be thin and is represented by a mean camber line. This allows the boundary condition of no flow normal to the wing surface to be transferred from the actual wing surface to the mean camber line.

b. All incidences on the wing surface are small, such that the sine of the incidence angles can be replaced by the angle itself. The incidences may be discontinuous.

Jet:

a. The jet sheet is thin.

b. The jet sheet deflection is small, such that the jet sheet can be assumed to remain in the plane of the wing.

The above assumptions imply the existence of certain other conditions which are of practical importance in use of the program. Though they may not seem to be obviously derived from the basic assumptions, these conditions are, nevertheless, controlling factors in the application of the method, and must be recognized and considered by the user. Such conditions include the following:

a. The wing and jet are represented by planar sheets of bound and trailing vorticity.

b. All potential field influences of the wing and jet originate in the x-y plane and all boundary conditions are satisfied in that plane.

c. The jet issues from the trailing edge of the wing, and its reaction force acts there also.

d. There are no losses or diffusion in the jet sheet behind the wing trailing edge.

e. There can be no gaps or slots in the wing (program restriction only).

f. Since the lifting surface theory is for a jet-wing system alone, the effects of a fuselage, nacelles, empennage, etc., are not accounted for. Fuselage and empennage effects are treated separately using approximate interference techniques and are discussed in subsequent sections.

g. Dynamic stability derivatives are calculated in a quasi-steady state only. That is, the disturbances induced by a dynamic motion are assumed to propagate instantaneously and no time lag is considered.

3.1.2 The Fundamental Case

An important feature of the EVD jet-wing computer program is the concept of the fundamental case. The use of fundamental cases is intended to increase the versatility and computational efficiency of the program and to make use of the information generated by the program easier for the user.

The fundamental case capability of the EVD method is a direct consequence of the linearization of the problem. This is fully discussed in Volume I and will not be repeated here. Fundamental cases are used to separate the aerodynamic influences of various types of jet-wing surface deflections so that these deflections and their associated loadings can be linearly combined to form a large number of jet-wing configurations. The types of deflections which may be treated as fundamental cases are as follows:

a. Wing camber
b. Wing twist
c. Leading edge and trailing edge flaps
d. Jet deflection

The fundamental case concept is not applicable either to the jet momentum coefficient, C_J, or to the planform of the wing. Variations in these quantities require reformulation of the problem solution.

The fundamental cases produce loadings which are linearly related to the deflection magnitude. Therefore, the loading of a particular jet-wing configuration can be calculated by summing the loadings of the fundamental cases. However, since the EVD method is a linearized lifting surface theory, the surface deflections used in either fundamental or composite cases should be limited to the range of angles where linear behavior is likely. Extrapolation of linearized loading characteristics into regions of high deflection angles could produce erroneous results.

3.1.3 The Composite Case

The purpose of the composite case capability in the EVD program is to provide the user with data for a large number of related jet-wing configurations at minimum cost. A composite case can be defined as a jet-wing configuration formed by linearly combining a set of fundamental cases to form the proper surface deflection angles, as

$$\varepsilon(x,y) = \sum_{m=1}^{K} a_m \, \varepsilon_m(x,y) \tag{3.1}$$

where $\varepsilon(x,y)$ are the composite surface deflections, $\varepsilon_m(x,y)$ are the surface deflections for the m^{th} fundamental case, and a_m is a modulation factor for the m^{th} fundamental case. Because of the linearization of the problem, the composite case loading can be similarly formed from the fundamental case loadings, as

$$\gamma(x,y) = \sum_{m=1}^{K} a_m \, \gamma_m(x,y) \tag{3.2}$$

Thus, by solving the basic jet-wing problem for a small number of fundamental cases, any number of composite cases can be formed by simple summation of solutions. The user of the program is again cautioned that, when using composite cases, the range of deflections where linear behavior is likely should not be exceeded. Also, when considering complex flap systems, the user is cautioned to limit the range of deflections so as not to significantly change the planform by flap chord extension.

3.1.4 Ground Effect Method

The jet-wing ground effect method developed for the STAMP package is based on a set of assumptions similar to those used in the EVD theory. That is, in addition to the basic assumptions concerning the nature of the flow field, the same assumptions regarding small deflection angles and shallow jet displacement have been assumed. It has been shown in Volume I, however, that these linearization assumptions for the ground effect problem involve an order of approximation one order lower than for the free air case.

In addition to the limitations imposed on the EVD method because of the linearization assumptions, there are limitations that must be imposed on the ground effect method because of the presence of the ground. Ground heights must be restricted to the range where no significant impingement of the jet on the ground occurs and to the range where the portion of the jet in which significant turning occurs is not "too close" to the ground. In addition, the wing must not be "too close" to the ground. There is no absolute definition for the term "too close," but in general it can be said that if variations in wing-ground height (either spanwise or chordwise) are of the same order of magnitude as the mean ground height, then the linearization assumptions are likely to be inadequate. The reader is referred to Volume I for a more complete discussion of the limitations of the ground effect method.

Although the ground effect method has been formulated subject to linearization assumptions similar to those in the EVD method, it has been shown (Volume I, Section 2.2.1.5) that, while the vortex strengths are linear in the surface deflections, the surface pressure coefficients are not because of the presence of an image system in the ground effect solution. Therefore, the nature of the fundamental/composite case feature of the solution is changed for ground effect. Separating a configuration into a set of fundamental cases and then combining the fundamental cases into a set of composite cases still offers the user a considerable savings in computation time because the solution matrix need be solved only once, but the fundamental case aerodynamic coefficients cannot be linearly combined to form composite case aerodynamic coefficients because of the non-linearity of the pressure coefficient. The aerodynamic coefficients for each composite case, therefore, must be integrated from the fundamental case vorticity distributions. Hence, the ground effect solution portion of STAMP has been structured so that only the composite case loadings are computed.

3.2 Jet-Wing Flow Field Method

The jet-wing flow field method incorporated in STAMP is an approximate analytical method for calculating perturbation velocities induced at arbitrary points in the vicinity of a jet-wing powered lift system. A fully non-planar mathematical model to represent the wing and jet sheet is employed, and on it is placed a vortex sheet whose strength is determined from the linearized EVD loadings. The shape of the jet is also calculated from the linearized loadings. The vorticity distribution has been simplified from the continuous distribution of the EVD method to a lattice of concentrated horseshoe vortices, which are considerably simpler mathematically, yet induce the same velocity field, except for a small region on either side of the jet-wing sheet.

Because of its non-planar nature, the flow field method is non-linear in all the deflections (i.e., camber, twist, flap deflection, jet deflection, and angle of attack) even though the vorticity distribution used is linear. Hence, the fundamental case loses its importance since flow field perturbabations cannot be linearly combined and, therefore, the computer program has been structured so that flow field calculations are performed only for composite cases.

A restriction must be applied to the flow field method. Use of the concentrated singularity approach to represent the wing and jet sheet, while desirable because of its mathematical simplicity, restricts the range of applicability of the method to the "far field" only. This is fully discussed in Section 2.3 of Volume I. It is reiterated here, however, that use of the flow field method is restricted to points that are no closer than, say, one element length from the wing or jet sheet, where an element length is defined as the maximum linear dimension of a jet-wing element nearest to the flow field point of interest. Induced velocities for points which fall close to the singularity sheet are likely to be unreasonably large because of the discrete singularity model employed and should be readily apparent to the user.

3.3 Vectored Jet Flow Field Method

The method of Wooler (reference 4) has been incorporated into STAMP for the analysis of the aerodynamic interference effects of vectored jets on an aircraft. This is an approximate method which treats a circular viscous jet issuing at an arbitrary angle into a uniform stream. The jet envelope is assumed to lie far enough from all aircraft components so that the following conditions apply:

a. The jet envelope shape is not disturbed by the presence of the aircraft.

b. Only the "far field" perturbations of the jet flow field have an influence on the aircraft.

The "power-off" solution is performed in the normal manner for a jet-wing with no jet sheet, but with the tail and fuselage included. For the vectored jet "power-on" condition, however, only composite cases are treated, and the wing loading must be constructed by superposition of the linear fundamental case loadings and the non-linear jet-induced loading. This induced loading is found by treating the jet-induced velocity perturbation distribution as equivalent wing camber.

The exit location of each jet translates as the aircraft rotates through angles of attack about the wing apex (origin), and the jet exit angle relative to the freestream also changes. Because of the non-linearity of the jet-induced flow field with jet exit location and angle, the program is structured to consider only the angles of attack of 0, 5, 10, 15, and 20 degrees. The jet envelope and flow field are recalculated at each of these angles. The jet flow field perturbations are then superimposed on the wing, and the resulting "power-on" wing loading is computed. The wing flow field is computed using this power-on loading and the combined wing and vectored jet flow field perturbations are then superimposed on the empennage and fuselage.

As many as ten vectored jets may be treated as one configuration and each jet may have an additional corresponding image jet about the $y = 0$ plane. The power-on portion of the program may be repeated for as many new jet configurations as desired. Any change in the strength, exit diameter, exit angle or exit location of any jet constitutes a new jet configuration.

The user must be careful to see that the jet envelope does not impinge on nor pass very near any part of the aircraft. If the jet passes near the aircraft, the far-field assumptions will be violated and the interference effects may be invalid. If the jet impinges on an aircraft component, the solution may be completely meaningless because of extremely large perturbation velocities arising on control points inside the jet envelope. If a control point is found inside the jet envelope, a warning message is printed, but no corrective action is taken by the program. In addition to jets passing near the aircraft, neighboring jets merging and mixing into each other cannot be treated accurately, since each jet is assumed to be isolated from all outside disturbances.

Cases of unsatisfactory jet interference solutions due to very close proximity of the jet to the aircraft are most likely to occur under conditions of low jet strength (i.e., jet exit velocity of the same order as the freestream velocity) and extreme exit angles (i.e., θ_0 near $0°$ or $\pm 180°$). In such cases the jet cannot penetrate very far normal to the freestream, but is rapidly turned to the freestream direction.

In ground effect the jet envelope is first computed as though there were no ground plane. For computing the flow field perturbation velocities, however, all jet segments after the one which impinges on the ground are assumed to lie on the ground plane.

Other factors affecting the jet interference solution accuracy are the jet segment spacing size and the downstream location at which the jet envelope is truncated. These effects are discussed in detail in Section 4.2.2.

3.4 Fuselage Method

The fuselage analysis method incorporated in STAMP is an approximate method based, in part, on slender body theory. The method is capable of considering, in an approximate sense, bodies with non-circular cross-sectional geometry, body camber (e.g., fuselage upsweep), and jet-wing or vectored jet induced downwash and sidewash on the fuselage. There is no capability in the method to consider flow separation from the fuselage afterbody, which is often significant for an upswept body at angle of attack typical of transport aircraft. In addition, the present method cannot consider the effects of ground proximity on the fuselage. Finally, since the mathematical model used to represent the fuselage is composed of sources and doublets only, the total force coefficients calculated by this method will be identically zero; however, the method does calculate non-zero pitching and yawing moments.

The slender body nature of the method limits its range of applicability to bodies that satisfy the slender body assumption that the cross-sectional radius changes slowly with distance along the axis (i.e., dR/dx is small). It must be emphasized that the slender body assumption is not valid at or near stagnation points of the body or near the ends of a body with rounded ends where dR/dx is not small compared to unity. For many bodies of interest, however, the regions where the slender body assumption is not valid are small.

Although the fuselage method does allow for non-circular body cross-sections and body camber, the method of treating these effects requires that the body be nearly axisymmetric. In other words, the body cross-sections should be sufficiently circular and the body axis should be sufficiently straight so that there are no significant non-axisymmetric velocity perturbations (i.e., u_θ velocities). For example, this method should not be used to determine the pressure distribution on a large fuselage protuberance, such as a radome, but should be adequate for milder protuberances such as landing gear fairings.

3.5 Empennage Method

The empennage method incorporated in STAMP is a linearized multiplanar lifting surface theory utilizing a vortex lattice to represent the loading distribution. The basic assumptions of the method are identical to those employed in the EVD method (Section 3.1) except that there is no jet capability in the empennage method and potential field influences are allowed to originate in a plane parallel to the x-y plane for the horizontal tail and in the x-z plane for the vertical tail.

The program has been structured such that only three fundamental cases can be considered for the isolated empennage: unity angle of attack on the horizontal tail, unity sideslip angle on the vertical tail, and tail camber. Elevator or rudder power $\frac{\partial C_m}{\partial \delta_e}$ on $\frac{\partial C_n}{\partial \delta_r}$ can be determined from the camber case by specifying the only camber as either unity elevator deflection or unity rudder deflection, respectively. For the jet-wing/empennage interference problem, the program has been structured to treat the actual composite case. That is, a complete jet-wing composite case is used to generate the jet-wing induced flow field velocities, and these velocities are then applied to the empennage which is treated as cambered (according to user input) and at the same angle of attack as the wing.

Since the empennage program is capable of treating intersecting lifting surfaces, extreme caution must be exercised near the region of intersection to prevent a singular solution. It is possible for a singularity to occur in the solution if the discrete horseshoe vortex of one tail surface intersects a normal-wash collocation point of the other tail surface. Therefore, it is recommended that to prevent the possibility of a singular solution that the line of intersection of the tail surfaces be a spanwise section edge for both a horizontal tail section and a vertical tail section.

4.0 FUNDAMENTALS OF PROGRAM USE

Since the analytical methods that comprise STAMP require some type of numerical definition of the various component geometries, it is necessary to establish a set of guidelines that can aid the user in preparing the geometry input. Proper preparation of geometric data is necessary for effective program utilization not only to ensure that a valid numerical solution is obtained, but also to prevent an overly detailed input which may needlessly consume large amounts of computation time. In the following sections some basic guidelines for the preparation of geometrical inputs for each aircraft component considered are discussed. It must be remembered that these are only guidelines; the user must apply his own engineering judgment and experience to each particular configuration to be analyzed.

4.1 Philosophy of Program Use

It is fundamentally important that the user understand the capabilities and the basic limitations and approximations of each analytical method that is part of the STAMP package. These are discussed in detail in Volume I of this report and are reviewed in Section 3.0 of this volume. A program of the magnitude and complexity of STAMP requires the user not only to understand the actual data input requirements, but also the user must carefully analyze the problem before the computer solution is attempted, applying his engineering knowledge and judgment to the preparation of the computer input data. Only then can the full potential of the program be realized.

Proper interpretation of the output data from STAMP is an equally important requirement for the effective use of the program. The user must be aware of the meaning of each aerodynamic coefficient calculated by the program and must be aware of those effects not included in the program analysis which may have an influence on the final result. Volume III of this report is intended to provide the user with a rational approach to be used in evaluating and modifying the computer data.

The point of this discussion is to emphasize that STAMP cannot be treated as a "black box" type program which converts an airplane design configuration into a complete set of aerodynamic coefficients. This can be summarized by a trite phrase that should be applied to all computer programs: GARBAGE IN → GARBAGE OUT.

4.2 Geometry Preparation

The following sections are intended to provide the user with a set of guidelines for preparing the detailed geometry input of the jet-wing, vectored jets, fuselage, and empennage. Included in this discussion are spacing considerations based on methods studies. Reference is made to Section 5.0, Input Instruction, which the user may choose to read concurrently.

4.2.1 Wing and Jet Sheet

The EVD jet-wing lifting surface program segment of STAMP is capable of evaluating the aerodynamic characteristics of jet-wings of arbitrary wing planform and arbitrary spanwise distribution of jet momentum. The jet-wing may be considered to be symmetric, asymmetric, or antisymmetric. A symmetric jet-wing is defined as one for which there is complete geometric symmetry about the x-z plane, including symmetry of

a. Planform geometry
b. Incidence of each element
c. Jet spanwise location
d. Jet momentum at each spanwise station
e. Jet deflection

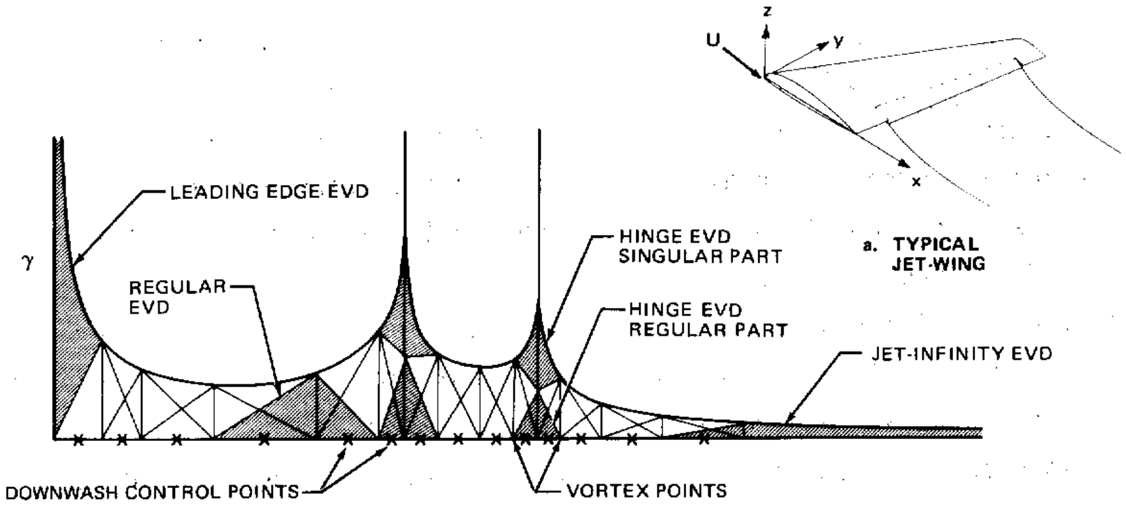
An anti-symmetric configuration has symmetry for items a, c, and d above; but anti-symmetry [i.e., $f(y) = - f(-y)$] for items b and e. Angle of attack becomes a meaningless quantity for an anti-symmetric jet-wing, and the program has been structured accordingly. An asymmetric configuration is defined as one having a non-symmetry in any of the above listed parameters.

The analysis of a jet-wing configuration begins with the definition of the planform geometry of the wing and jet sheet. In all jet flap cases the jet sheet extends downstream to infinity. Two methods are provided to allow the rapid input of the wing planform geometry (Section 5.2). The first is for the general case and can be used to define any planform shape that can reasonably be approximated by straight line segments. The second method is for wings of symmetric trapezoidal planform and requires only the input of aspect ratio, taper ratio, and quarter-chord sweep. In either case, however, the wing planform geometry will be approximated by a series of rectangular elements (figure 4.1c), resulting in discontinuous leading and trailing edges, except for rectangular planforms. Highly swept leading and/or trailing edges may require more spanwise divisions in order to provide a reasonable planform geometry approximation.

Following planform definition, the wing and jet are divided up into spanwise sections of arbitrary width. The arrangement of these sections is a matter of user judgment and is mainly determined by considering two factors. First is the planform geometry, which may have discontinuities such as flap edges, flap extensions, leading and trailing edge breaks, jet sheet edges, etc. The root of swept wings must also be considered as a planform discontinuity, since the leading and trailing edges have breaks at the root. This restriction does not apply for rectangular wings. Spanwise section edges should be positioned to coincide with these planform discontinuities. Second is anticipated gradients in spanwise loading (i.e., dc_ℓ/dy). Since the program treats the spanwise loading as constant over each section, there should be close spanwise spacing in regions where rapid loading variation is anticipated. For example, near the wing tip or near spanwise edges of a flap or a jet, there will normally be a rapid change in the spanwise loading. Therefore, the spanwise spacing in these regions should be narrow so that the "stairstep" loading assumed in the EVD method can adequately represent the actual spanwise loading distribution. Guidelines have been established, through experience and the element spacing study of Section 2.2.5 of Volume I, for the spanwise spacing of sections. It is recommended that at the spanwise station of a discontinuity in loading the section width be limited to 2-4 percent of the wing semispan, increasing steadily to no more than 15 percent for spanwise sections in regions of nearly constant loading. These guidelines should provide a reasonable stepwise representation of the spanwise loading while keeping the total number of divisions to a usable number.

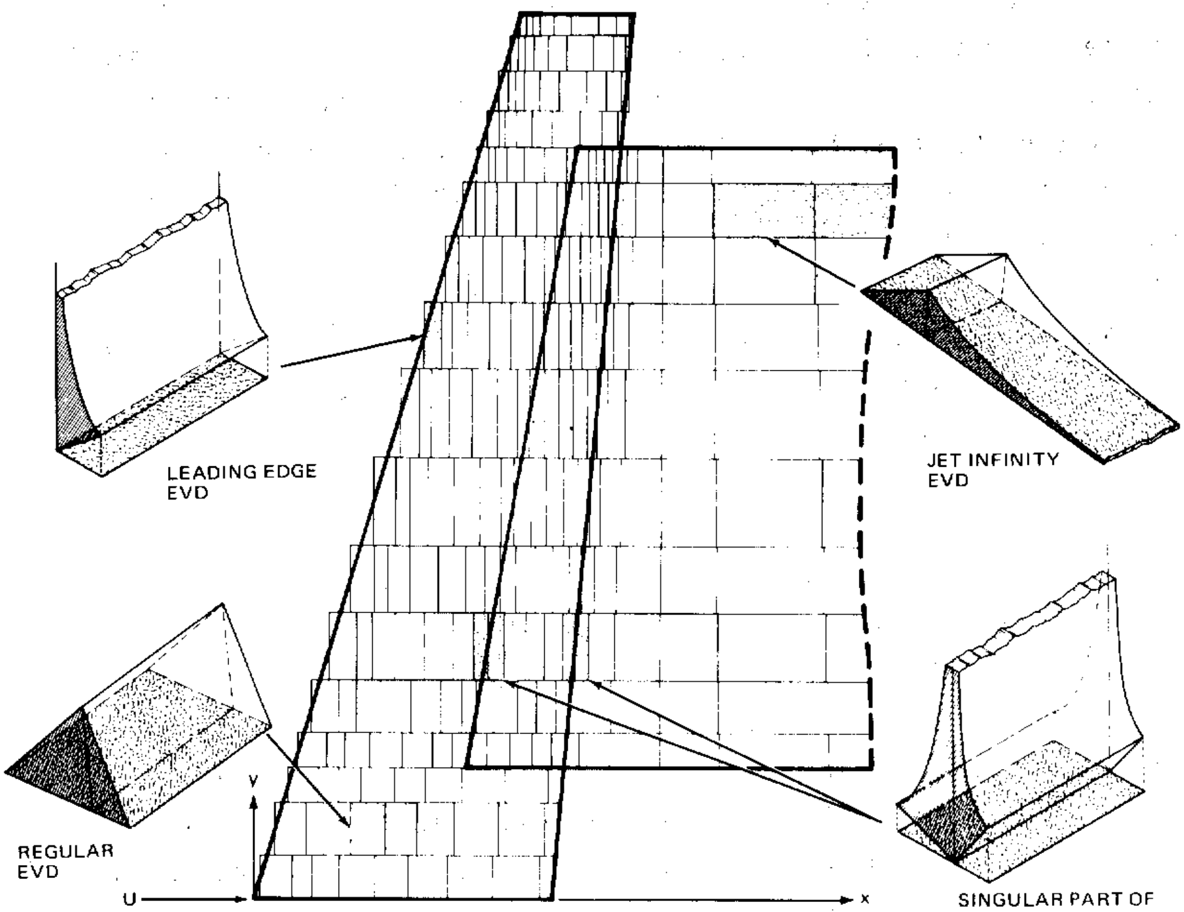
Once the spanwise division of the jet-wing planform is complete, each spanwise section must be divided in the chordwise sense so that the complete planform is ultimately represented by an array of rectangular elements suitable for representation by an array of Elementary Vortex Distributions. This representation is illustrated in figure (4.1) for a typical jet-wing planform. Note that no two elements overlap, nor are there any gaps. The chordwise arrangement of elements is also a matter of user judgment. A close uniform spacing could always yield a good, smooth chordwise loading distribution; but the linearly varying nature of the loading given by the Regular EVD element type allows a much coarser spacing to be used in regions where the loading is expected to vary linearly. In addition, special EVD elements have been developed in order to provide a more accurate representation of the chordwise loading in the special regions which are either difficult to approximate by linearly varying EVD's or which would require a large number of Regular EVD's. These regions are at the leading edge of a section (Leading Edge EVD), at the hinge line of a flap (Hinge EVD), at the leading edge of a jet deflected relative to the wing trailing edge (Hinge EVD), and on the last jet chordwise element which must extend downstream to infinity (Jet Infinity EVD). Use of these special EVD's in conjunction with the Regular EVD's makes chordwise spacing less critical and reduces the required number of elements on each section. The user need not specify the type of EVD to use, since the program internally specifies the type for each element. However, the special Hinge EVD is an option which the user may choose not to use since it has been shown to be of significantly less importance than the other special EVD's. The four EVD element types are discussed in detail in Section 2.2.2 of Volume I. Section camber must also be taken into consideration when selecting the chordwise spacing distribution. In regions where large variations in camber are expected to cause large variations in chordwise loading, the element spacing must be finer.

It is more difficult to establish a set of guidelines for chordwise spacing than for spanwise spacing because the chordwise loading distribution cannot always be simply deduced for complex three-dimensional flows. However, some basic spacing criteria will be outlined, but the judgment and experience of the user must be applied to each particular section type encountered. Although the loading gradient $\frac{d\Delta c_p}{dx}$ is quite large on the leading edge



a. TYPICAL JET-WING

b. REPRESENTATION OF TYPICAL SECTIONAL VORTEX DISTRIBUTION



c. TYPICAL USE OF EVD ELEMENTS

Figure (4.1). Program Representation of a Jet-Wing element of a section, it is recommended that this element be 5-10 percent of the local wing chord since the leading edge EVD element is an inverse square root singularity and hence will account for the gradient itself. A smaller leading edge element may be required if there is a large amount of camber at the leading edge, for example, if there is a leading edge device deflected. The second element on a section should actually be smaller than the first, say 2-4 percent, since it is a Regular EVD type and must be able to represent the transition between the leading edge singularity and the more linear variation aft of the leading edge. Subsequent element sizes can increase steadily to be as large as 20 percent of the local wing chord, but then must be decreased in size in a similar manner as the next large chordwise loading gradient is approached. It is suggested that elements of 2-5 percent be used on either side of a hinge line, although a larger element may be acceptable if the hinge singularity option is specified. Elements at the trailing edge of the wing on unblown sections must be reasonably small, say 2-5 percent, since the loading must go to zero at the trailing edge and hence the gradient may be large. Trailing edge elements on blown sections will, in general, follow the same criteria as for elements adjacent to a hinge line. The spacing of jet elements is somewhat simpler because the loading on the jet always decreases from its value at the trailing edge to zero at infinity. It will usually suffice to make the first jet element 2-4 percent of the local wing chord and increase element size rapidly for the remaining elements. The jet need be specified for only 2 or 3 local chords downstream before the infinity element is used, and the last regular jet elements can be as large as 1-2 chords.

The chordwise spacing guidelines presented should be sufficient for typical STOL transport wing sections. The user should always keep in mind the necessity to restrict the total number of elements on the wing and jet to the maximum number allowed by the program (400), but it is always desirable to use the minimum number possible which will yield a good loading distribution so that computation time and cost can be minimized. With experience the user will quickly gain familiarity with spacing criteria for the most efficient program operation.

Since the sectional geometric characteristics of a wing are approximated by a mean camber line in the EVD method, it is important that the user be able to translate a typical wing section into a suitable camber distribution. Representation of several camber lines for typical wing sections are illustrated in figure (4.2). All of the sections originate with a camber line for the clean configuration (i.e., no devices deflected) which is then modified to account for the deflections of leading and/or trailing edge device deflections. Again, engineering judgment must be used to determine the best representation of the mean camber line for complicated sections such as those in figure (4.2). It is suggested that the clean configuration camber line be considered separately from the changes produced by deflections, and that the cambers be combined by using the fundamental/composite case capability of the program, as discussed in Section 4.3. Ideally, the basic camber distribution would be fixed, and simple variations in camber due to deflections would be superimposed on it. However, deflections are often not representable as simple deflections in the camber line because of such effects as Fowler flap motions and changes in slot size, so the user must include some allowance for these effects in the basic camber distribution. Thus, the camber lines actually treated in the program may be approximations to the real camber lines. Rigorous treatment of such details is beyond the scope of the linearized approach.

4.2.2 Vectored Jets

The vectored jet flow field segment of STAMP computes the envelope shape (centerline location and cross sectional shape) of isolated circular jets issuing into the freestream at an arbitrary angle. Once the jet envelope shape is known, the perturbation velocities induced at any point in the flow field can be computed.

The jet envelope is represented by a series of adjacent segments of elliptical cross section and varying length, Δs [see figure (4.3)]. The lengths of these segments are controlled by the constant interval $\Delta z'$, which is input, and the local jet angle θ, which is computed.

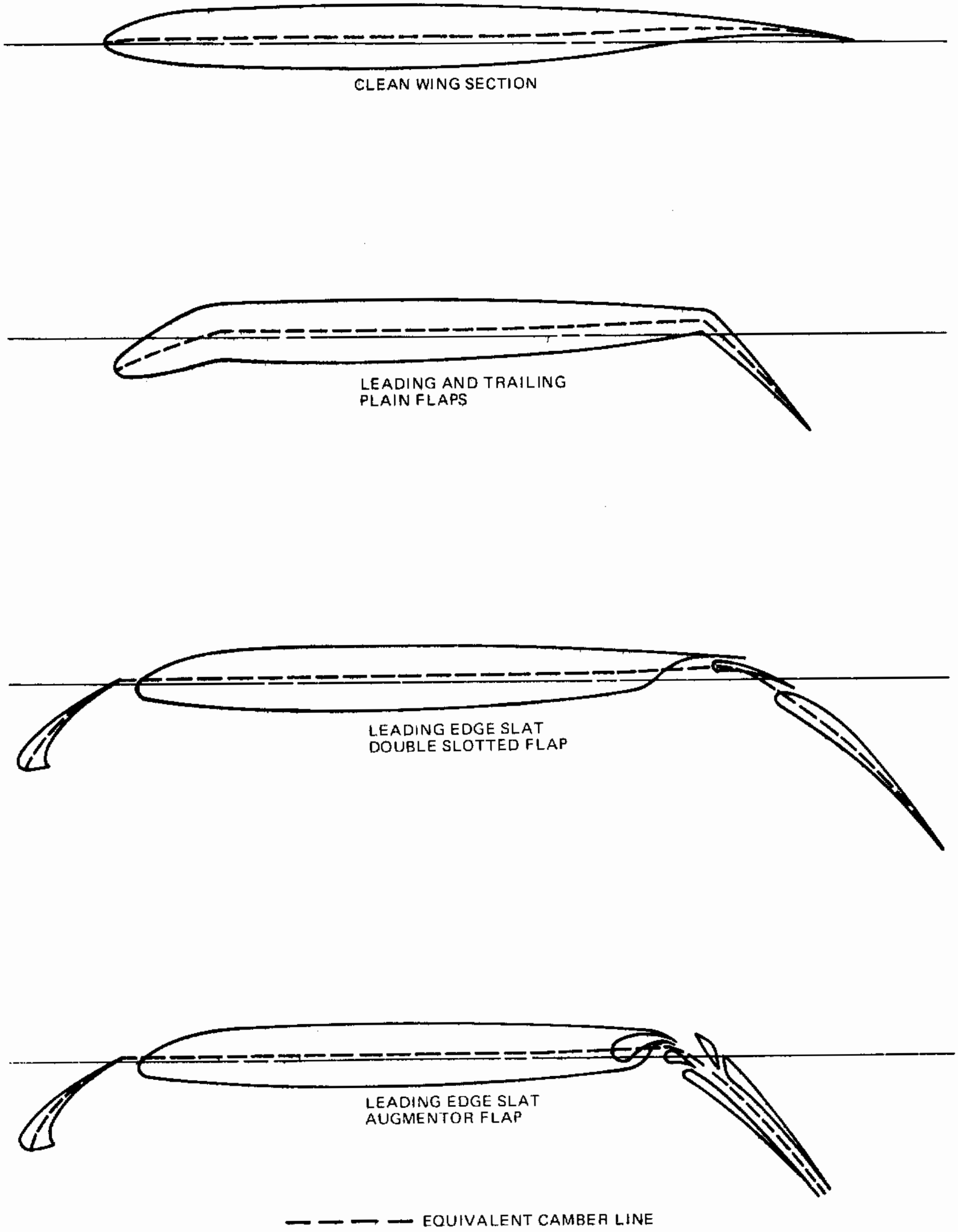
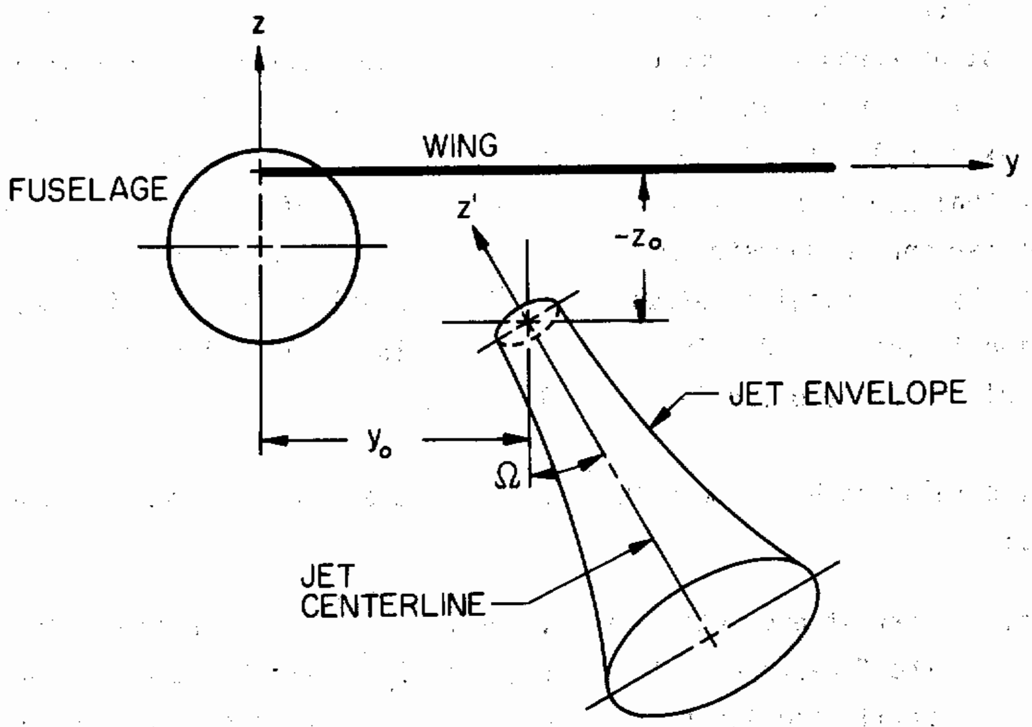
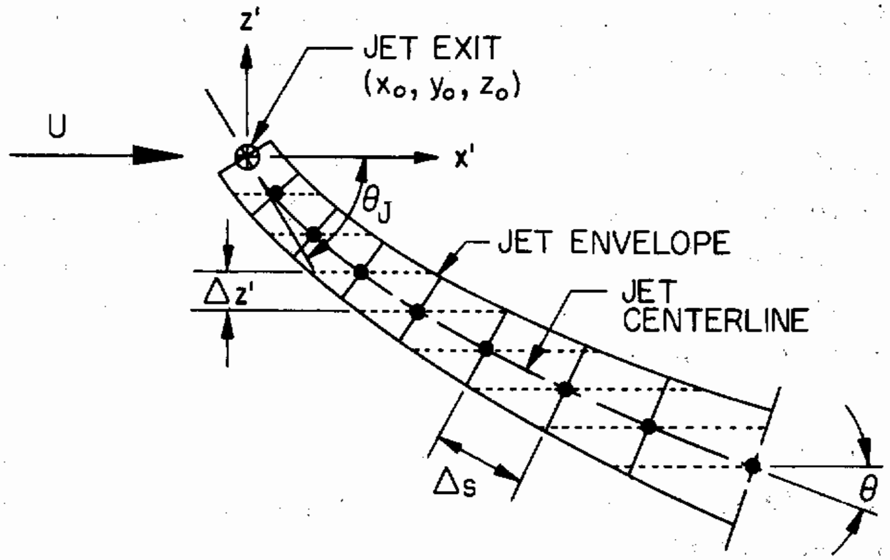


Figure (4.2). Representation of Wing Camber by Equivalent Camber.



(a) View Looking Forward (Along x and x' axes)



(b) View in the Plane of Jet Flow (i.e., x'-z' plane)

Figure (4.3). Vectored Jet Geometry

In order to achieve accurate flow field interferences, the jet envelope shape must be adequately represented by a limited number of segments (maximum of 200). The controlling input parameters, jet strength (CJ), exit diameter (DDO), exit angle (THTA), and the normalized $\Delta z'/D_O$ spacing size (DELZ) must be "balanced" so as to produce a jet envelope of accurate shape which extends far enough downstream from the aircraft so that truncation of further segments has a negligible effect. The jet centerline location is somewhat sensitive to the $\Delta z'/D_O$ interval size, and intervals which are too coarse may result in a distorted jet centerline shape.

The relationships among the input parameters may be summarized as follows:

a. Very strong jets (large value of $\frac{C_J S}{{D_O}^2}$) penetrate deeply into the freestream. Therefore, $\Delta z'/D_O$ must be large enough to permit the jet to turn to within one degree of the freestream within 200 segments.

b. Very weak jets (small $\frac{C_J S}{{D_O}^2}$) have only shallow penetration and turn very rapidly to align with the freestream. The $\Delta z'/D_O$ spacing must be fine enough to accurately represent the rapid turning rate and subsequent shallow penetration.

c. The greatest penetration is achieved by jets issuing nearly normal to the freestream (θ_J roughly 90°). Jets which issue more nearly parallel to the freestream (either forward or backward) will have more shallow penetration, and thus, will require finer $\Delta z'/D_O$ spacing.

Parametric studies have been made to investigate the relationships among the input parameters, the resulting number of jet segments, and the convergence of the jet centerline location. The expressions

$$\frac{\Delta z'}{D_O} = K(\theta_J)\left(\frac{C_J S}{{D_O}^2}\right)^{1/2} \qquad (4.1)$$

where $K(\theta_J) = -.00012\ \theta_J + .021$ for $|\theta_J| < 90°$ (4.2)

and $K(\theta_J) = +.00012\ \theta_J + .021$ for $|\theta_J| > 90°$ (4.3)

have been found to satisfy the required balance among the input parameters for approximately 100 to 150 jet segments. Satisfactory convergence of the jet centerline locations have also resulted for all but extreme jet angles (θ_J near 0° or 180°). Equation (4.1) is shown graphically in figure (4.4).

Equation (4.1) has been made the default option in the program when the input value of DELZ is given as 0.0. It is suggested that this default always be used unless the jet issues very nearly parallel to the freestream (say, $|\theta_J| < 5°$, or $|\theta_J| > 160°$). To insure a satisfactory solution, finer $\Delta z'/D_0$ spacing should be used for such critical jets, but spacing which is too fine will result in premature jet truncation because of exceeding 200 segments.

Truncation of the jet automatically occurs after the segment at which the local jet angle has decayed to one degree. This cutoff criterion, while arbitrary, insures that the jet extends many diameters downstream from the exit, when used in conjunction with the default $\Delta z'/D_0$ spacing size. This is shown in figure (4.5).

The detailed print option (IPRINT = -1), discussed in Section 6.6, will produce output which will aid the user in dealing with jets which require special attention.

The jet dihedral angle has no effect on the envelope shape, but only on the jet orientation relative to the aircraft. After the jet envelope shape is established, the plane of jet flow is rotated through the dihedral angle about an axis (x') parallel to the x-axis and passing through the center of the jet exit disk.

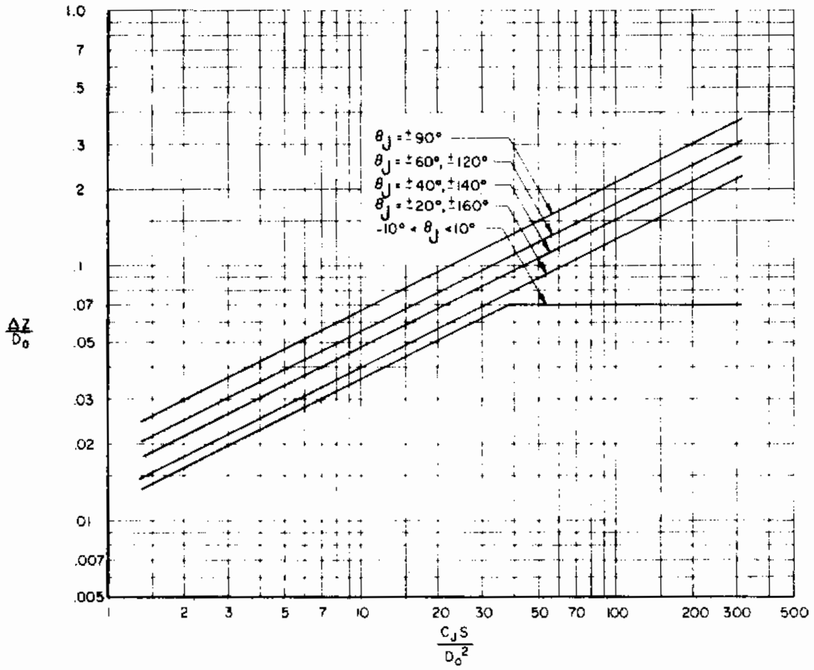


Figure (4.4). Default Segment Spacing Size for Vectored Jet Input

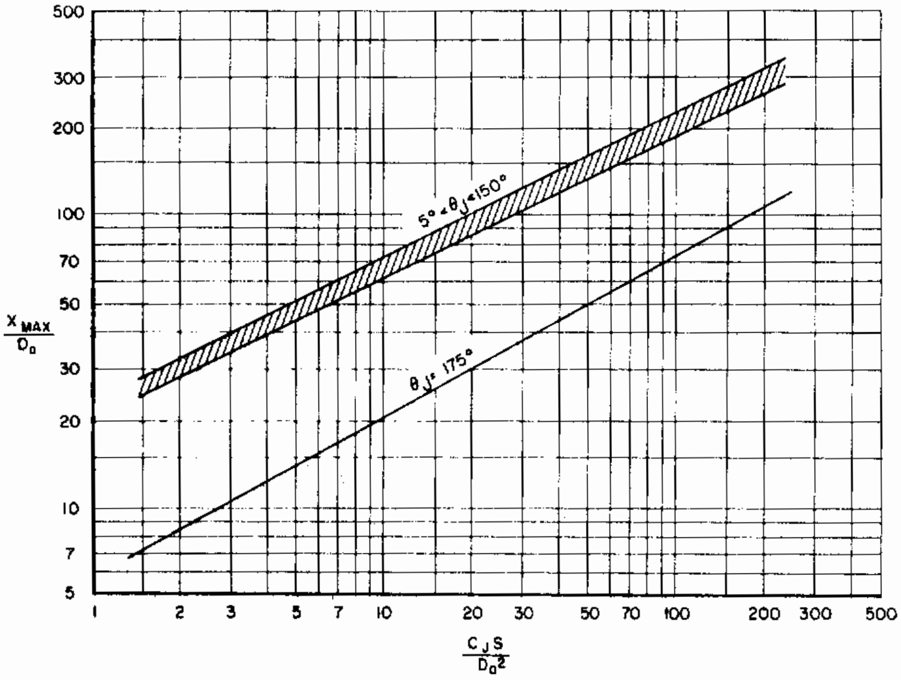


Figure (4.5). Maximum Streamwise Extent of Vectored Jet Envelope (Based on Cutoff at $\theta = 1$ Degree and Default Segment Spacing Size).

4.2.3 Fuselage

The fuselage analysis program segment of STAMP is capable a evaluating certain aerodynamic characteristics of fuselages of the types described in Section 3.4. The fuselage may be considered to be either symmetric or asymmetric with respect to the x-z plane. The geometry package included in the program has been designed to facilitate rapid fuselage geometry input by allowing the user to specify:

a. Circular body cross-sections
b. Elliptical body cross-sections
c. Arbitrary body cross-sections
d. Body camber

Options a or b along with d allows a complete fuselage to be input with the user knowing only the radius of circular sections or the semi-minor and semi-major axes of elliptical sections, plus the shape of the body axes, while option c requires that the user actually divide the fuselage surface into a large number of trapezoidal elements and input the coordinates element-by-element. It may be desirable to make certain approximations to the fuselage geometry which will allow a more rapid geometry input.

Irrespective of mode of input, user judgment must be exercised in dividing the fuselage surface into streamwise segments and circumferential divisions. A set of guidelines is provided, which has been established through experience and spacing studies, to aid the user in choosing an adequate set of fuselage divisions. In a manner similar to the EVD method, it is necessary to space fuselage elements closely in regions of large body curvature where significant loading gradients are to be expected, and to space elements more coarsely over the relatively flat portions of the fuselage.

First, the streamwise segments should be formed, starting at the leading edge of the body. For the slender spheroids (ellipsoids of revolution) considered in spacing studies it has been found that the leading edge segment should be less than one percent of the body length, increasing steadily to no more than 5-7 percent until the maximum diameter section is reached, and then the segment size should be decreased in a similar manner as the aft end of the body is approached. For more complicated shapes, the segments must be kept small in the vicinity of irregularities in the body shape.

Following division of the body into streamwise segments, circumferential divisions must be defined for each segment. Again the general guidelines to follow is to space elements closely in the vicinity of anticipated large loading gradients. For the circular and elliptical section inputs, the user need only choose the number of circumferential elements on a section. The program internally chooses the element distribution based on a cosine spacing algorithm. It is recommended that 15-20 elements be specified for the half-circumference. For arbitrary section inputs user judgment is required. It is likely that additional elements would be required for irregularly shaped segments.

In addition to body geometry criteria, the interference effects of the jet-wing on the fuselage should be considered for element spacing. Since the gradient of the jet-wing induced downwash, $\frac{dw}{dx}$, is the greatest in the vicinity of the wing-body intersection, it may be desirable to reduce the size of the streamwise elements in that region in order to better represent the effects of interference. Finally, the fuselage analysis method treats body camber in the same sense as induced downwash, so in the regions of rapid change in body camber a finer streamwise spacing may be desirable.

4.2.4 Empennage

The empennage analysis program segment of STAMP is capable of analyzing the aerodynamic and stability and control characteristics of an arbitrary horizontal/vertical tail combination in the flow field of a jet-wing high lift system. The empennage may be a conventional tail, a T-tail, or a mid-tail arrangement.

The general guidelines for element spacing on the tail surfaces are basically the same as those adopted in the EVD method with certain qualification. Because the vortex distribution is discrete rather than continuous, large variations in the chordwise loading can only be represented by using a large number of vortex lattice elements. This is particularly true at the leading edge of a section or at a flap hinge line because the discrete singularity method cannot use the special loading functions as can EVD. However, it is probably not as critical to obtain an accurate chordwise load distribution on the tail as it is on the wing, except possibly in determining such quantities as elevator power or rudder power (see Volume III).

Spanwise section spacing on both tail surfaces should be reduced in the vincinity of the juncture between the horizontal tail and the vertical tail, particularly for mid-tail and T-tail type arrangements.

4.3 Fundamental and Composite Cases

The concepts of the fundamental case and the composite case have been discussed in Sections 3.1.2 and 3.1.3, respectively, as well as in Volume I. This section is intended to provide the user with some insight into the effective implementation of these concepts.

Since the loading due to the deflection of a portion of the jet-wing surface can be treated independently of the loading due to other deflections in the EVD method, it is convenient to divide the total surface deflection of a particular configuration into its fundamental parts, treating each part by itself, and then combine the fundamental loadings in such a manner as to produce the composite loading of the complete configuration. The EVD segment of STAMP is structured such that associated with each EVD element is a set of fundamental incidence angles. Each of these incidence angles is the resultant angle due to the deflection pattern of one fundamental case. For example, figure (4.6) shows a complex jet-wing configuration which is divided into nine of these fundamental cases. In most problems, the user will find it best to consider the deflection of a particular component (e.g., a flap, a slat, the jet, etc.) as a separate fundamental case. It is usually convenient to combine all the invarient deflections (e.g., camber and twist) into one fundamental case. Then a composite case can be formed by modulating the influence of any of the fundamental cases and summing the result, as

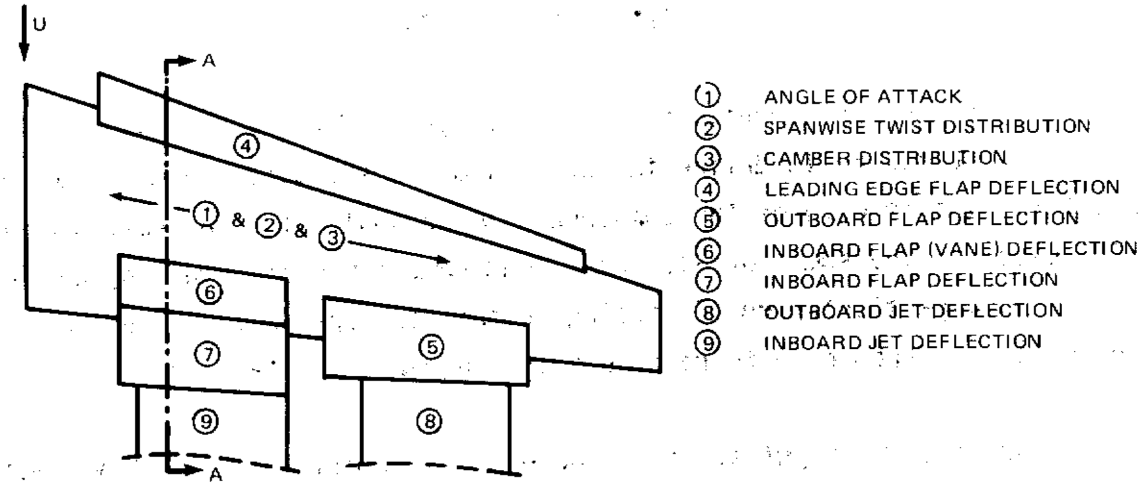
$$\varepsilon(x,y) = \sum_{m=1}^{K} a_m \, \varepsilon_m(x,y) \qquad (4.3)$$

Figure (4.6b) shows the sectional characteristics of Section AA for various fundamental cases. It should be noted that the incidence of all the elements outside the range of deflection is zero. Thus, when the deflection is varied, these elements will have no change in incidence. However, since the deflection of even one EVD element produces a loading distribution on every element of the jet-wing system, variation in the deflection of a fundamental case will produce a corresponding linear variation in the loading of every EVD element.

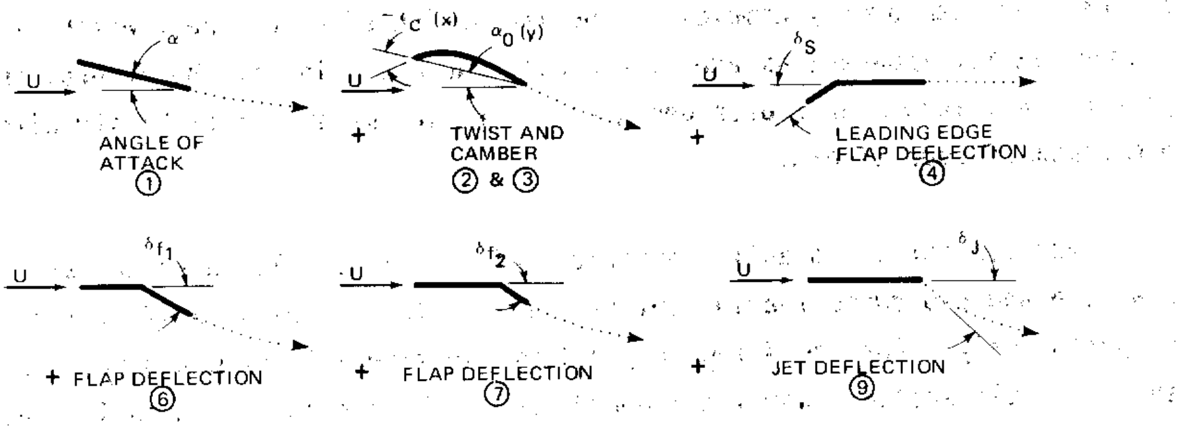
The user should realize that the inherent advantage of the fundamental/composite case capability is that a large number of composite cases can be evaluated using the same set of fundamental case data with only a minimal increase in computation time. Therefore, the user should carefully plan his computer runs so that the fundamental case solutions can be exploited to their full potential.

The composite case assumes a more fundamental nature for ground effect runs, as discussed in Section 3.1.4, because the aerodynamic characteristics cannot be linearly combined. Although the program inputs are the same for ground effect as for free air cases, only the composite cases are printed by the program for ground effect runs.

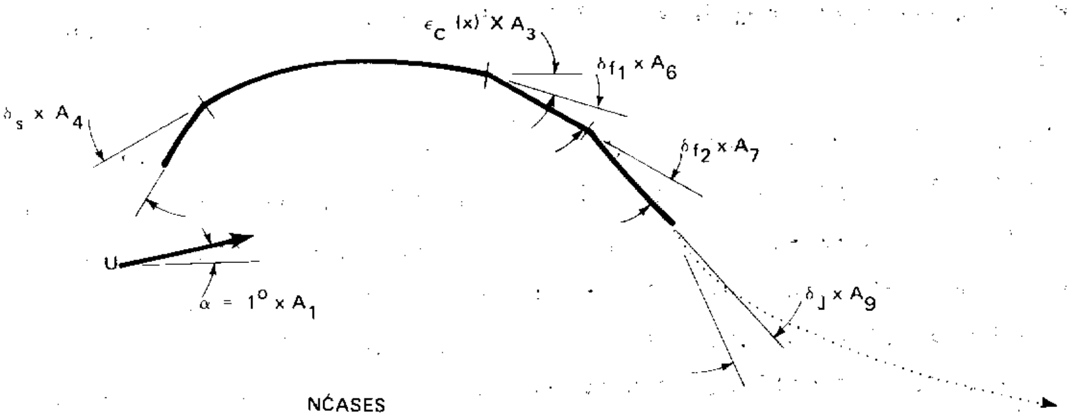
The other major segments of STAMP cannot utilize the fundamental case nature of the EVD method because of the non-linear interferences induced by the jet-wing flow field. For example, the empennage and fuselage segments of the program are executed for each jet-wing composite case since the interferences due to the jet-wing vary non-linearly with jet-wing fundamental case deflections.



a. ARRANGEMENT OF FUNDAMENTAL CASES FOR A COMPLEX JET WING



b. SEPARATE FUNDAMENTAL CASES OF A TYPICAL WING SECTION (SECTION A-A)



$$\text{COMPOSITE CASE DEFLECTION} = \sum_{i=1}^{\text{NCASES}} \text{FUNDAMENTAL CASE FACTOR } A_i \times \left(\text{FUNDAMENTAL CASE DEFLECTION}\right)_i$$

$$\text{COMPOSITE CASE LOADING} = \sum_{i=1}^{\text{NCASES}} \text{FUNDAMENTAL CASE FACTOR } A_i \times \left(\text{FUNDAMENTAL CASE LOADING}\right)_i$$

c. FORMATION OF A TYPICAL COMPOSITE CASE (SECTION A-A)

Figure (4.6). Treatment of Fundamental and Composite Cases

5.0 INPUT INSTRUCTIONS

The program input consists of various types of cards containing all the information necessary to define the program options desired and the geometric characteristics of each of the components to be analyzed. The card sequence listed below is the same for all cases, but some types of cards may not be required in a particular run depending on the user options. The number of cards will vary from case to case, but the sequence is always the same.

Throughout the input stage of the program, various schemes have been devised to eliminate the need for repetitious data. While some of these schemes may seem overly complex to the inexperienced user, their worth will become readily apparent as experience with the operation of the program is gained. All inputs have been designed to provide maximum flexibility with a minimum of user effort.

For symmetric and anti-symmetric configurations, all input and all spanwise and chordwise loading output data are for the right half of the component only. For the jet-wing planform, sectional input would begin at the right tip section, working inboard section by section. Similarly, only the right half of fuselage meridians need be input for symmetric cases. Further symmetry considerations will be discussed throughout the input description.

Figure (5.1) shows the reference coordinate system used for both input and output data. This is a wind axis system with the freestream direction aligned with the x-axis. The origin of the coordinate system is coincident with the wing apex, the z-axis is positive upwards, and the y-axis forms the right hand coordinate system. It must be noted that if a wing is to be analyzed in the yawed position, it must be input as a non-symmetric planform, rotated about the wing apex (z-axis). All wing sections will still be aligned with the x-axis, including the tip sections. The printed output will also be referenced to the wind axis system, and the user must make suitable transformations if coefficients are required in the body axis system.

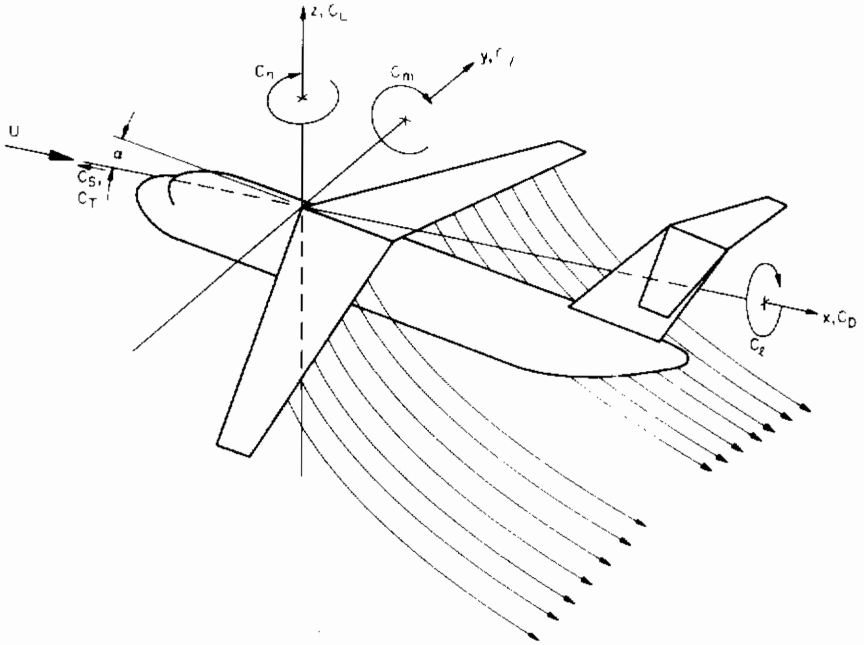


Figure (5.1a). Reference Coordinate System for Aerodynamic Coefficients.

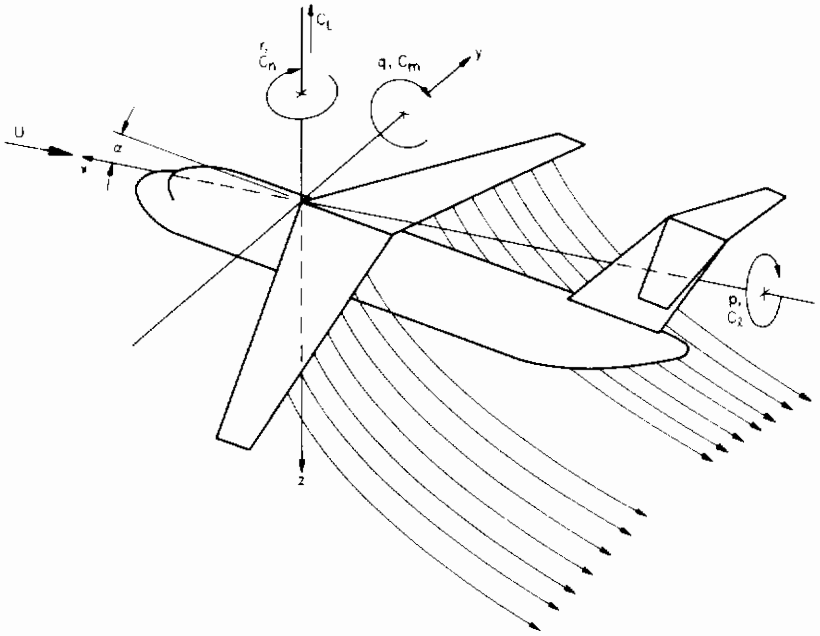


Figure (5.1b). Reference Coordinate System for Dynamic Stability Derivatives.

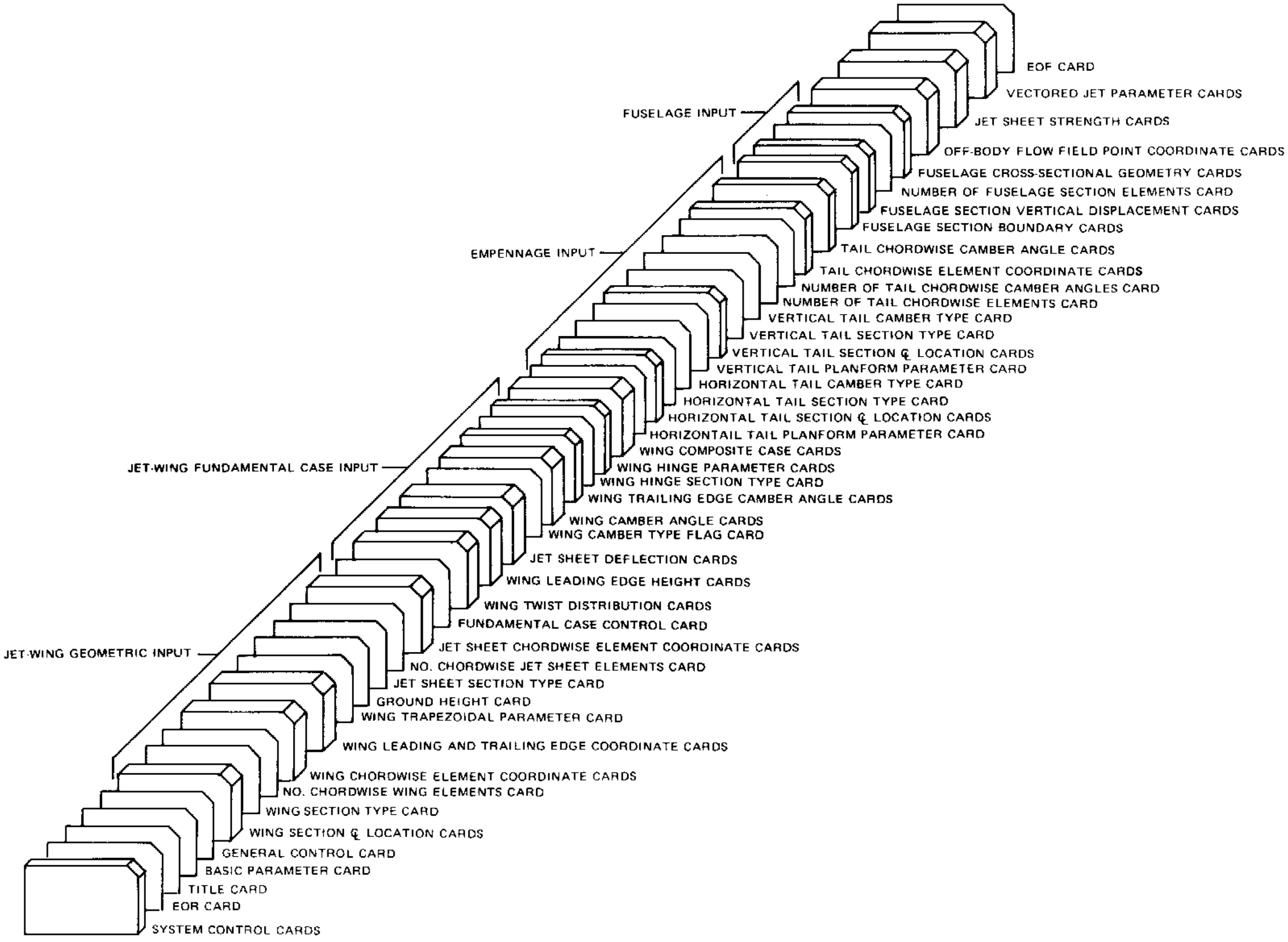


Figure (5.2). Arrangement of STAMP Input Card Deck.

In the input description which follows, all input values are real floating point numbers (F format), unless otherwise specified. Floating point numbers are classified as dimensional or dimensionless, those being dimensional having the same scale for all aircraft components. All variables specified as integers must be right justified in their fields.

5.1 Basic Parameters and Control Flags

The cards in this section must always be input.

- Title Card - This card provides any desired description of the run. The title will be printed at the top of the first page of output.

Column	Name	Explanation
1-80	TITLE	Run title (any acceptable machine characters).

- Basic Parameter Card - This card contains basic aircraft parameters.

Column	Name	Explanation
1-10	AREA	Wing area, in units of (SPAN)^2 to be used for normalization of aerodynamic coefficients.
11-20	SPAN	Wing span, in any desired units.
21-30	CREF	Wing reference chord, to be used for normalizing various aerodynamic coefficients. Units of SPAN. If a value of 0.0 is input, the mean aerodynamic chord will be used.
31-40	XMC	Point about which pitching moments will be taken, measured from the wing apex (origin). Units of SPAN.
41-50	XCG	Wing center of gravity location, measured from wing apex (origin), which will be used as pitching axis for computation of stability derivatives due to pitching. This parameter must be input only if IDERIV $\neq$ 0. Units of SPAN.
51-60	TCR	Average thickness-chord ratio of the wing. Used only if the jet-wing flow field segment of the program is executed (NPOINT $>$ 0, NROWH $>$ 0, NROWV $>$ 0, or NSEG $>$ 0).
61-70	SSLIP	Sideslip angle of the fuselage, in degrees. Required for fuselage runs only.

- General Control Card - This card contains control flags which describe the basic characteristics of the run.

Column	Name	Explanation
1-2	NROWS	Number of spanwise section (rows) into which the wing is divided. For symmetric or anti-symmetric wings, only the number of sections on the right half should be input. If there is no wing, NROWS = 0. Otherwise, $3 \le \text{NROWS} \le 40$. Integer.
3-4	NCASES	Number of wing fundamental cases. It must be noted that the angle of attack case is always set up automatically, so that the number of cases for which data will be input must be one less than NCASES ($1 \le \text{NCASES} \le 10$). Integer. Not required if NROWS = 0. (Note: There is no angle of attack case for anti-symmetric runs.)
6	ISYMM	Symmetry indicator flag (symmetry about the x-axis). Integer. = 0 configuration is symmetric > 0 configuration is non-symmetric < 0 configuration is anti-symmetric
7-8	IPRINT	Print output control flag. Integer. = 2 print geometry details and total aerodynamic coefficients = 1 in addition, print spanwise loading = 0 in addition, print chordwise loading = -1 special print option. Not intended for general use.
10	JETFLG	Jet indicator flag. Integer. = 0 there is a jet sheet. = 1 there is no jet sheet or vectored jets, and no jet inputs will be read. = 2 there are vectored jets
12	IGTYPE	Wing planform geometry indicator flag. Integer. = 1 wing planform is completely arbitrary, and sectional leading and trailing edge coordinates will be read to define planform = 2 wing planform is trapezoidal, and simplified planform input will be read. Not required if NROWS=0.

Column	Name	Explanation
14	IHINGE	Hinge EVD indicator flag. Integer. = 0 regular EVD will be used on all hinge elements. > 0 logarithmically singular Hinge EVD will be used on all hinge elements. This option is not permitted for use in computing stability derivatives or for ground effect runs (IDERIV > 0 or IGRND > 0). Not required if NROWS = 0.
16	IDERIV	Dynamic stability derivative control flag. Integer. = 0 a normal run will be executed, with no stability derivatives computed > 0 a normal run will be executed, and in addition, a stability derivative run will be executed. This option requires approximately double the normal computing time.
18	IGRND	Ground effect flag. Integer. = 0 a normal run will be executed with no ground effect solution > 0 a normal run will be executed, and in addition, the ground effect solution will be executed for up to eight ground heights. Each ground height will require additional computing time equal to approximately one-half of the normal run computing time. Not required if NROWS = 0.
19-20	IFFLD	Jet-wing flow field function/punch flag. Integer. > 0 jet-wing loading data will be punched on cards for later use in the remainder of the program. NROWS must not be zero. Use of the punched output can save considerable computer time if new flow field runs are to be made. = 0 no punched output. < 0 jet-wing loading data will be read in for use in the remainder of the program, but the EVD program will not be run. All normal jet-wing geometry inputs are required. Vectored jets cannot be used in this option.

Column	Name	Explanation
22-24	NPOINT	Flow field point flag. Indicates the number of (x,y,z) coordinates for which off-body flow field is required. Integer. (NPOINT ≤ 200). Input 0 is no flow field solutions are required. Integer.
25-26	NROWH	Number of spanwise sections (rows) into which the horizontal tail is divided. For symmetric cases input only the number of sections on the right panel. However, if there is also a vertical tail, both the left and right panels of the horizontal must be input. If there is no horizontal tail, NROWH = 0. Integer.
28	ICAMH	Horizontal tail camber flag. Integer. = 0 no camber > 0 camber or elevator deflection, and inputs for camber will be read.
29-30	NROWV	Number of spanwise sections (rows) into which the vertical tail is divided. There are no symmetry considerations for the vertical tail. If there is no vertical tail, NROWV = 0. Integer.
32	ICAMV	Vertical tail camber flag. Integer. = 0 no camber > 0 camber or rudder deflection, and inputs for camber will be read.
33-34	NSEG	Number of streamwise segments into which the fuselage is divided. If there is no fuselage, NSEG = 0. Maximum of 40 segments. Integer.
36	ISECT	Fuselage section input type. Integer. = 1 arbitrary fuselage segments will be input. = 2 circular cross-section fuselage segments will be input. = 3 elliptical cross-section fuselage segments will be input.
38	ICAMF	Fuselage camber flag. Integer. = 0 no fuselage camber (i.e., upsweep) > 0 camber of the fuselage centerline, and inputs will be read.

5.2 Jet-Wing Geometry Input

This section describes the jet-wing geometry input. If NROWS = 0, this section should be skipped.

- Section Centerline Location Cards - These cards contain the spanwise locations of the centerline of each wing (and jet) section. Eight values per card, maximum of 5 cards (40 sections) allowed.

Columns	Name	Explanation
1-10, 11-20 etc.	Y	Spanwise distance from wing centerline (x-axis) to section centerline, normalized by SPAN/2. All values must be (-1.0 < Y < 1.0). NROWS values must be input, beginning at the right wing tip and working to: (a) wing centerline, for symmetric or anti-symmetric wings (b) left wing tip, for non-symmetric wings.

- Wing Section Type Card - This card indicates the chordwise arrangement of EVD elements for each section on the wing.

Columns	Name	Explanation
1-2, 3-4	ICTYPE	Type number of each wing section. The arrangement of chordwise EVD elements in a row (both number of elements and the x/c location of each) determines the wing row type. Any two sections which have the same number and x/c location of all EVD elements, have the same ICTYPE value. NROWS values must be input. There may not be more than 10 different values (i.e., section types). The highest value input is called NWTYPE, and all values less than NWTYPE must be used, that is, no "gaps" are allowed in the sequence 1 through NWTYPE. Integer.

- Number of Chordwise Wing Elements - This card contains the number of chordwise EVD elements for each wing section type.

Columns	Name	Explanation
1-2, 3-4, etc.	NI	Number of chordwise EVD elements of each wing section type. There must be NWTYPE values ($1 \leq NWTYPE \leq 10$) in the sequence for section type 1, 2, 3, . . . (i.e., ascending order of row types). At least 2 but not more than 20 chordwise EVD elements must be input on each wing section. ($2 \leq NI \leq 20$). Integer.

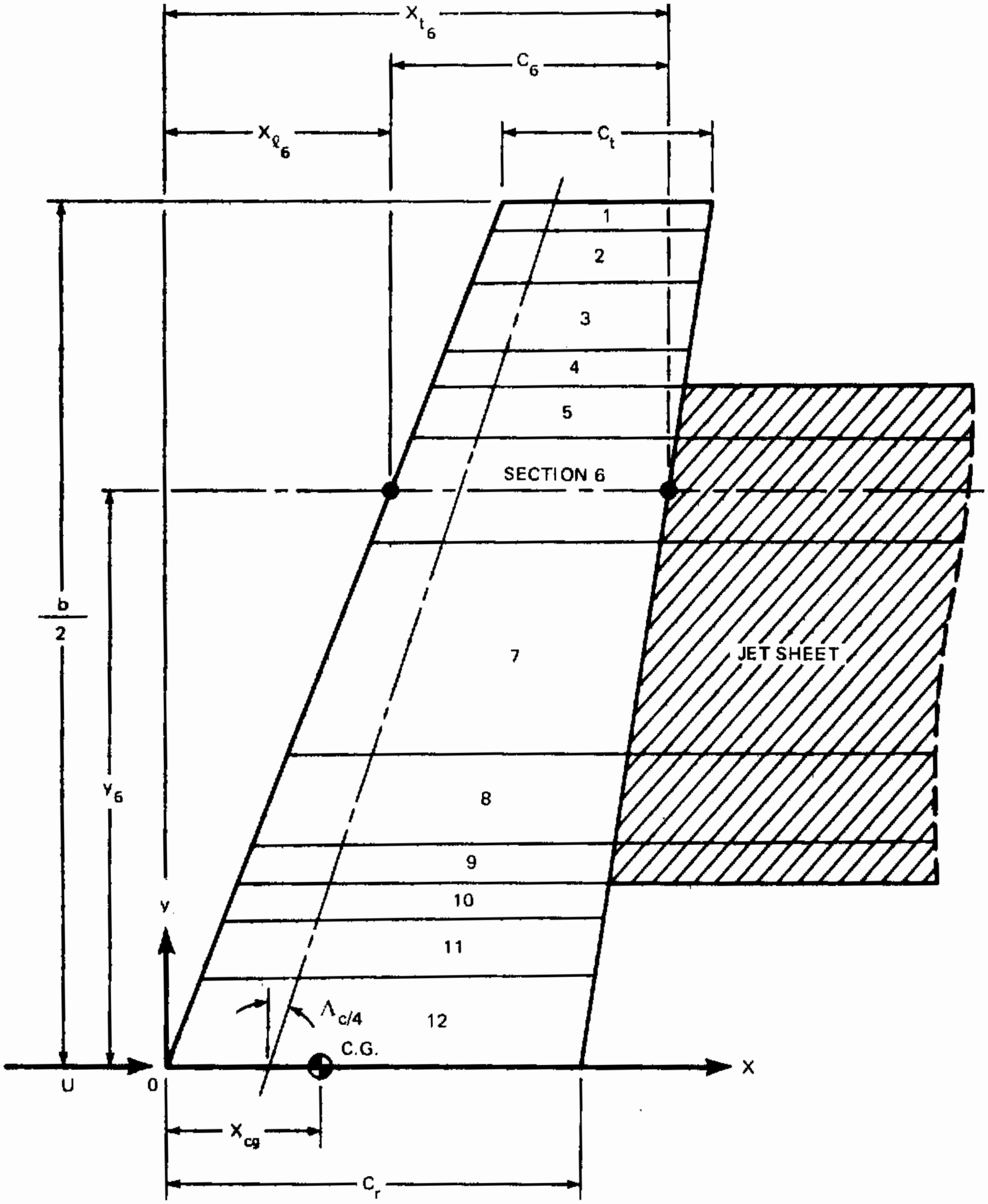


Figure (5.3). Program Planform Dimensions and Notation

- Wing Chordwise Element Coordinates - These cards contain the x/c coordinates of each EVD element of each section type. NWTYPE sets of cards required, each with NI values of x/c. Maximum of 10 sets, 3 cards per set.

Columns	Name	Explanation
1-10, 11-20, etc.	XB	The chordwise coordinate of each EVD vortex point, measured from the sectional leading edge and normalized by the sectional chord. The vortex point is defined as the leading edge of all rectangular elements. The first value of each set must be 0.0 (leading edge) and the last value must be less than 1.0. A maximum of 20 values (i.e., 20 EVD elements) is permitted for each set, but the total number of elements on the wing and jet combined must not exceed 400.

- Leading and Trailing Edge Coordinates - These cards contain the coordinates of the leading and trailing edges of the given sections. Only stations on each side of breaks in the leading or trailing edges need be input. The leading and trailing edges are interpolated for sections not given by straight lines between those sections which are given. This method of planform definition is used only if IGTYPE = 1. Number of cards required is ≥ 2 and ≤ NROWS.

Column	Name	Explanation
1-10	Y	Spanwise distance from a section centerline to the centerline of the wing (x-axis), normalized by the half span. Each value must be the same as one of those already input on the Section Centerline Location Cards. Values on the right wing half are positive, and values on the left wing half (nonsymmetric only) are negative.
11-20	XLEAD	Chordwise distance from section leading edge (at section centerline) to the wing apex (y-axis). Units of SPAN.
21-30	XTRAIL	Chordwise distance from section trailing edge (at section centerline) to the wing apex (y-axis). Units of SPAN.

Column	Name	Explanation
1	9	A 9 must appear in CC1 of the next card after all the above coordinate cards to signal that all desired sections have been input. This card is required only if IGTYPE = 1. (Input 9999 for CDC systems.)

- Trapezoidal Wing Parameters - This card contains planform information for simple trapezoidal wings. It replaces the above coordinate cards when IGTYPE = 2.

Column	Name	Explanation
1-10	ARATIO	Wing aspect ratio, ($SPAN^2/AREA$).
11-20	SWEEP	Sweep angle of wing quarter-chord line, in degrees.
21-30	TR	Wing taper ratio, ($CHORD_{WING\ TIP}/CHORD_{WING\ CENTERLINE}$).

- Ground Height - This card contains the height of the wing above the ground. Required only if IGRND > 0. Up to eight values may be input.

Columns	Name	Explanation
1-10 11-20, etc.	GHITE	Height of wing above ground in units of SPAN.

- Jet Row Type Card - This card indicates the chordwise arrangement of EVD elements for each section on the jet sheet. Required only if JETFLG = 0.

Columns	Name	Explanation
1-2, 3-4, etc.	IJTYPE	The type number of each jet section. This variable is similar to the ICTYPE variable, except that for sections with no jet a value of 0 must be input. The number of different jet section types is called NJTYPE. The number of nonzero values input is NROWSJ, the number of sections having a jet. Integer.

- Number of Chordwise Jet Elements - This card contains the number of chordwise EVD elements for each jet section type. Required only if JETFLG = 0.

Columns	Name	Explanation
1-2, 3-4, etc.	NI	Number of chordwise EVD elements of each jet section type. This variable is similar to NI for wing sections above, except that there must be NJTYPE sections. At least 2 but not more than 10 chordwise EVD elements must be input on each wing section ($2 \leq NI \leq 10$). Integer.

- Jet Chordwise Element Coordinates - These cards contain the x/c coordinates of each element of each jet section type. NJTYPE sets of cards required, each with NI values of x/c. Maximum of 10 sets, 2 cards per set. Required only if JETFLG = 0.

Columns	Name	Explanation
1-10, 11-20, etc.	XB	The chordwise coordinate of each EVD vortex point, measured from the sectional leading edge (at centerline) and normalized by the sectional chord. The first value of each set must be 1.0 (trailing edge) and a maximum of 10 values is permitted for each set ($NI \leq 10$).

- Fundamental Case Cards - These cards describe the type of fundamental case and the geometric details of each case for the (NCASES - 1) input cases. For each case, a fundamental case control card must be input immediately followed by the appropriate sets of geometric parameter cards.

- Fundamental Case Control Card - This card identifies the types of linear geometric variations to be included in each fundamental case. In each of the flags below, a zero value indicates omission of the respective type of input for that fundamental case. A non-zero value indicates the variation will be included and input must be given to define it.

Column	Name	Explanation
2	INTWST	Spanwise twist distribution flag. Integer.
4	INHITE	Leading edge vertical displacement flag. Integer.
6	INDELJ	Jet deflection flag. Integer.

Column	Name	Explanation
8	INCAMB	Camber flag. Integer.
10	INBETA	Wing hinge deflection flag. Integer.

- Twist Distribution Card - These cards contain the spanwise distribution of wing twist. NROWS values required, eight per card. Required only if INTWST ≠ 0.

Columns	Name	Explanation
1-10, 11-20, etc.	TWIST	Sectional wing twist, in degrees, at section centerline, with respect to the wing reference plane. Positive is in the same sense as a positive angle of attack.

- Leading Edge Height Card - These cards contain the displacement of the wing leading edge from the wing reference plane. NROWS values required, eight per card. Required only if INHITE ≠ 0.

Columns	Name	Explanation
1-10, 11-20, etc.	HO	Displacement coordinates of sectional leading edge from the wing reference plane, normalized by the sectional chord. These values are used only for computation of the moment arm of the jet reaction thrust contribution to pitching moments. Leading edge displacement may be the result of dihedral, twist, nonlinear movement of a leading edge device, etc. Translation resulting from ordinary linear leading and trailing flap deflections are accounted for automatically by the program.

- Jet Deflection Card - These cards contain the spanwise variation of jet deflection, relative to the trailing edge. NROWSJ values are required, eight per card. Required only if INDELJ ≠ 0.

Columns	Name	Explanation
1-10, 11-20, etc.	DJ	Jet turning angle, in degrees, trailing edge (i.e., relative to the mean line at the trailing edge). Positive deflection is downward.

- Camber Type Flag Card - This card indicates the chordwise distribution of camber for each section on the wing. Required only if INCAMB ≠ 0.

Columns	Name	Explanation
1-2, 3-4, etc.	ICT	Camber type number of each wing section. The arrangement of camber, including all positions (x/c) and angles must be the same in order for two sections to have the same value of ICT. NROWS values must be input, but a maximum of 10 different types is allowed. The highest value input is called NCT, and there may be no "gaps" in the numbering sequence between 0 and NCT. A zero value indicates no camber. Integer.

- Camber Angle Cards - These cards contain the camber angles for each camber section type. NCT sets of cards required, three cards per set maximum. Required only if INCAMB ≠ 0.

Columns	Name	Explanation
1-10, 11-20, etc.	AC	The camber angle, in degrees, at the downwash control point of each EVD element (i.e., the point arbitrarily chosen as halfway between any two adjacent XB points). Positive values are in the same sense as positive angles of attack.

- Trailing Edge Camber Angle Card - These cards contain the trailing edge angle due to camber only. NROWSJ values are required. Required only if both INCAMB ≠ 0 and JETFLG = 0

Columns	Name	Explanation
1-10, 11-20, etc.	ACTE	Trailing edge angle, in degrees, due to camber only. These values are required only for sections which have a trailing jet, and are used only in computing the total jet deflection angle with respect to the freestream.

- Hinge Section Type Card - This card identifies the arrangement of leading and trailing flap hinges on each section. NROWS values are required. Required only if INBETA ≠ 0.

Columns	Name	Explanation
1-2, 3-4, etc.	IHT	Hinge section type flag. If there are no hinges, a value of 0 should be input. Sections with the same hinge section type must be alike in number of hinges, their location (x/c) and their type (leading or trailing flap). There may be as many as NROWS different section types. The number of different types is called NHT, and there may be no gaps in the sequence 0 through NHT. Integer.

- Hinge Parameter Cards - These cards contain the location, type (leading or trailing flap), and turning angle of each hinge on a given type of hinge section. NHT cards required. Each section may have a maximum of four hinges. Required only if INBETA ≠ 0.

Columns	Name	Explanation
1-10, 21-30, etc.	XBH	Distance from the sectional leading edge to the hinge point, normalized by the sectional chord. This value must be the same as one of the XB coordinates input for the section. There should always be at least one regular element forward and one regular element aft of the elements adjacent to a hinge point.
11,31, etc.	ILT	Hinge type identification flag. Integer. = 0 trailing flap hinge (positive deflection in the sense of positive angle of attack). ≠ 0 leading flap hinge (positive deflection in the sense of negative angle of attack).
12-20, 32-40, etc.	BETA	Hinge deflection angle, in degrees, of the element behind the hinge point, relative to the element before the hinge point.

- Composite Case Cards - These cards specify the desired superposition of the linear fundamental cases. A maximum of 24 composite cases may be requested. No composite cases may also be specified (9 card alone required).

Columns	Name	Explanation
1-2, 9-10, etc.	N	Fundamental case number which is to be included in forming a given composite case. Integer.
3-8, 11-16, etc.	A	Multiplication factor to be applied to the fundamental case identified in the previous two card columns.
1	9	A 9 must appear in CC1 of the card following the composite case cards, to signal the end of composite case input. (Required even if there are no composite case cards input). (Input 9999 for CDC systems.)

5.3 Empennage Geometry Input

This section describes horizontal tail and vertical tail geometry input. If both NROWH = 0 and NROWV = 0, this section should be skipped.

5.3.1 Horizontal Tail - Skip this section if NROWH = 0.

- Horizontal Tail Planform Card - This card defines the basic planform parameters for the horizontal tail.

Column	Name	Explanation
1-20	ARH	Horizontal tail aspect ratio ($SPANH^2/AREAH$).
11-20	TRH	Horizontal tail taper ratio.
21-30	SWH	Horizontal tail quarter chord sweep (degrees).
31-40	SPANH	Span of the horizontal tail. Same units as wing span.
41-50	HL	Horizontal tail length. Streamwise distance from the apex of the wing to the apex of the horizontal tail. Same units as wing span.
51-60	HH	Horizontal tail height. Vertical distance from wing plane to horizontal tail plane. Same units as wing span.

- Horizontal Tail Section Centerline Cards - These cards contain the spanwise locations of the centerline of each horizontal tail section. Eight values per card, maximum of 3 cards (20 sections).

Columns	Name	Explanation
1-10, 11-20, etc.	YH	Spanwise distance from tail centerline, normalized by SPANH/2. All values must be ($-1.0 < YH < 1.0$). NROWH values must be input, beginning at the right tip section and working to: (a) Left tip section if there is also a vertical tail (NROWV > 0) or for asymmetric runs (ISYMM > 0). (b) Center section for symmetric or antisymmetric runs (ISYMM ≤ 0) with no vertical tail.

- Horizontal Tail Section Type Card - This card indicates the chordwise arrangement of elements for each section of the horizontal tail.

Columns	Name	Explanation
1-2, 3-4, etc.	IHX	Type number of each horizontal tail section. The chordwise arrangement of elements in a row determine the tail row type. Any two sections which have the same number and x/c location of elements can have the same IHX value. NROWH values must be input, but no more than 10 different values (section types) may be used. The highest value is called NXTYPE and all values less than NXTYPE must be used. Integer. Total number of horizontal tail elements ≤ 80.

- Horizontal Tail Camber Type Card - This card indicates the chordwise arrangement of camber for each section of the horizontal tail. Omit this card if ICAMH = 0.

Columns	Name	Explanation
1-2, 3-4, etc.	IHC	Camber type number of each horizontal tail section. The chordwise arrangement of camber in a row determines the tail row type. Any two sections with the same number of elements and same camber distribution can have the same IHC value. NROWH values must be input, but no more than 10 different values may be used. The highest value is called NCTYPE and all values less than NCTYPE must be used. Integer.

5.3.2 Vertical Tail - Skip this section if NROWV = 0.

- Vertical Tail Planform Card - This card defines basic planform parameters for the vertical tail.

Column	Name	Explanation
1-10	ARV	Vertical tail aspect ratio (SPANV^2/AREAV).
11-20	TRV	Vertical tail taper ratio.
21-30	SWV	Vertical tail quarter chord sweep (degrees).
31-40	SPANV	Span of the vertical tail. Same units as wing span.
41-50	VL	Vertical tail length. Streamwise distance from the apex of the wing to the apex of the vertical tail. Same units as wing span.
51-60	VH	Vertical tail height. Vertical distance from the wing plane to the root of the vertical tail.

- Vertical Tail Section Centerline Cards - These cards contain the spanwise locations of the centerline of each vertical tail section. Eight values per card, maximum of 2 cards (10 sections).

Columns	Name	Explanation
1-10, 11-20, etc.	ZV	Spanwise distance from vertical tail root chord, normalized by SPANV/2. All values must be ($0.0 < ZV < 1.0$). NROWV values must be input, beginning at the uppermost section and working to the lowermost section.

- Vertical Tail Section Type Card - This card indicates the chordwise arrangement of elements for each section of the vertical tail.

Columns	Name	Explanation
1-2, 3-4, etc.	IVX	Type number of each vertical tail section. Essentially the same as IHX for the horizontal tail. However, section types are shared with the horizontal tail. Thus, if a row on the horizontal has the same number and x/c location of elements as a row on the vertical, the rows can have the same type number. Total number of section types for the complete empennage is $NXTYPE \leq 10$. Total number of vertical tail elements ≤ 20.

- Vertical Tail Camber Type Card - This card indicates the chordwise arrangement of camber for each section on the vertical tail. Omit this card if ICAMV = 0.

Columns	Name	Explanation
1-2, 3-4, etc.	IVC	Camber type number of each vertical tail section. Essentially the same as IHC for the horizontal tail. Section type are shared analogous to the x/c section types. Total number of camber types for the complete empennage is $NCTYPE \leq 10$.

5.3.3 Common Empennage Input

This section describes inputs common to both the horizontal tail and vertical tail. Omit this section if NROWH = 0 and NROWV = 0.

- Number of Chordwise Tail Elements - This card contains the number of chordwise elements for each tail section type.

Columns	Name	Explanation
1-2, 3-4, etc.	NX	Number of chordwise elements for each tail section type. There must be NXTYPE values input in the sequence of section types 1, 2, 3, . . . NXTYPE. ($1 \le NX \le 10$). Integer.

- Number of Chordwise Camber Angles - This card contains the number of chordwise camber angles for each camber section type. Omit this card if ICAMH = 0 and ICAMV = 0.

Columns	Name	Explanation
1-2, 3-4, etc.	NC	Number of chordwise camber angles for each tail camber type. There must be NCTYPE values input in the sequence of section types 1, 2, 3, . . . NCTYPE. ($1 \le NC \le 10$). Integer.

Note: Each tail section has a section type number and, for camber cases, a camber type number. The section type number (IHX or IVX) and camber type number (IHC or IVC) for a section need not be the same, but the number of chordwise elements [NX(IHX) or NX(IVX)] for the section must be the same as the number of camber angles [NC(IHC) or NC(IVC)] for that section.

- Tail Chordwise Element Coordinates - These cards contain the x/c coordinates of each element for each section type. NXTYPE sets of cards are required, each with NX values of x/c. Maximum of 10 sets of cards, 2 cards per set.

Columns	Name	Explanation
1-10, 11-20, etc.	XBT	The chordwise coordinate of the leading edge of each tail element, measured from the section leading edge and normalized by the section chord. The first value of each set must be 0.0 and the last value must be less than 1.0.

- Tail Camber Angles - These cards contain the camber angles of each element for each section type. NCTYPE sets of cards are required, each with NC values of camber angles. Maximum of 10 sets of cards, 2 cards per set. Omit these cards if ICAMH = 0 and ICAMV = 0.

Columns	Name	Explanation
1-10, 11-20, etc.	CAMBER	The camber angle, in degrees, of each tail element at the element normal-wash control point (3/4-point of the element). A positive camber angle on the horizontal tail is in the same sense as a positive angle of attack and a positive camber angle on the vertical tail is in the same sense as positive rudder deflection (trailing edge left) or negative sideslip.

5.4 Fuselage Geometry Input

This section describes fuselage geometry input. If NSEG = 0, skip this section.

- Fuselage Section Boundary Cards - These cards define the streamwise location of the fuselage section boundaries. Eight values per card, maximum of 6 cards (41 boundaries).

Columns	Name	Explanation
1-10, 11-20, etc.	XF	Streamwise distance from the wing apex to the fuselage section boundary. Same units as wing span. NSEG + 1 values required. Boundaries forward of the wing apex will have negative values of XF; boundaries aft of the wing apex will have positive values of XF.

- Fuselage Section Vertical Displacement - These cards define the vertical displacement of the centroid of the fuselage section boundaries. Eight values per card, maximum of 6 cards (41 boundaries). Omit if ICAMF = 0.

Columns	Name	Explanation
1-10, 11-20, etc.	ZCL	Vertical distance from the wing plane to the centroid of each fuselage section boundary. Same units as wing span. Positive for centroids above the wing plane.

- Number of Fuselage Section Elements - This card contains the number of circumferential elements for each fuselage segment. NSEG values required.

Columns	Name	Explanation
1-2, 3-4, etc.	NF	Number of circumferential elements into which each fuselage segment is divided. For symmetric cases, only the number of elements of the right half of each section is required. (3 ≤ NF ≤ 20).

- **Fuselage Section Input for Arbitrary Cross-Sections** - These cards contain coordinates for each fuselage cross-section. Omit these cards if ISECT ≠ 1. For each fuselage section, a set of (Y,Z) coordinates must be input for both the forward boundary of the section and the aft boundary of the section. However, if the previous section has the same number of elements as the present section, input only the aft boundary coordinates. The program internally equates the forward boundary coordinates with the aft boundary coordinates of the previous section. Note that for the most forward and most aft boundaries, all the element coordinates must be input even though the local radius must be zero. See figure (5.4). A minimum of NSEG + 1 sets of cards are required and a maximum of 2(NSEG) sets of cards is allowed. Maximum of 6 cards per set. Start at bottom of section and proceed counterclockwise.

Columns	Name	Explanation
1-10, 11-20, etc.	Y	Value of the Y-coordinates for each element on the fuselage section boundary. NF + 1 values required. See figure (5.4).

Columns	Name	Explanation
1-10, 11-20, etc.	Z	Value of the Z-coordinates for each element on the fuselage section boundary. NF + 1 values required. See figure (5.4)

- **Fuselage Section Input for Circular Cross-Sections** - These cards contain the radius of each fuselage section boundary. Omit these cards if ISECT ≠ 2. NSEG + 1 values required. Maximum of 6 cards, 8 values per card (41 boundaries).

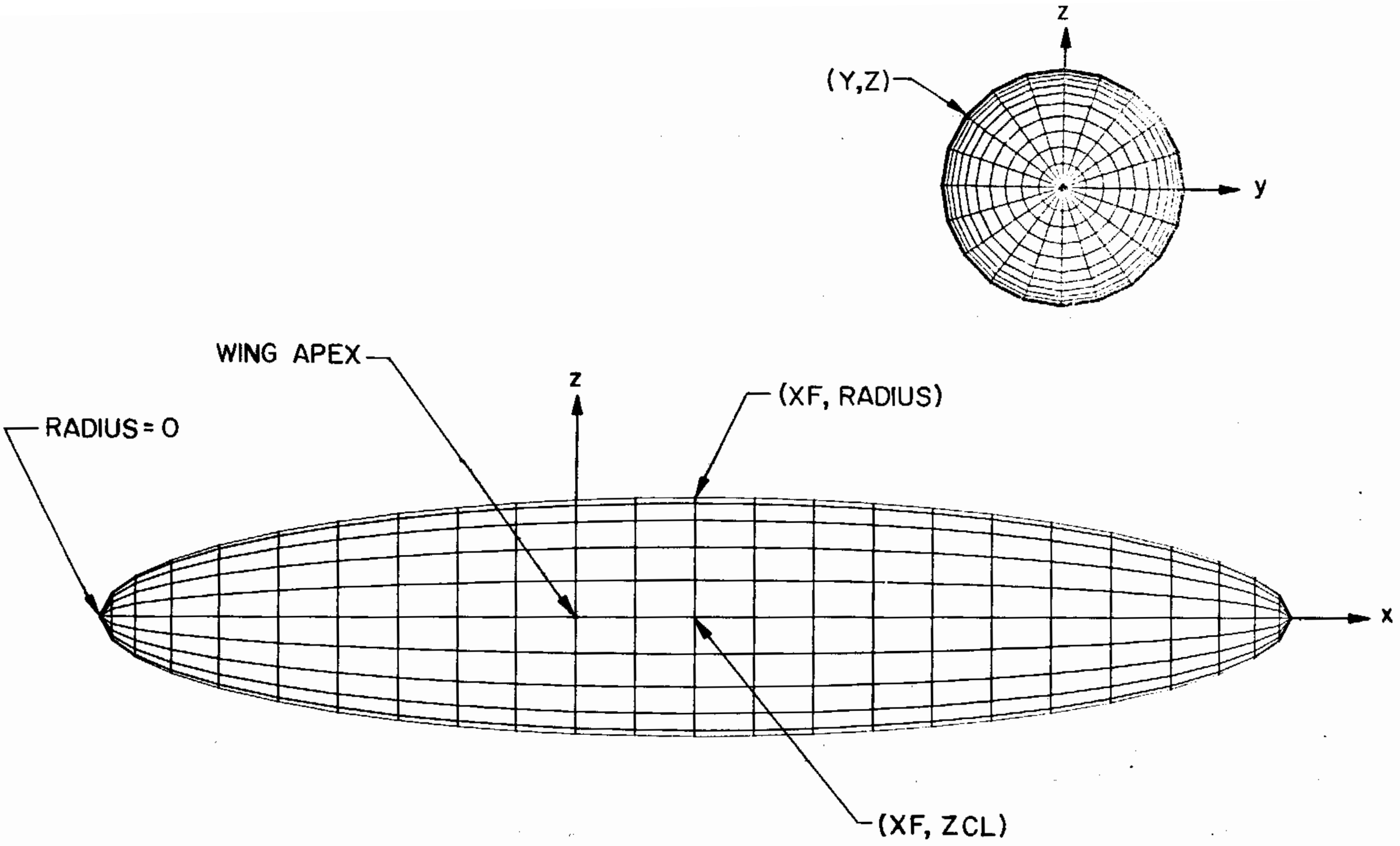


Figure (5.4). Program Representation of a Fuselage.

Columns	Name	Explanation
1-10, 11-20, etc.	RADIUS	Radius of the section boundary. Same units as wing span. The first and last values input must be 0.0 since the body is closed at both ends.

- Fuselage Section Input for Elliptical Cross-Sections - These cards contain the semi-minor axis and semi-major axis of each fuselage section boundary. Omit these cards if ISECT ≠ 3. NSEG + 1 values of each axis required. Maximum of 12 cards, 8 values per card (41 boundaries).

Columns	Name	Explanation
1-10, 11-20, etc.	AMIN	Semi-minor axis of the section boundary. Same units as wing span. The semi-minor axis always corresponds to the Y-axis. The first and last values input must be 0.0 since the body is closed at both ends. See figure (5.4).
1-10 11-20, etc.	AMAJ	Semi-major axis of the section boundary. Same units as wing span. The semi-major axis always corresponds to the Z-axis. The first and last values input must be 0.0 since the body is closed on both ends. See figure (5.4).

5.5 Off-Body Flow Field Input

This section describes the input necessary for the calculation of off-body flow field velocities. If NPOINT = 0, skip this section.

- Flow Field Point Coordinates - These cards specify the (X,Y,Z) for points at which the off-body flow field will be calculated. NPOINT cards required.

Column	Name	Explanation
1-10 11-20 21-30	X Y Z	Coordinates of the off-body points in the body axis system (i.e., relative to the wing apex). Same units as wing span.

5.6 Jet Input

This section describes the input necessary to describe either a jet sheet exiting from the wing trailing edge or vectored jets. Skip this section if JETFLG = 1.

5.6.1 Jet Sheet Strength Input - Skip this section if JETFLG ≠ 0.

- Jet Strength Cards - These cards contain the jet strength for all wing sections which have a jet. NROWSJ values are required. An unlimited number of sets of cards may be input, maximum of 5 cards per set.

Columns	Name	Explanation
1-10, 11-20, etc.	CMU	Sectional jet momentum strength for each jet row. CMU is defined as CMU = J/qc, where J is the sectional jet momentum flux (units of force) and q and c are the dynamic pressure and sectional chord, respectively. There may be no zero values input unless all values are zero (indicating no jet at all).
	9	A 9 must appear in CC1 of the card following all the CMU cards to signal the end of the case input. If no further cases are required, this card may be omitted. (Input 9999 for CDC systems.)

5.6.2 Vectored Jet - These cards specify the strength and geometric characteristics of each circular vectored jet. One card is required to describe each jet. A maximum of ten jets may be included in each jet configuration group. Any number of jet configuration groups may be input. These cards are required only if JETFLG = 2.

Column	Name	Explanation
1-10	CJ	Jet thrust coefficient per engine.

$$C_J = \rho_j A_j V_j^2 / qS$$

Column	Name	Explanation
11-20	DO	Jet exit disk diameter, normalized by SPAN/2.
21-30	THJ	Jet exit angle, in degrees, in the plane of jet flow, measured from the freestream direction at zero angle of attack. Positive deflection is downward, in the same sense as for a jet sheet deflection, DJ [see figure (4.3)]. Values from |1| to |179| degrees are acceptable.

Column	Name	Explanation
31-40	DELZ	Incremental step size (in the plane of jet flow) to be used in calculating jet envelope shape, normalized by DO/SPAN/2. The value input must be compatible with the jet strength, diameter, angle, and number of segments computed to describe the jet shape. Refer to figure (4.4)for correlation of these parameters. If 0.0 is input the spacing will automatically be selected.
41-50	XO	Coordinates of center of jet exit disk, each normalized by SPAN/2.
51-60	YO	
61-70	ZO	
71-77	DIHED	Jet dihedral angle in degrees, measured from the Y = constant plane. Angles for jets tending to issue toward the right wing tip are positive [see figure (4.3)].
78	ISY	Jet symmetry flag (may take on values different from wing symmetry flag ISYMM). Integer. Jets located in the Y = 0 plane (i.e., YO = DIHED = 0) are automatically considered non-symmetric. = 0 symmetric jet. An image jet (about the Y = 0 plane) will be computed in addition to the input jet. = 1 non-symmetric jet. Only the input jet will be considered and no image jet will be computed.
79	JETREP	Repeat jet characteristics flag. Integer. = 0 all characteristics of this jet are unique, and the jet path will be computed. = 1 the jet characteristics CJ, DO, THTA, DELZ are identical to those of the immediately preceeding jet. These input quantities will be redefined as those of the previous jet, but the remaining input quantities (XO,YO,ZO, DIHED) will be read and used for this jet as usual. The jet path need not be recomputed, but the shape of the previous jet will be used instead, resulting in a saving in computer time.

Column	Name	Explanation
80	LAST	Last jet flag. This flag indicates whether or not this is the last jet to be included in a given configuration group. Integer. = 0 this is not the last jet in this configuration group. Another card will be read. = 1 this is the last jet in this configuration group. No further cards will be read until a new jet configuration is considered.
1	9	A 9 must appear in CC1 of the card following all Vectored Jet Parameter Cards to signal that no more vectored jet configuration groups are required for this run. Required only if JETFLG = 2. (Input 9999 for CDC systems.)

5.7 Special Runs Using Previously Calculated Wing Loading Data

It is often desirable to save the jet-wing loading data generated by the EVD portion of the STAMP program so that future runs for tail, fuselage, or flow field data can be made without rerunning the EVD portion of the program. Jet-wing data is saved on punch cards if the user specifies IFFLD > 0. To reuse this data, merely specify IFFLD < 0, use the same jet-wing geometry data as in the run that generated the punched output, add any other components desired (e.g., tail, fuselage, off-body points), and add the loading data as follows: Remove all CMU cards from the original deck and replace them with the complete punched output deck. The ending 9 card will come after the punched output deck.

This rerun feature can be used for runs with no jets at all or with jet sheets, but not for vectored jet runs. Ground effect is allowed but dynamic stability derivatives are not. This feature will save considerable computer time since the EVD portion of the program is, in general, the most time consuming segment of the program.

6.0 OUTPUT DESCRIPTION

The following sections provide a summary description of the output produced by STAMP. These sections are structured in approximately the same order as the program output, but there may be some variance due to the multiple case nature of the program and the user chosen options. Figure (6.1) illustrates the most general output arrangement, including multiple case loops.

The program output consists of data written on the standard print unit 6, and in addition certain jet-wing loading data may be written on standard punch unit 7. Three print options, controlled by the value of the IPRINT flag described in Section 5.1, are available to the user. The normal printout is given for the default value of zero and includes chordwise and spanwise loading and total aerodynamic coefficients of the jet-wing, as well as a standard set of output for flow fields, empennage, and fuselage. The other print options may be used if the details of the spanwise and/or chordwise loading are not required, reducing the amount of print and saving a small amount computing time. It is strongly recommended that the full program output be requested, however, because the chordwise and spanwise loading data may be helpful in confirming the validity of the element spacing used and the savings in computer time realized by not printing these data is not significant.

In addition to the three print flag options described in Section 5.1, a program troubleshooting option (IPRINT < 0) is available for complete program functional checkout. A massive amount of data is printed, including jet-wing downwash matrices, augmented downwash matrices, vortex strength soltutions, matrix back substitutions, vectored jet envelope shapes, vectored jet-induced flow field velocities, empennage normalwash matrices, fuselage pressure coefficients, etc. This option is not recommended for normal program use.

The program output has been structured such that each output section is preceeded by a title surrounded by an asterisk box. In addition, a header is printed at the top of nearly every page of output, giving the date and

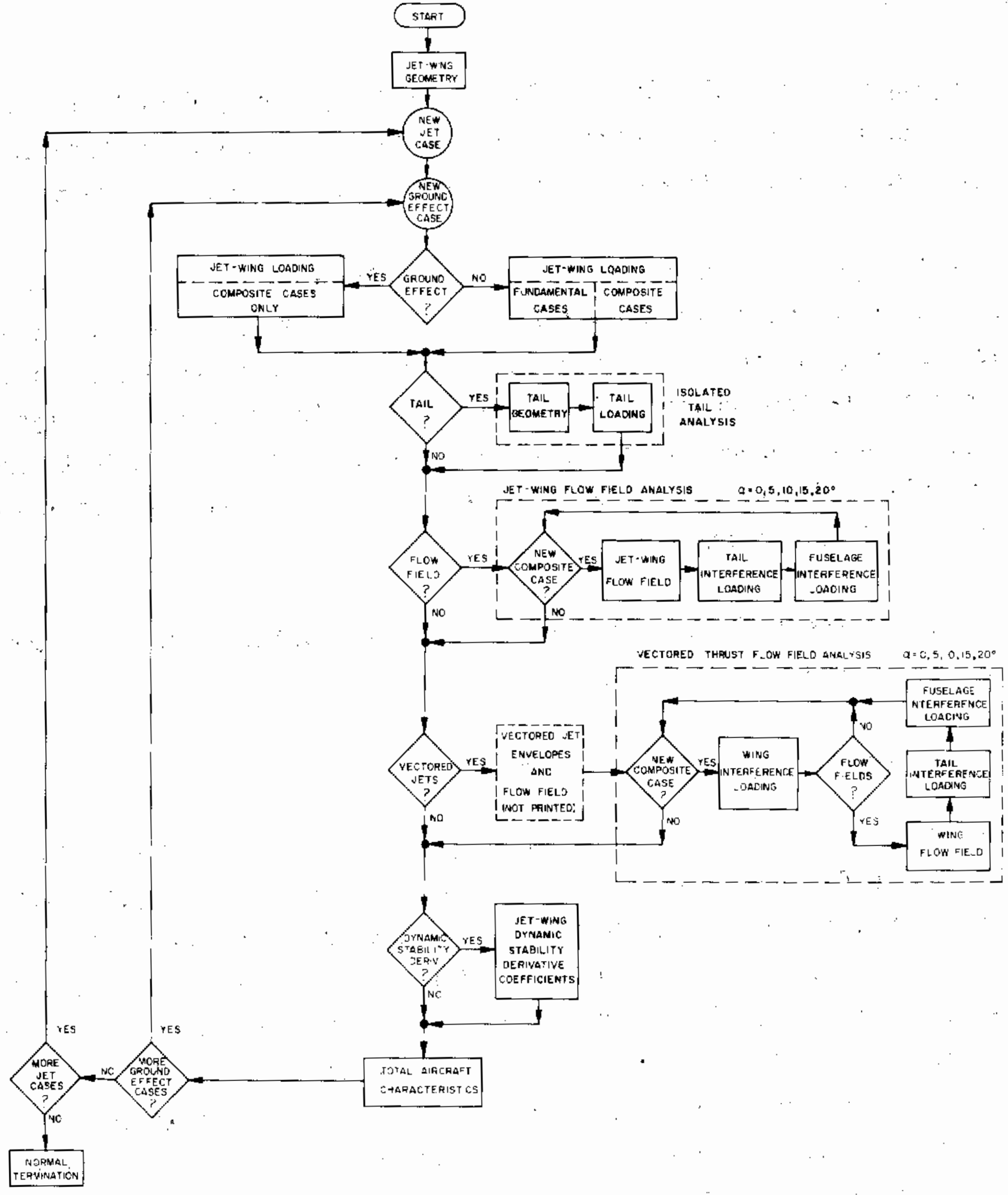


Figure (6.1). Flow Chart of STAMP Printed Output Sequence time of the run; specifying whether there is no jet, a jet sheet, or vectored jets, and specifying the jet case number; sepcifying if there is ground effect and, if so, giving the ground height (in the units of wing semispans); and specifying whether the current program function produces dynamic stability derivative data.

The program automatically normalizes all the geometry data to the units of wing semispan. Thus, all input geometry data is normalized by SPAN/2. The program works exclusively with a wing which has a span of 2.0 units.

Even though many significant digits are printed for most output data, this is by no means meant to imply that the coefficients are accurate to such a degree. So many digits are printed only so that the effects due to small deflections or planform changes will be apparent to the user in the F-type format used. While changes in the last significant digits may indicate the trends of effects due to small changes, they must not be construed to indicate their magnitude accurately.

In reading the following sections it will be helpful to refer to the sample problem output data in Appendix B.

6.1 Basic Parameters and Control Flags

The program name and input case title are printed at the top of the first page for reference. Next, the basic aircraft and jet-wing parameters are printed. Since the program will automatically redefine some of these quantities, both the input value and the internally used value are printed for reference. Note that AREA, SPAN, and CREF will be used to normalize all total aerodynamic coefficients for all aircraft components, and XMC and XCG will be used for all total aircraft moments and stability derivatives. CMAC and ARATIO apply only to the jet-wing. Next the basic case control flags are printed. These are fully described in Section 5.1.

6.2 Jet-Wing Data

The first set of data to be printed is jet-wing data or, for a no-jet case, wing data alone. These data will not be printed if there is no EVD solution to be made (NROWS = 0 or IFFLD < 0).

6.2.1 Fundamental Case Geometry

A header is printed to identify the data as "Element Geometry and Fundamental Case Data for All Fundamental Cases." A summary description of the components of each fundamental case is printed to serve as a guide to the subsequent data throughout the program. Internally generated fundamental cases to be used for dynamic stability derivative data are also identified. The following descriptors, and any combination thereof, are used to identify the fundamental case type:

a.	ALPHA CASE	(angle of attack case)
b.	WING TWIST	(twist case)
c.	WING CAMBER	(camber case)
d.	L.E. HEIGHT	(leading edge height case)
e.	JET TURNING	(jet deflection case)
f.	L.E. HINGE	(leading edge flap case)
g.	T.E. HINGE	(trailing edge flap case)

Next, a complete description of the elemental and sectional geometry is printed, section-by-section, starting at the right wing tip. The first line of data for each section contains the section number, the centerline value of y for that section (Y), the section half-width (DELTA), the leading and trailing edge x-coordinates for the section centerline (XLEAD and XTRAIL), the section chord (CHORD), and the tangent of the leading edge sweep angle for the section (TANLE). It should be remembered that these data are in the wind axis system and all linear dimensions are in the units of wing semispan. Next, for each of the ten fundamental cases (FC), the twist angle (TWIST), jet deflection angle (DJ) for sections with a jet sheet, and the leading edge height (HL) are printed. This is followed by a complete description of each element on the section for each fundamental case, including

a. The element number

b. Chordwise location of the leading edge of each element, normalized by local chord (x/c).

c. Chordwise location of the leading edge of each element, dimensional (x).

d. Chordwise element length (DEL), normalized by local chord.

e. Total incidence angle of each element (EPS) for each fundamental case, in degrees.

The same data are printed for each jet element, except EPS is replaced by THETA, the total jet turning angle, which is printed only for the leading jet element. The element length (DEL) for the last jet element is printed as ******* since this element is of infinite length (and is assigned the value 10^{10} by the program).

Following the fundamental case geometry data, the sectional jet strength (CMU) value at each spanwise section is printed, whether or not it is a jet section. This printout is omitted if JETFLG $\neq$ 0.

6.2.2 Fundamental Case Coefficients

Once the basic problem solution is complete, the jet-wing chordwise loading Δc_p is printed for all fundamental cases. These data are printed section-by-section, first for the wing element loading and then for the jet element loading. In addition, detailed loading at small intervals are given for all Leading Edge and Hinge EVD elements to aid in plotting pressure distributions. The hinge detailed print is given only if IHINGE $\neq$ 0. Otherwise, hinge loadings are represented by Regular EVD's rather than singular Hinge EVD's. If the user desires a more detailed pressure distribution in the vicinity of the leading edge of a section or near a hinge point, he may take the average value of the singular EVD printed by the program and use the following equations to determine Δc_p:

Leading Edge:

$$\Delta c_{pXB} = \frac{2}{3}\Delta c_{pi}\left[\left(\frac{XB}{\delta_i}\right)^{-1/2} - \frac{XB}{\delta_i}\right] + \Delta c_{pi+1}\frac{XB}{\delta_i} \qquad (6.1)$$

Hinge Point:

$$\Delta c_{pXB} = \frac{-4\beta_i}{\pi}\left[\log\left(c|XB-XH|\right) + \frac{(XB-XH)}{\delta_{i-1}}\log\left(c\delta_{i-1}\right)\right]$$

$$+\left[\Delta c_{pi} + \frac{\Delta c_{pi}-\Delta c_{pi-1}}{\delta_{i-1}}\left(XB-XH\right)\right] \quad (\text{for } XB < XH) \qquad (6.2a)$$

$$\Delta c_{pXB} = \frac{-4\beta_i}{\pi}\left[\log\left(c|XB-XH|\right) - \frac{(XB-XH)}{\delta_{i}}\log\left(c\delta_{i}\right)\right]$$

$$+\left[\Delta c_{pi} - \frac{\Delta c_{pi}-\Delta c_{pi+1}}{\delta_{i}}\left(XB-XH\right)\right] \quad (\text{for } XB > XH) \qquad (6.2b)$$

where c is the local chord, δ is the element length, XB is the point where the loading is required, XH is the hinge point, and i refers to the leading edge or hinge element. XB, XH, and δ are normalized by c.

The complete spanwise loading data for each fundamental case are printed immediately following the chordwise data, including lift, pitching moment, center of lift, and induced drag for each wing section. At the bottom of each column of sectional data, the values labeled total are the corresponding integrated aerodynamic coefficients for the entire wing. The sectional pitching moment coefficients are taken about the sectional leading edge, while the total pitching moment coefficients are taken about the wing apex (origin) and about the input XMC location. Sectional center of pressure and center of lift data are normalized by the sectional chord. The total values indicate the chordwise centers of pressure and lift, each normalized by both the reference chord and the wing semispan. In the sectional induced drag output, the quantity GAMMA is the sectional vorticity in the Trefftz plane and ALFIN is the sectional downwash at infinity. The total quantity printed under the ALFIN column is the induced drag calculated by the momentum method in the Trefftz plane.

For dynamic stability derivative runs (IDERIV > 0), the pitching rate derivatives for the jet-wing are printed directly below the spanwise loading data for the last fundamental case, which is internally generated by the program.

Finally, a summary table of all total aerodynamic coefficients for all fundamental cases is printed. These values are the same as those printed under the spanwise loading data, but also included in this table are total wing rolling moment, yawing moment, and side force. The abbreviations used in this table are defined in the list of symbols. The most important and most frequently used data are labeled by an asterisk.

6.2.3 Composite Case Coefficients

If the user has specified any composite cases, the loading data for them are printed immediately following the fundamental case data, one composite case at a time. The first line of output following the headers contains the input fundamental case multiplication factors for the composite case. Reference to the summary description of each fundamental case in the geometry section of the output will aid in determining the composition of each composite case. Next, the composite case chordwise loading data are printed for each section, both on the wing and on the jet sheet. The Δc_p distribution is given both for zero angle of attack (denoted by CPO) and for the variation in loading with angle of attack (denoted by CPA). Therefore, the pressure jump coefficient at an arbitrary angle of attack for a composite case is simply

$$\Delta c_{p_i} = \mathrm{CPO} + (\mathrm{CPA})\ \alpha \tag{6.3}$$

where α is the angle of attack in degrees. More detailed chordwise pressure distributions in the vicinity of the leading edge and near a flap hinge can be calculated in a manner analogous to that for fundamental cases using equations (6.1) and (6.2), but now with the composite loading.

Composite case spanwise loading data are printed in a manner such that the user can readily calculate the sectional and total aerodynamic coefficients at any angle of attack (within the constraints of linearized theory, of course). The first page of spanwise loading data gives the linear aerodynamic coefficients (lift, pitching moment, and lift center), and both the sectional and total values are given for zero angle of attack (denoted by the suffix 0) and for the variation with angle of attack (denoted by the suffix A). Thus, to determine the sectional lift at any angle of attack, the program output must be combined as

$$c_\ell = \text{CLO} + (\text{CLA})\ \alpha \tag{6.4}$$

while for the sectional pitching moment,

$$c_m = \text{CMO} + (\text{CMA})\ \alpha \tag{6.5}$$

Sectional pressure or lift center cannot be linearly combined directly, but rather they must be recalculated from the moment and lift data. Hence, the sectional pressure and lift centers at any angle of attack can be calculated from the program output as, respectively,

$$\bar{X}_{c.p} = -\frac{\text{CMGO} + (\text{CMGA})\ \alpha}{\text{CLGO} + (\text{CLGA})\ \alpha} \tag{6.6}$$

$$\bar{X}_{c_\ell} = -\frac{(\text{CMGO} + \text{CMMUO}) + (\text{CMGA} + \text{CMMUA})\ \alpha}{\text{CLO} + (\text{CLA})\ \alpha} \tag{6.7}$$

Total aerodynamic coefficients, printed at the bottom of each column of data, are combined in an analogous manner. Total pitching moments are printed both relative to the wing apex and relative to the input moment center. The first two lines of the total pressure and lift center data are the chordwise location normalized by the reference chord, and the third and fourth lines are these quantities normalized by the wing semispan.

The second page of composite case spanwise loading data gives the (nonlinear) induced drag coefficients. For the induced drag calculated by the pressure integration method, three terms are required for each coefficient, one for zero angle of attack (suffix 0), one for the variation with angle of attack (suffix A), and one for the variation with angle of attack squared (suffix A2). Thus, the sectional induced drag at any angle of attack can be calculated from the program output as

$$c_{d_i} = \mathrm{CDO} + (\mathrm{CDA})\ \alpha + (\mathrm{CDA2})\ \alpha^2 \qquad (6.8)$$

The total induced drag terms are printed at the bottom of each column and are treated similarly. The two columns of data following the pressure integral induced drag are the vorticity at infinity and the induced downwash at infinity, both used to calculate induced drag by the momentum method. The total quantities printed under the latter column are the three induced drag coefficients calculated by the momentum method. The column CTO is the thrust coefficient at zero angle of attack and is simply

$$\mathrm{CTO} = \mathrm{CMU} - \mathrm{CDO} \qquad (6.9)$$

and is printed for reference only, as is the CMU column.

A summary of the total aerodynamic coefficients is printed for the composite case, giving both alpha and alpha-squared variations, where applicable. The most important terms are denoted by asterisks. Finally, a summary table of the variations with angle of attack of total lift, pitching moment (about XMC), rolling moment, induced drag (momentum or Trefftz plane method, denoted by CDITZ), and yawing moment (about XMC) is printed. The coefficients are evaluated at angles of attack from -10 degrees to +30 degrees in one degree increments.

6.2.4 Dynamic Stability Derivatives

The jet-wing dynamic stability derivatives due to rates of pitch, roll, and yaw are calculated by the program when IDERIV > 0. The pitch rate (q) derivatives have already been discussed since they are printed following the last fundamental case output data. The rolling (p) and yawing (r) rate derivatives are calculated in a second execution of the program (see Appendix A) and are identified by the header "Second Run for Stability Derivative Case." All input fundamental cases are modified for the yawing rate stability derivatives while the additional fundamental case created for pitching rate is modified for rolling rate. All stability derivatives are calculated for a rate of one radian per second. The output for fundamental case yawing and rolling rate derivatives is quite explicit, giving the equations necessary to determine the necessary moments and forces. For example, the equation

$$CN(R) = CNR*R + CNR2*R**2 \qquad (6.10)$$

means that the yawing moment due to a rate of yaw R can be found from the two derivatives, CNR and CNR2, which are calculated by the program.

Stability derivatives output for composite cases again includes the input fundamental case multiplication factors. Most of the composite case stability derivatives include terms which vary with angle of attack and the rates themselves. The output gives all the terms explicitly as well as formulas for calculating the moments and forces due to the rates of rotation as a function of angle of attack and rotation rate. Finally, a table of the variation with angle of attack of the rolling and yawing rate dynamic stability derivatives is printed.

6.2.5 Ground Effect Coefficients and Derivatives

Output for the jet-wing in ground effect, although similar to the free air solution data in many respects, has some basic differences which the user must recognize. Primarily, since the aerodynamic coefficients and derivatives for a jet-wing in ground effect are not linearly combinable, only composite case loading data are calculated. The ground height in units of wing semispan is printed in the page header for reference. The fundamental case multiplication factors are printed at the top of the chordwise loading data, as done before. To avoid confusion, note that the composite case geometry is still formed from the originally input fundamental case geometries, so the multiplication factors are necessary in the ground effect output, but the aerodynamic coefficients are no longer formed by simple summation of the fundamental case results. The composite case chordwise loading data are printed for each section, both on the wing and on the jet sheet. Unlike the free air solution, the Δc_p distribution in ground effect requires a term for zero angle of attack (CPO), a term for the variation with angle of attack (CPA), and a term for the variation with angle of attack squared (CPA2). Therefore, the pressure jump coefficient at an arbitrary angle of attack for a composite case in ground effect is

$$\Delta c_{p_i} = \text{CPO} + (\text{CPA})\,\alpha + (\text{CPA2})\,\alpha^2 \tag{6.11}$$

More detailed chordwise pressure distributions in the vicinity of the leading edge can be calculated using equation (6.1). The program does not allow hinge singularities for ground effect runs.

Composite case spanwise and total loading data in ground effect are printed in a manner analogous to the free air case output except that each lift and moment term has an additional term one order higher in angle of attack than before. Thus, the sectional and total lift, pitching moment, and induced drag can be calculated at any angle of attack from, respectively,

$$c_\ell = \text{CLO} + (\text{CLA})\,\alpha + (\text{CLA2})\,\alpha^2 \tag{6.12}$$

$$c_m = \text{CMO} + (\text{CMA})\,\alpha + (\text{CMA2})\,\alpha^2 \tag{6.13}$$

$$c_{d_i} = \text{CDO} + (\text{CDA})\,\alpha + (\text{CDA2})\,\alpha^2 \tag{6.14}$$

The sectional and total pressure or lift center must be determined by dividing the appropriate sectional moment by the appropriate sectional lift.

A summary of the total aerodynamic coefficients in ground effect is printed for the composite case, giving the alpha and alpha-squared terms, where applicable. Finally, a summary table of the variation of the aerodynamic force and moment coefficients with angle of attack is printed, identical to the free air solution output.

Dynamic stability derivative output for the ground effect solution is different from that of the free air solution because only the composite case data are calculated and, as a result of the non-linearity in the pressure coefficient, additional stability derivatives are required. The output explicitly defines the forces and moments resulting from the rotation rates as a function of angle of attack and the rates themselves, and prints a table of all required terms. Finally, a table of the variation with angle of attack of the pitching, rolling, and yawing dynamic stability derivatives in ground proximity is printed.

6.3 Empennage Data

For configurations including a horizontal tail and/or a vertical tail, a summary of the planform geometry characteristics and other pertinent geometric data are printed. All tail geometric data are in the units of wing semispan, referenced from the origin of the coordinate system (wing apex). Symbols used in this printout are defined in the list of symbols.

6.3.1 Isolated Empennage Coefficients

Immediately following the tail geometry summary, loading data for the empennage isolated from the other aircraft components are printed. Labeled as "Isolated Tail Analysis," this solution is used to determine, within the limits of linearized theory, the angle of attack derivatives of the horizontal tail, the sideslip derivatives of the vertical tail, the basic load level due to tail camber (including elevator and rudder deflection), and the empennage dynamic stability derivatives. The program internally generates the necessary cases to accomplish this solution, based on the user inputs. Up to three cases can be included in the isolated tail solution:

a.	Alpha Case	(angle of attack)
b.	Beta Case	(angle of sideslip)
c.	Camber Case	(including deflections of elevators, rudder or trim tabs)

Headers are printed above the tail chordwise loading data to identify these cases. The tail chordwise loading Δc_p is printed section by section, horizontal tail first, then vertical tail. It must be noted that the Δc_p values printed here are obtained from vortex lattice theory, and hence, only a simple discrete distribution has been assumed. It is recommended that if the empennage pressure distributions must be plotted, that the values of Δc_p printed be placed at the quarter-chord point of their respective elements.

Following the empennage chordwise loading data, the spanwise and total aerodynamic coefficients are printed. Case assignment is the same as for the chordwise loading. For each case, the horizontal tail sectional lift, pitching moment (about the section leading edge), and induced drag are printed, normalized by the sectional chord. At the bottom of each column, the total horizontal tail lift, pitching moment (about the input aircraft XMC), and induced drag are printed, normalized by wing reference area and reference chord. In addition, the total horizontal tail rolling moment (CLLH) is given. In a similar manner, the side force, yawing moment, induced drag, and rolling moment contributions of the vertical tail are printed.

Use of the isolated empennage coefficients is primarily intended to supplement the interference analysis data (Section 6.3.3), which is non-linear in angle of attack because of the non-linear influence of the jet-wing and vectored jet flow fields. For example, since the non-isolated empennage loading is calculated at specific angles of attack (0, 5, 10, 15, and 20 degrees),local interpolation of these data to some angle in between the specified angles can be done using the angle of attack derivatives from the isolated empennage loading. Also, quantities such as elevator power can be obtained from the isolated empennage solution by representing the elevator as horizontal tail camber. This is more fully discussed in Volume III.

6.3.2 Dynamic Stability Derivatives

The empennage dynamic stability derivatives due to rates of pitch, roll, and yaw are calculated when IDERIV > 0 and are printed immediately after the isolated empennage solution. Based on the assumptions discussed in Section 4.2.4 of Volume I, all of the tail dynamic stability derivatives are independent of both angle of attack and the rotation rates. The derivatives printed are per radian rate of rotation and are non-dimensionalized by wing area, reference chord, and span. The axis of rotation is the input aircraft XCG for pitching and yawing, and the x-axis for rolling.

6.3.3 Non-Isolated Empennage Coefficients

When the user inputs both a wing (with or without jet or jet sheet) and a tail (horizontal and/or vertical), the program will automatically calculate the flow field interference of the wing and jets on the tail and will then calculate the non-isolated empennage aerodynamic characteristics. These interference tail data are printed in the same format as the isolated tail data, except that five cases are now printed, one for each of the angles of attack. Each case is composed of the specified angle of attack, induced camber equivalent to the jet-wing and vectored jet flow field, and the physical camber distribution input by the user. A complete set of interference empennage data are calculated and printed for each composite case of the jet-wing.

6.4 Fuselage Data

Because of the massive amount of output data required to completely define the geometry and loading for the fuselage, the normal output mode for the fuselage analysis segment of STAMP has been limited to a brief summary of the results. This summary, described in the next section, should be adequate for most users of the program. The full output of fuselage analysis can be obtained by specifying the special print option (IPRINT = -1), but since this will also cause jet-wing and empennage matrices to be printed, it is recommended that the detailed print for fuselages be limited to fuselage alone runs (i.e., no wing, jet, or empennage).

6.4.1 Fuselage Coefficients

A summary table of the fuselage sectional and total pitching and yawing moments at angles of attack of 0, 5, 10, 15 and 20 degrees are printed under all normal print modes. Included in this table is the streamwise coordinate of the center of each fuselage segment relative to the wing apex, positive aft. If a wing with or without jets is also input, the program automatically calculates the flow field interference of the wing and jets on the fuselage and includes these flow field data in the fuselage solution. Because of the nonlinearity of the jet-wing flow field, the fuselage coefficients are calculated and printed for each composite case of the jet-wing.

If the special print option is specified (IPRINT = -1), in addition to the summary table described above, the complete details of the fuselage geometry and loading are also printed. The first page of output is a summary of the fuselage axial flow solution, the longitudinal portion of the slender body solution. The data are printed for each fuselage segment and may be summarized as follows:

a. SEG - Streamwise segment number
b. X - Streamwise location of the center of each segment, relative to the origin (wing apex).
c. R - Radius of the equivalent body of revolution at x.
d. RP - Gradient of the radius $\left(dR/dx\right)$ at x.
e. A - Cross-sectional area of the equivalent body of revolution at x.
f. AP - Gradient of the cross-sectional area $\left(dA/dx\right)$ at x.
g. UX - Streamwise perturbation velocity due to the longitudinal flow over the equivalent body of revolution at x.
h. UR - Radial perturbation velocity due to the longitudinal flow over the equivalent body of revolution at x.

Following the axial flow solution, the cross-flow solution is calculated and the complete pressure distribution is determined and printed. The circumferential geometry and pressure distributions at five angle of attack are printed next, one segment per page. For symmetric fuselages only the right half of the segment circumference is printed.

6.4.2 Dynamic Stability Derivatives

Fuselage dynamic stability derivatives are calculated and printed only once in any run, immediately following the first loading summary table printed. Based on the assumptions and limitations of the fuselage method discussed in Section 4.1 of Volume I, there are only two fuselage dynamic stability derivative contributions. The derivatives are per radian and are non-dimensionalized by wing area, reference chord, and span. The axis of rotation is the input XCG.

6.5 Jet-Wing Flow Field Data

The jet-wing flow field segment of STAMP is executed whenever the user has requested flow field data explicitly or whenever jet-wing interference effects on the empennage or fuselage are required. Flow field output data is printed for each composite case at five angles of attack, since the induced velocity field is non-linear in all the deflection angles. Page headers indicate the composite case number and the angle of attack of the data.

Each set of flow field output contains the jet-wing induced velocity at the following points:

a. NMAXH points on the horizontal tail.
b. NMAXV points on the vertical tail.
c. NSEG points on the fuselage centerline.
d. NPOINT off-body points.

The data for each point consists of the following:

a. (x,y,z) location of the point, in the body axis system with its origin at the wing apex, in the same units as wing semispan.
b. (U/UINF,V/UINF,W/UINF) - Perturbation velocities in the (x,y,z) directions, respectively, normalized by the freestream velocity. If there are vectored jets, the perturbation velocities induced by them are also included.
c. DOWNWASH ANGLE (DEGREES) - This is defined by

$$\varepsilon = - \arctan [W/UINF/(1 + U/UINF)]$$

and is positive for downwash.
d. SIDEWASH ANGLE (DEGREES) - This is defined by

$$\beta = \arctan [V/UINF/(1 + U/UINF)]$$

and is positive for induced flow in the +y direction.

Two warning messages may be printed in place of the flow field data, each indicating that the flow field point in question falls exactly on either a bound or trailing discrete vortex filament. The program internally sets these induced velocities to zero. The user is warned that the resultant velocity at such a flow field point is, therefore, in error and any aerodynamic data generated for the associated element on the wing, tail or fuselage may also be in error.

6.6 Vectored Jet Data

The vectored jet flow field segment of STAMP is executed in order to find the interference effects of vectored jets on the wing, tail or fuselage. The vectored thrust power-off condition is always computed first, and then the interference effects are computed for as many groups of vectored jets as are input.

Detailed output for vectored jets is given in two parts:

a. Vectored jet envelope shape.
b. Flow field perturbation velocities induced by all jets.

This output is given only when the detailed print option (IPRINT = -1) is selected by the user.

The vectored jet envelope shapes are given for each jet included in a given configuration group, in the sequence in which they were input. Page headers indicate the jet number and angle of attack of the data and the following input quantities:

CJ	Jet thrust coefficient.
DO/B/2	Jet exit diameter, normalized by half span.
THETAJ	Jet exit angle relative to the freestream (including the angle of attack), in degrees.
DIHED	Jet exit dihedral angle, in degrees.
ISYM	Jet symmetry flag.
XO/B/2	Jet exit coordinates, normalized by half span.
YO/B/2	
ZO/B/2	
H	Incremental step size of jet segments, normalized by half span.

The jet envelope is described by the following:

N	Jet segment number
X/B/2	Local segment coordinates, normalized by half span
Y/B/2	
Z/B/2	
UJ/U	Local jet velocity, normalized by freestream velocity
THETA	Local angle relative to the freestream, in degrees

The flow field data is given for all jets in a configuration group summed together and includes the following:

N	Flow field point number in the following sequence:

a. NMAX points on the wing
b. NMAXH points on the horizontal tail

c. NMAXV points on the vertical tail
d. NSEG points on the fuselage centerline
e. NPOINT off-body points

X/B/2	Nondimensional coordinates of flow field points
Y/B/2	
Z/B/2	
VX	Perturbation velocities in the (x,y,z) directions, respectively, normalized by the freestream velocity
VY	
VZ	

If the number of jet segments or the number of flow field points is odd, the data for the last (even) point in either of the sets of output data above is undefined. This sometimes results in the appearance very strange numbers, but these have no effect on the jet computations and should be ignored.

A warning message will be printed, as in the jet-wing flow field output of Section 6.5, if a flow field point happens to lie inside the jet envelope. In such an event, the induced velocities due to the jet segment immediately surrounding the flow field point are computed as usual, i.e., no remedial action is taken. The user is warned that the resulting interference loading may have large local errors because the jet is probably "passing through" the aircraft component containing the point in question.

Since the vectored jet envelope shapes vary non-linearly with the exit angle relative to the freestream, the jet envelope and resulting perturbation velocities are repeated for angles of attack of 0, 5, 10, 15, and 20 degrees. The jet envelopes are assumed to be uncoupled from the wing-induced flow field perturbations, and therefore, are not to be recomputed for each composite case.

6.7 Summary of Aerodynamic Coefficients

When the configuration run is composed of a wing (with or without jets) and a tail and/or a fuselage, a summary of the aircraft aerodynamic coefficients is printed for each composite case. The summary table will include data for the following components at angles of attack of 0, 5, 10, 15, and

20 degrees:

a. Wing (or jet-wing)
b. Horizontal tail
c. Vertical tail
d. Fuselage
e. Complete aircraft

For each component the three static aerodynamic force components and the three static aerodynamic moment components are printed, where applicable, and the basic dynamic stability derivatives are also printed. Only the constant term of each stability derivative is printed here since it is that term which is likely to be used in the linearized equations of motion normally employed in stability and control studies, but the user is cautioned to check the angle of attack and rate dependent terms printed earlier in the program to see if their magnitude is significant.

If vectored jets are included in the configuration, two summary tables will be printed, the first for vectored jets off, the second for vectored jets on.

7.0 OPERATIONAL CONSIDERATIONS

The STOL Aerodynamic Methods Program (STAMP) has been written entirely in the Fortran IV Language and can be run on any large scale computing system. Wherever possible, the use of special capabilities of particular systems or Fortran processors have been avoided in order to facilitate conversion from the IBM 370 system of the Douglas Aircraft Company, under which the program was originally developed, to any other system. The program has been converted for use on CDC 6000 series computers, and it is expected that only minor changes would be required for running under any other large computing system.

7.1 Overlay Structure

Because of the very large size of STAMP, the overlay feature of Fortran IV has been utilized to permit use of the minimum possible core. The overlay structure of STAMP is shown in figure (7.1). This structure requires approximately 210000 (octal) words of storage on the CDC 6500 and approximately 250K bytes of storage on the IBM 370.

7.2 Peripheral Storage Devices

STAMP requires the use of 13 temporary peripheral storage devices, plus any required by the local system. Units 1, 8, and 9 are mass storage (direct access) disk files and the remaining devices may be either tape or disk units. The peripheral storage units have the following functions:

Unit	Characters	Function
1	Binary	Mass storage of downwash influence coefficients and partial augmented solution matrix for no ground effect.
2	Binary	Solution matrix input to quasi-inversion routine (MATRIX).
3, 4	Binary	Utility storage during matrix quasi-inversion and back-solution.
5	IBCDIC or BCD	Standard card input unit.
6	IBCDIC or BCD	Standard printed output unit.
7	IBCDIC or BCD	Standard punched output unit.

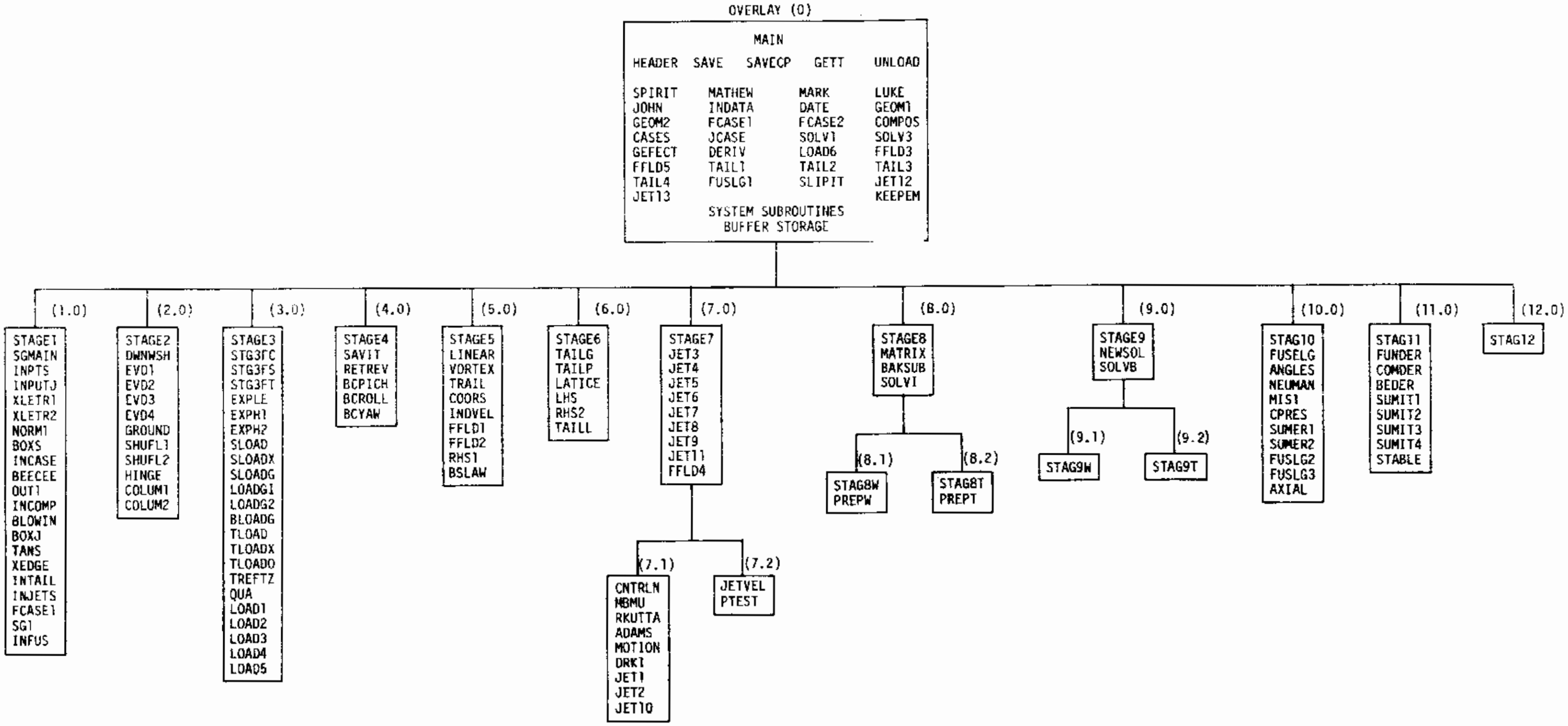


Figure (7.1). Program Overlay Structure.

Unit	Characters	Function
8		Mass storage at downwash influence coefficients and partial augmented solution matrix in ground effect, tail normal-wash influence coefficients, and vectored jet-induced flow field velocities.
9	Binary	Mass storage of wing ground effect perturbation velocity influence coefficients and fuselage geometry characteristics.
10,11	Binary	Storage of wing matrix quasi-inversion data for subsequent use in wing back-solution.
12,13	Binary	Storage of tail matrix quasi-inversion data for subsequent use in tail back-solution.

7.3 Program Execution Time

The computing time required for execution of STAMP is mainly dependent on the following factors:

a. Number of EVD elements on wing and jet-sheet
b. Number of wing composite cases
c. Requirement for jet-wing flow field data (i.e., whether or not an empennage, fuselage or off-body flow field points are required)
d. Number of vectored jets
e. Whether or not ground effects are required
f. Whether or not dynamic stability derivatives are required

The number of wing fundamental cases has a relatively small effect on computing time, as do the number of tail elements and fuselage segments.

The estimated computer running time must be found by summing the time required for each phase of the program. The four phases of STAMP which contribute significantly to the time are:

1. EVD jet-wing solution $\left(T_{EVD}\right)$
2. Jet-wing flow field solution $\left(T_{\substack{J-W\\F.F.}}\right)$
3. Vectored jet flow field solution $\left(T_{\substack{V.T.\\F.F.}}\right)$

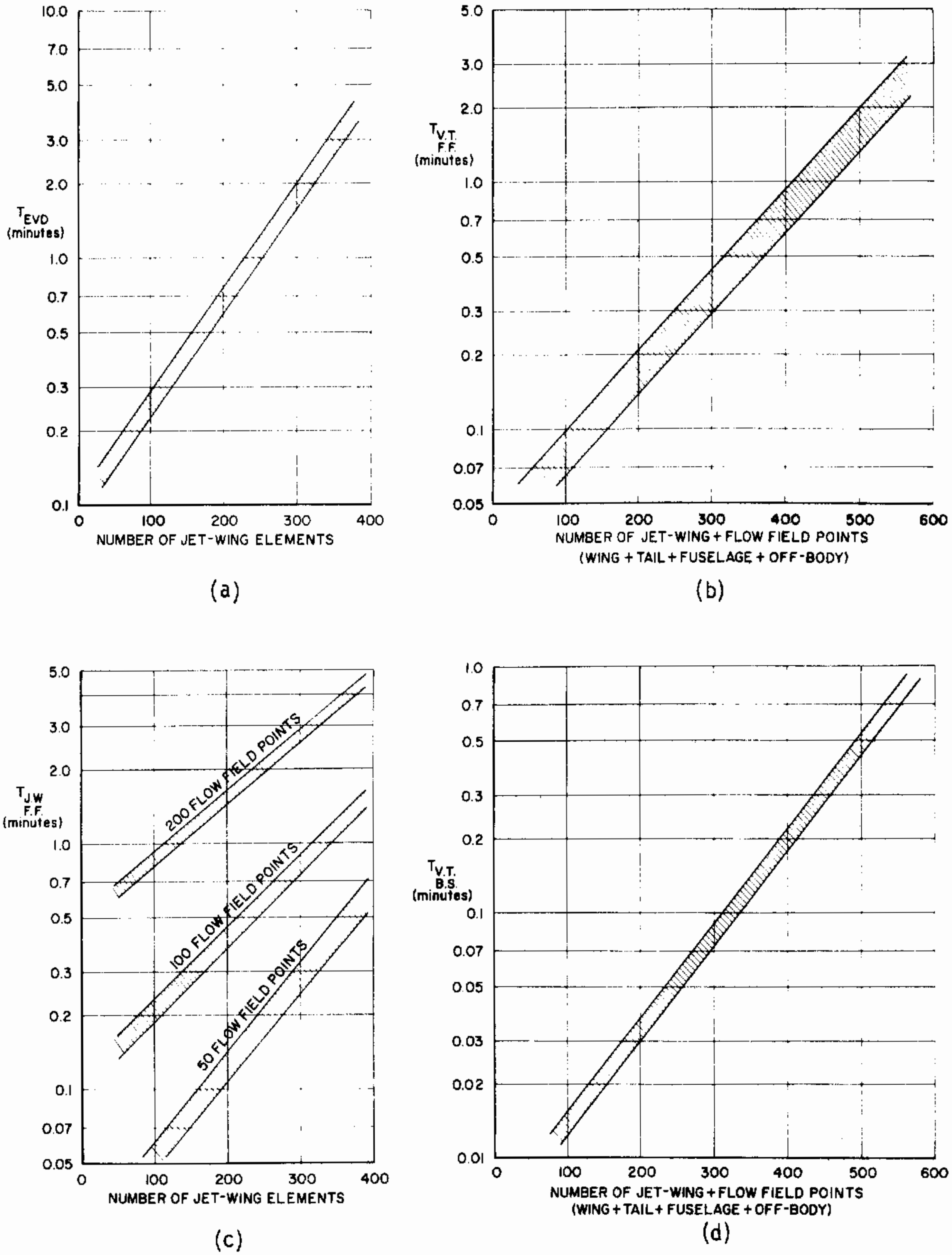


Figure (7.2). Program Execution Time Parameters.

4. Wing matrix back-solution for vectored jet-induced wing loading $\left(T_{\substack{V.T.\\B.S.}}\right)$

An approximate estimate of execution time may be derived from the following formula:

$$T = (1 + G)\left\{\left(1 + \frac{N_{C_\mu}}{2}\right) T_{EVD} + 5\ N_{C.C.}\left(N_{C_\mu} + N_{V.T.} + 1\right) T_{\substack{J\text{-}W\\F.F.}} + \left[5\ J\ T_{\substack{V.T.\\F.F.}} + T_{\substack{V.T.\\B.S.}}\right]\left[N_{V.T.}\right]\right\} \tag{7.1}$$

where G = number of ground effect cases input (i.e., not including the free-air case)

J = number of vectored jets in each vectored jet configuration

$N_{C.C.}$ = number of wing composite cases input

N_{C_μ} = number of jet sheet C_μ cases input

$N_{V.T.}$ = number of vectored jet configurations input (i.e., not including the power-off case)

The values of T_{EVD}, $T_{\substack{J\text{-}W\\F.F.}}$, $T_{\substack{V.T.\\F.F.}}$, and $T_{\substack{V.T.\\B.S.}}$ are dependent on number of wing elements, tail elements, fuselage elements, etc., and are shown in figures (7.2a) through (7.2d). CPU (Central Processor Unit) time refers to real time used in calculation. IO (Input/Output) time refers to an equivalent time charged for processing on peripheral storage devices and may be known as PP (Peripheral Processor) time. This includes reading and writing of temporary data sets such as the card input unit, the printed output unit, scratch tapes, punched cards, etc. Total time is the sum of CPU time and IO time.

Other machines and other installations may require a different amount of time for execution. It is expected that CPU time will vary among machines, but will be representative of the speed of any particular model. On the other hand, IO time is expected to vary from installation to installation, depending upon the facilities available, the demand placed upon them, their cost, etc.

8.0 REFERENCES

1. Goldhammer, M. I., Lopez, M. L., and Shen, C. C.: Methods for Predicting the Aerodynamic and Stability and Control Characteristics of STOL Aircraft. Vol. I: Basic Theoretical Methods. AFFDL TR-73-146, Vol. I, also McDonnell Douglas Report No. MDC J5965-01, 1973.

2. Goldhammer, M. I. and Lopez, M. L.: Method for Predicting the Aerodynamic and Stability and Control Characteristics of STOL Aircraft. Vol. III: Engineering Methods. AFFDL TR-73-146, Vol. III, also McDonnell Douglas Report No. MDC J5965-03, 1973.

3. Hess, J. L. and Riddell, T. M.: Direct Solution of a Square Matrix Whose Size Greatly Exceeds High-Speed Storage. Douglas Report No. DAC 70000, July, 1969.

4. Wooler, P. T., Kas, H. C., Schwendemann, M. F., Wasson, H. R., and Ziegler, H.: V/STOL Aircraft Aerodynamic Prediction Methods Investigation. Vol. III: Manual for Computer Programs. Technical Report AFFDL-TR-72-26, Vol. III, January, 1972.

APPENDIX I

SUBROUTINE DESCRIPTIONS

OVERLAY SEGMENT 0

Main Control and General Utility Routines

Main Program (STAMP)

This program controls the sequence and use of all calculations in the STAMP program system. The program first reads the title card and the general geometry parameters and basic control flags. Certain logical control variables are defined according to the input flags, and the consistency of the flags is checked.

Next the loop on jet blowing strength is entered. This is the outermost program loop. Certain flags are initialized and STAGE1 is called to read all input and prepare all the jet-wing geometry data. On successive cycles, STAGE1 merely reads the jet data for a new jet strength case.

The loop cycling the ground height is next begun, and additional control flags and counters are defined. The case of no ground effect is always treated before the input ground heights are considered.

If dynamic stability derivatives are required, one extra fundamental case is added to those input and STAGE4 is called to define the equivalent induced camber due to pitching. The printout of the jet-wing geometry and incidence angles is provided by calling STAGE1 if this has not been done previously.

Next, STAGE2 is called for calculation of the jet-wing influence coefficient matrix. For symmetric jet-wings with dynamic derivatives required, an anti-symmetric matrix is also formed. The matrix solution is obtained from STAGE8, saved on disk by calling SAVECP, and punched on cards if requested.

Finally, the basic jet-wing run is completed by calling STAGE3, where all the aerodynamic coefficients are computed.

The isolated tail analysis is performed by successive calls to STAGE6 and STAGE8. The tail geometry is first defined. Then the influence coefficients are computed, the matrix solved, and the loading integrated to get aerodynamic coefficients. The process is repeated in order to compute the tail dynamic stability derivatives, if required.

The jet-wing flow field analysis begins by calling STAGE5 to compute all the jet-wing induced velocities on the tail and fuselage. The tail matrix system is back-solved in STAGE9 and all tail and fuselage aerodynamic coefficients are computed by calling STAGE6 and STAGE10. This sequence is repeated for each jet-wing composite case.

The vectored jet power-on interference analysis is begun by calling STAGE7 to calculate all jet envelopes and flow field induced velocities. The interference wing loading is computed by back-solving the wing matrix system in STAGE9 using the equivalent induced camber derived from the vectored jet induced velocities. Then for each wing composite case the complete wing loading is computed in STAGE3 and the wing flow field analysis is repeated as above, but including the jet flow field effects on all aircraft components.

If dynamic stability derivatives are required, STAGE4 is called to redefine the fundamental case incidence angles due to rates of roll and yaw, and STAGE1 prints the revised jet-wing geometry characteristics. STAGE2 is then called to augment the jet-wing dynamic stability matrix and define the right-side column matrix. The jet-wing matrix system is solved in STAGE8 and the dynamic stability derivatives are computed in STAG11. Finally, the summary of aerodynamic coefficients for all aircraft components are printed by calling STAG12.

The program next checks whether a new ground height case is required, or whether to start over on a new jet case. After all jet and ground height requirements have been completed, an attempt is made to begin a brand new run, and if no further input data is found, the execution is terminated.

Page Header Information (HEADER)

This subroutine prints program status information at the top of new groups of output data.

Save Basic Run Solution (SAVECP)

For stability derivative runs, the loading solution from the first run must be saved for use in the second run. This is accomplished by writing the data on mass storage unit 1.

Jet-Wing Loading Input or Output (UNLOAD)

This subroutine writes the jet-wing loading on unit 7 for card punching, or reads the jet-wing loading from input cards, for special flow field analysis of the tail or fuselage.

Scratch File Writing (SAVE)

Because of the large amount of writing to scratch files required in the matrix solution, this subroutine is used to write whole arrays as records without reference to subscripts, thus significantly increasing the solution speed.

Scratch File Reading (GETT)

As with subroutine SAVE, reading whole arrays without reference to subscripts significantly reduces the matrix solution time.

OVERLAY SEGMENT 1

Input and Jet-Wing Geometry Formation

Main Program (STAGE1)

This program controls the reading of all input data, calculation of most geometry characteristics, and printing of wing geometry and jet strength. The basic sequence of functions which are performed is as follows:

Read all jet-wing inputs and calculate all wing planform geometry parameters and deflection angles. Print jet-wing geometry characteristics. Read the tail and fuselage geometry inputs. Read and print the jet sheet strength or vectored jet parameters.

Some of the above operations may be executed separately, depending on commands from the main program.

Jet-Wing Geometry Control (SGMAIN)

This subroutine controls reading of the jet-wing spanwise and chordwise spacing, reading of planform geometry data for irregular or simple trapezoidal wings, conversion of the jet-wing geometry to a system with unit half-span, and reading of the jet sheet chordwise spacing. Next, the EVD element geometric characteristics on the jet-sheet are calculated. Finally, the fundamental case input data is read, from which the appropriate deflection angle arrays are derived.

Wing Element Input (INPTS)

This subroutine reads the spanwise spacing arrangement, and then the number and chordwise spacing of elements on the wing.

Irregular Wing Planform Input (XLETR1)

This subroutine reads the leading and trailing edge input coordinates to define the arbitrary planform, and linearly interpolates where necessary to define the leading and trailing edges at any sections for which no input is given.

Trapezoidal Wing Planform Input (XLETR2)

This subroutine reads the general planform parameters for simple symmetrical trapezoidal wings, and calculates the leading and trailing edge coordinates at each required spanwise section.

Wing Parameter Normalization (NORM1)

This subroutine normalizes all the dimensional planform parameters by the wing half-span. From this point on, the program deals entirely with the scaled wing, where the wing span is two units.

Jet Element Input (INPUTJ)

In a manner similar to that used in INPTS, the number and chordwise spacing of elements on the jet are read. Note that the spanwise spacing of jet elements is the same as for the wing, but that some or all of these jet sections may have no chordwise elements, thus effectively eliminating the jet at those sections.

EVD Element Definition (BOXS)

This subroutine uses the input data from the above subroutines to compute the length, width, and location of each EVD element on the wing and jet. In addition, the appropriate EVD type is chosen for each element, and several other parameters are defined for later internal use by the program.

Leading and Trailing Edge Sweep (TANS)

This subroutine computes the sweep angle of the leading and trailing edges at each section. This data is needed later for calculation of side force.

Fundamental Case Input (INCASE)

This subroutine reads, for each fundamental case, the flags indicating the types of deflections to be used, then the magnitude of each particular type of deflection. The types of fundamental case deflections read are twist angles, leading edge vertical displacement (for use later in pitching moment calculations), jet deflection angles, camber angles, and hinge location and deflection angles.

Fundamental Case Boundary Conditions (BEECEE)

From the fundamental case input, this subroutine computes, for each fundamental case, the resulting incidence angle of each EVD element relative to the freestream. This is done by summing the accumulative deflections due to twist, camber, and flap deflection from the leading to the trailing edges. The effect of jet deflection is also included on the first jet element of each section. Thus, the effects of all deflections are superimposed and the individual effect of, for example, deflection of a particular flap is no longer distinguishable by the program. Therefore, if the user wishes to be able to see the effects of that flap deflection separately from those of the other deflections, he must make use of a separate fundamental case, where only the flap deflection is present.

Fundamental Case Output (OUT1)

This subroutine prints out, for each fundamental case, the location, size, and resultant incidence angle of each EVD element. Parameters of spanwise significance are also printed, including wing chord, leading and trailing edge coordinates, leading edge height, twist angle, etc.

Composite Case Input (INCOMP)

This subroutine reads the user's requirements for composite cases, in which the linear fundamental cases may be superimposed in any combination to form a deflection pattern of particular interest.

Tail Geometry Input (INTAIL)

This subroutine reads the data describing the horizontal and vertical tail geometry, and computes several geometric parameters for later integral use by the program.

Fuselage Geometry Input (INFUS)

This subroutine reads the data describing the fuselage.

Flow Field Input (XEDGE)

This subroutine reads the coordinates of off-body flow field points, normalizes them by the wing half-span, and computes certain wing geometric parameters for later use in calculating the jet-wing flow field.

Jet Sheet Strength Input (BLOWIN)

This subroutine simply reads the sectional value of jet momentum coefficient, c_μ, for each section which has been specified to have jet elements.

Additional Jet Sheet Parameter Definition (BOXJ)

This subroutine defines an additional jet strength parameter for later internal use, and checks on the consistency of the input c_μ data. The jet strength from the previous c_μ case, if any, is also saved for later internal use.

Vectored Jet Input (INJETS)

This subroutine reads the geometric and strength characteristics of all vectored jets, computes certain additional parameters, and checks the validity of the input data.

OVERLAY SEGMENT 2

Formation of the Jet-Wing System of Simultaneous Linear Equations

This program controls development of the EVD jet-wing problem. The large square left-side matrix of EVD downwash influence coefficients is computed by calling DWNWSH. Next, this matrix is augmented in SHUFL1. If previous c_μ cases have been run, these steps are skipped and a simple reaugmentation is performed in SHUFL2. For each fundamental case, the right side column of constant boundary conditions is formed in COLUM1. Since the downwash influences of the logarithmic singular part of the Hinge EVD's depend only on the hinge turning angles and thus are known, these influences are computed in HINGE and superimposed directly on the right side matrix in COLUM2.

Downwash Influence Coefficients (DWNWSH)

This subroutine calculates the downwash on every element due to the influence of all elements including ground effect image jet-wing elements. Onset flow perturbation velocities due to ground effect are also calculated. Each element is selected in turn, and the influence of all elements on it are computed one at a time by calling the appropriate downwash influence function: EVD1, EVD2, EVD3 or ground (singular part of Hinge EVD not yet considered). The coefficients are superimposed according to symmetry and stored on mass storage units 1 or 8, and the ground effect onset velocities are stored on unit 9.

Regular EVD Downwash (EVD1)

This function computes the downwash at any point in the jet-wing plane due to a regular triangular distribution of vorticity on an element located anywhere on the jet-wing [see figure (I.1a), Volume I]

Leading Edge EVD Downwash (EVD2)

This function computes the downwash at any point in the jet-wing plane due to a square root singular distribution of vorticity on an element located at the leading edge of the wing [see figure (I.1b), Volume I]

Jet Infinity EVD Downwash (EVD3)

This function computes the downwash at any point in the jet-wing plane due to a quadratic decaying distribution of vorticity on a trailing jet element. This element begins several chords behind the wing trailing edge and extends downstream to infinity, where the vorticity decays to zero.

Singular Part of Hinge EVD Downwash (EVD4)

This function computes the downwash at any point in the jet-wing plane due to a logarithmic distribution of vorticity on an element located on the hinge line of a deflected flap, jet, or leading edge flap [see figure (I.1c), Volume I]

Image Jet-Wing Perturbations (GROUND)

This subroutine computes the downwash and onset flow perturbation velocities at any point in the jet-wing plane due to a ground effect image jet-wing horseshoe vortex element.

Initial Matrix Augmentation (SHUFL1)

This subroutine reads from mass storage unit 1 or 8, the matrix rows corresponding to elements on the jet; augments the rows.

Additional Matrix Augmentation (SHUFL2)

This subroutine reads, from mass storage unit 1 or 8, the augmented matrix rows corresponding to elements on the jet; re-augments it according to the new jet strength values.

Column Matrix Formation (COLUM1)

This subroutine defines the right side column matrix of jet-wing boundary conditions corresponding to the input fundamental case deflections. Each column corresponds to one fundamental case.

Singular Part of Hinge EVD Downwash (HINGE)

This subroutine computes and sums the influence of the singular part of each Hinge EVD on all other elements of the jet-wing system.

Column Matrix Augmentation (COLUM2)

This subroutine adds the influence of the singular part of each Hinge EVD to the right side column matrix.

OVERLAY SEGMENT 3

Calculation of Jet-Wing Static Aerodynamic Characteristics

Main Program (STAGE3)

This program controls the sequence of calculations, as well as the type of aerodynamic characteristics which are to be calculated. For the basic condition (i.e., vectored thrust power off, out of ground effect) the jet-wing pressure loading is computed by calling STG3FC. The sectional and total aerodynamic coefficients for the fundamental cases are computed in STG3FS, and the sectional and total coefficients are computed in STG3C for all composite cases. For jet-wings in ground effect, the composite case sectional and total coefficients are computed in STG3GE. For wings with vectored jets, in or out of ground effect, the sectional and total coefficients for all composite cases are computed in STG3VT.

Jet-Wing Chordwise Loading (STG3FC)

The loading on each element is computed from the vorticity solution produced in the problem solution component program. If requested, the loading is printed, along with detailed expansions of the loading on all Leading Edge and Hinge EVD elements.

Wing Spanwise and Total Loading Control (STG3FS)

Spanwise variation of lift, pitching moment and induced drag are computed by calling SLOAD. Total induced drag is also computed by a momentum analysis by calling subroutines SLOADG and TREFTZ. Total lift, pitching moment, and rolling moment are computed in TLOAD, and total induced drag (by pressure integration) and yawing moment are computed by calling TLOADX. This sequence is repeated for each fundamental case.

Summary Table (STG3FT)

If requested, subroutine STG3FT prints a table of all total aerodynamic coefficients for all fundamental cases.

Initialized Data (BLKDTA)

The block data subroutine defines an array of label names for use in labeling the summary tables of total aerodynamic coefficients.

Singular EVD Loading Expansions (EXPLE, EXPH1, EXPH2)

The loading at five points on the EVD element is computed for the singular Leading Edge EVD and the leading and trailing parts of the singular Hinge EVD, respectively.

Wing Spanwise Loading (SLOAD)

This subroutine integrates the chordwise loading at each section to arrive at the sectional value of lift, induced drag and pitching moment. The loading for either fundamental or composite cases can be computed.

Wing Spanwise Cross-Product Loading (SLOADX)

The sectional cross-product values of induced drag are computed by chordwise integration of the loading of each element. This subroutine is utilized only in computation of composite cases.

Jet-Wing Spanwise Vorticity (SLOADG)

This subroutine integrates the chordwise loading at each section to get the total vorticity of the jet-wing system for use in calculation of induced drag by a momentum analysis. Integration is from the leading edge to the trailing edge for unblown sections, and from the leading edge to infinity for sections with a jet.

Wing Spanwise Loading (LOADG1)

This subroutine integrates the chordwise loading at each section to arrive at the sectional values of lift, induced drag and pitching moment. This routine applies for wings in ground effect with a jet sheet or vectored jets.

Wing Spanwise Cross-Product Loading (LOADG2)

The sectional cross-product values of induced drag are computed by chordwise integration of the pressure of each element, for wings in ground effect with a jet sheet or vectored jets.

Jet-Wing Spanwise Vorticity (GLOADG)

This subroutine is essentially the same as subroutine SLOADG except that this is used only for a jet-wing in ground effect.

Wing Loading (TLOAD)

This subroutine performs spanwise integration of the sectional values of lift, jet strength, and pitching moment to produce total lift, induced drag, jet strength, and pitching and rolling moments.

Wing Total Loading (TLOADO)

This subroutine computes total lift and pitching and rolling moment for all composite cases at zero angle of attack. The coefficients are computed by summing the fundamental case coefficients, each multiplied by its respective input scale factor.

Wing Total Loading (TLOADX)

This subroutine computes total induced drag (both momentum and wing pressure integral methods) and yawing moment coefficients at zero angle of attack by spanwise integration of the appropriate sectional data.

Vectored Thrust Reaction (TLODVT)

This subroutine computes the contributions of all vectored jets to all wing force and moment coefficients.

Trefftz Plane Downwash (TREFTZ)

This subroutine computes the induced downwash at the Trefftz Plane (infinity) due to all the loading of the complete jet-wing system, for either fundamental or composite cases. These data are used in the momentum induced drag method.

Quadrature Integration (QUA)

This subroutine computes certain integrals required in calculation of induced drag in ground effect by the momentum method.

Wing Composite Case Loading Control (STG3C)

This program controls the calculation of all jet-wing loads for composite cases with no ground effect or vectored thrust. The composite chordwise loading is first computed by superimposing the loading from the required fundamental cases, each multiplied by its respective input scale factor. For composite cases, both the loading at zero angle of attack and the linear variation with angle of attack must be computed. For the nonlinear coefficients, the variation with angle of attack becomes quadratic, and a new "cross-product" term, linear in angle of attack, is also required. The computation of all spanwise and total coefficients corresponding to the zero, linear, quadratic, and cross-product angle of attack terms are computed by calling the various loading utility subroutines. Finally, subroutine TABLE is called to give a variation of total lift, drag, and moments with angle of attack. The entire sequence of computations above is repeated for each composite case requested.

Composite Case Summary (TABLE)

This subroutine computes and prints a table of the variation of lift, induced drag (momentum method), and pitching, yawing, and rolling moments with angle of attack. The printout is given for each composite case.

Wing Loading Control in Ground Effect (STG3GE)

This program controls calculation of all jet-wing loads for each composite case in ground effect with no vectored thrust. The chordwise pressure coefficients are first computed by combination of the vorticity and the onset flow perturbation velocities induced by the image jet-wing. Next, the leading edge suction is computed; and sectional lift, pitching moment and induced drag are computed by calling LOADG1, LOADG2, and GLOADG. The computation of all total static aerodynamic coefficients is accomplished by calling TLOAD, TLOADX, and TREFTZ. Note that the non-linear ground effect requires "cross-product" and squared terms for lift and pitching moment as well as induced drag. Finally, subroutine TABLGE is called to give a variation of the total coefficients with angle of attack.

Composite Case Summary (TABLGE)

This subroutine calculates and prints a table of the variation of lift;

induced drag (momentum method); and pitching, rolling, and yawing moments with angle of attack for jet-wing composite cases in ground effect with no vectored thrust.

Wing Loading Control with Vectored Thrust (STG3VT)

This program controls calculation of all wing loads for each composite case for wings with vectored jets in or out of ground effect. The chordwise pressure coefficients are first computed with or without the ground influence. Next, the sectional lift, induced drag and pitching moments are computed in LOADG1. Finally, the total lift; induced drag; and pitching, rolling, and yawing moments are computed by calling GLOADG, TREFTZ, TLOAD, TLOADX, and TLODVT. This sequence is repeated for angles of attack of 0, 5, 10, 15, and 20 degrees.

OVERLAY SEGMENT 4

Dynamic Stability Derivatives - Utility Routines

Main Program (STAGE4)

This program controls the execution of the utility routines which define the quasi-steady jet-wing boundary conditions arising from rates of pitch, roll, and yaw. During the basic run, the pitching rate boundary conditions are defined in BCPICH. During the second dynamic stability run, the rolling and yawing rate boundary conditions are defined in BCROLL and BCYAW, respectively. Certain jet-wing pressure coefficients and incidence angles may also be stored or retrieved as required.

Geometric Angle Storage (SAVEIT)

This subroutine stores on unit 1 some of the jet-wing geometric incidence angles which are to be altered during a second dynamic stability derivative run.

Geometric Angle Retrieval (RETREV)

This subroutine reads back from unit 1 the jet-wing incidence angles which were previously saved by SAVEIT.

Pitching Rate Induced Camber (BCPICH)

This subroutine defines the induced camber angles on all wing and jet EVD elements, which result from simulation of the wing pitching about the input center of gravity location.

Rolling Rate Induced Camber (BCROLL)

This subroutine defines the induced camber angles on all wing and jet EVD elements, which result from simulation of the wing rolling about the x-axis.

Yawing Rate Induced Camber (BCYAW)

This subroutine defines the induced camber angles on all wing and jet EVD elements, which result from simulation of the wing yawing about the z-axis.

OVERLAY SEGMENT 5

Jet-Wing Flow Field

Main Program (STAGE5)

This program controls the calculation of perturbation velocities induced by the jet-wing for each composite case. First, the jet-wing composite case vorticity and deflection are summed for each EVD element. Then the angle of attack and thickness contributions to jet-wing vorticity are added, the equivalent horseshoe vortex element strengths are computed, and the equivalent non-planar jet-wing geometry is computed. These operations are performed by calls to LINEAR, VORTEX, TRAIL, and COORS. Next, the vectored jet induced velocities, if any, are retrieved and all jet-wing induced velocities are computed in INDVEL. Finally, the flow field influence on tail camber are defined in RHS1. The procedure is repeated for each of the five angles of attack of 0, 5, 10, 15, and 20 degrees.

Linear Jet-Wing Vorticity (LINEAR)

This subroutine adds the jet-wing vorticity due to angle of attack to the composite vorticity at zero angle of attack, and scales up the vorticity according to the thickness/chord ratio.

Integration of Bound Vorticity (VORTEX)

This subroutines integrates the distributed bound vorticity of each EVD element to get the strength of an equivalent discrete bound vortex filament.

Integration of Trailing Vorticity (TRAIL)

This subroutine integrates the distributed trailing voriticity of each EVD element to get the strength of the equivalent discrete trailing vortex filaments.

Horseshoe Vortex Geometry (COORS)

This subroutine computes the x, y, and z coordinates and incidence angles of the discrete horseshoe vortex elements of the equivalent non-planar jet-wing system.

Induced Velocities (INDVEL)

This subroutine computes the perturbation velocities induced by the equivalent jet-wing horseshoe vortex system. Velocities at control points on the fuselage, tail, and input off-body points are computed, and vectored jet-induced velocities are included.

Biot-Savart Law (BSLAW)

This subroutine computes the velocity induced at any flow field point by a discrete horseshoe vortex element and its images about the $y = 0$ plane and the ground plane.

Tail Boundary Conditions (RHS1)

This subroutine defines the right side column matrix, for the tail, resulting from the jet-wing and vectored jet induced velocities. The flow field effect is treated as effective tail camber.

OVERLAY SEGMENT 6

Horizontal and Vertical Tail Analysis

This program controls calculation of all geometry, downwash, sidewash, and aerodynamic loading on the horizontal and vertical tail. There are six different phases of execution, which are dependent on the requirements of the main program. These include: (1) Calculation of the horizontal and vertical tail planform parameters, printing the geometry, calculation of the size and location of each horseshoe vortex element, and calculation of the influence coefficients for the isolated tail, by calling TAILG, TAILP, LATICE, and LHS, respectively; (2) Calling RHS2 to set up the right side column matrix of tail static and dynamic boundary conditions; (3) Calling TAILL to compute the loading of the isolated tail, the loading of the tail in the jet-wing flow field, or the special (anti-symmetric) dynamic stability derivatives due to rate of roll.

Tail Planform Geometry (TAILG)

This subroutine computes all the planform geometric parameters for trapezoidal horizontal and vertical tail surfaces.

Tail Planform Geometry Print (TAILP)

This subroutine prints a summary of the horizontal and vertical tail planform geometric parameters.

Tail Element Geometric Parameters (LATICE)

This subroutine computes the size and location of each discrete vortex element on the horizontal and vertical tail, as well as other parameters for later internal use by the program.

Tail Influence Coefficients (LHS)

This subroutine computes the left hand side matrix of downwash and sidewash influence coefficients for the horizontal-vertical tail combination. The matrix is stored row by row on mass storage unit 8.

Tail Column Matrix (RHS2)

This subroutine defines the right side column matrix of static and dynamic tail boundary conditions for both the isolated tail and the tail in the jet-wing flow field.

Tail Loading (TAILL)

This subroutine first reads the self-induced normal-wash influence coefficients and sums up the total normal-wash at the bound vortex elements. This information is later used for calculating tail induced drag. Next, the chordwise pressure loading is computed. Finally, all the spanwise and total static and dynamic aerodynamic coefficients are computed and printed.

OVERLAY SEGMENT 7

Vectored Jet Interferences

Main Program (STAGE7)

This program controls calculation of the vectored jet envelopes and their induced flow field velocities according to the method of Wooler. The basic control flags and certain jet parameters are first initialized. Next, the flow field coordinates are defined for which induced velocities are required. The input jet parameters are arranged, the jet envelopes are computed by calling CNTRLN, and the jet centerline shape is printed. Next, the induced perturbation velocities are computed by calling JETVEL and the velocities induced by all jets are summed. This is repeated for all jets, and finally, the total perturbation velocities are saved on unit 8. The entire sequence is repeated for angles of attack of 0, 5, 10, 15, and 20 degrees.

Vectored Jet Envelope (CNTRLN)

This program controls computation of the centerline shape and cross-sectioned size of each jet.

Jet Source and Doublet Strength (MBMU)

This subroutine computes the strengths of the sources, sinks and doublets at each segment of a jet. The cross-sectional dimensions of the jet envelope are also computed.

Initial Predictor-Corrector Values (RKUTTA)

This subroutine controls calculation of the initial values for the Adams predictor-corrector scheme to calculate the jet centerline.

Adams Predictor-Corrector Method (ADAM3)

This subroutine solves the Adams predictor-corrector equation for estimating the location of each segment of the jet centerline.

Runge-Kutta Solution (DRK1)

This subroutine uses a Runge-Kutta fourth-order method for solution of first order simultaneous ordinary differential equations.

Equations of Motion (MOTION)

This subroutine solves the equations of fluid motion for use in computing the location of the jet centerline.

Flow Field Point Test (PTEST)

This subroutine checks for any flow field point falling inside the jet envelope. If such a point is found, a warning message is printed, but no corrective action is taken.

Vectored Jet-Induced Velocities (JETVEL)

This program calculates the induced velocities at all required flow field points due to the sources, sinks, and doublets on each segment of a jet.

OVERLAY SEGMENT 8

Matrix Quasi-Inverse Solution

Main Program (STAGE8)

This program simply calls STG8W or STAG8T for solution of the jet-wing or tail solution matrices, respectively.

Matrix Quasi-Inverse Solution (MATRIX)

This subroutine solves a matrix system of arbitrary dimension with an arbitrary number of right sides by a Gaussian triangularization method. At intermediate points in the solution, the triangular matrices are stored on scratch files. This information may later be used to rapidly arrive at a solution for arbitrary new right sides, thus giving the method a "quasi-inverse" capability. The method can solve large matrix systems because only a small portion of the matrix is treated at one time, the remainder being stored on scratch files.

Back-Substitution Check (BAKSUB)

This subroutine multiplies the left-side jet-wing influence coefficient matrix by the vorticity solution to arrive at the right side column matrix. This result is printed as a check of the solution accuracy, and should correspond exactly to original fundamental case incidence angles in radians.

Jet-Wing Matrix Solution (STG8W)

This program controls the quasi-inverse solution of the jet-wing matrix system. The left-side influence coefficient matrix is assembled in the proper form by calling PREPW, the quasi-inverse solution is computed by MATRIX, and the solution accuracy is checked, if required, by calling BAKSUB.

Jet-Wing Matrix Preparation (PREPW)

This subroutine retrieves the jet-wing influence coefficients from storage units 1, 8, or 9, and writes them in a row-by-row sequence required for input to the solution routine.

Tail Matrix Solution (STG8T)

This program controls the quasi-inverse solution of the tail matrix system. The left-side influence coefficient matrix is assembled in the proper form by calling PREPW and the quasi-inverse solution is computed by MATRIX.

Tail Matrix Preparation (PREPT)

This subroutine retrieves the tail influence coefficients from storage on unit 8 and writes them in a row-by-row sequence required for input to the solution routine.

OVERLAY SEGMENT 9

Matrix Quasi-Inverse Back-Solution

Main Program (STAGE9)

This program simply calls STG9W or STG9T for back-solution of the jet-wing or tail matrices, respectively.

Matrix Quasi-Inverse Back-Solution (NEWSOL)

This subroutine back-solves a matrix system for a new right side column matrix, using the data previously stored on scratch tapes by subroutine MATRIX during the first solution.

Jet-Wing Back-Solution Control (STG9W)

This program controls the jet-wing matrix system back-solution for a new right side column matrix. The right side boundary conditions are the equivalent induced camber angles on the jet-wing due to the vectored jets.

Tail Back-Solution Control (STG9T)

This program controls the tail matrix system back-solution for a new right side column matrix. The right side boundary conditions are the equivalent induced camber angles on the tail due to the vectored jets.

OVERLAY SEGMENT 10

Fuselage Analysis

Main Program (STAG10)

This program controls the approximate slender body fuselage analysis. The fuselage geometry is defined by calling FUSELG and the axial perturbations are computed by AXIAL. Next, for each fuselage segment, the cross-sectional flow perturbations are computed. For non-circular cross-sections, the element angles are computed by ANGLES, and the two-dimensional cross-flow velocities are computed in NEUMANN. Then the velocities are combined to form the pressure coefficients on each segment by calling CPRES, and these pressures are integrated in SUMER1 to get the sectional pitching and yawing moments. Finally, SUMER2 integrates all the segments to get total pitching and yawing moments.

Fuselage Geometry (FUSELG)

This subroutine computes all required geometry parameters from the input data, for circular, elliptical, and arbitrary cross-sections.

Axial Flow Analysis (AXIAL)

This subroutine computes the axial and radial perturbation velocities arising from the fuselage axial area distribution.

Sectional Geometry (ANGLES)

This subroutine calculates the dimensions and angular orientation of each element of an elliptical or arbitrary fuselage cross-section.

Cross-Flow Analysis (NEUMANN)

This subroutine computes the cross-flow velocities on a fuselage segment with elliptical or arbitrary cross-section. Each segment is treated as an isolated two-dimensional body.

Sectional Pressure Distribution (CPRES)

This subroutine combines the axial and cross-sectional perturbation velocities on each element of a segment to find the pressure distribution on the segment.

Sectional Moments (SUMER1)

This subroutine integrates the sectional pressure distribution to find the contributions of the segment to fuselage pitching and yawing moments.

Total Fuselage Moments (SUMER2)

This subroutine integrates the sectional moment contributions to get the total fuselage pitching and yawing moments.

OVERLAY SEGMENT 11

Jet-Wing Dynamic Stability Derivatives

Main Program (STAG11)

This program controls calculation of jet-wing dynamic stability derivatives due to pitching, rolling, and yawing rates. For jet-wings in ground effect, all derivatives are computed by calling GEDER. For jet-wings not in ground effect, fundamental case derivatives are computed by calling FUNDER and composite case derivatives by calling COMDER.

Fundamental Case Dynamic Stability Derivatives (FUNDER)

This subroutine controls calculation of the rolling and yawing dynamic stability derivatives for all fundamental cases, for jet-wings not in ground effect. (The pitching rate derivatives were previously computed in STG3FS). The pressure coefficients from the basic run are retrieved from storage unit 1 and the aerodynamic coefficients are computed by calling SUMIT1, SUMIT2, and SUMIT3.

Composite Case Dynamic Stability Derivatives (COMDER)

This subroutine controls calculation of the rolling and yawing dynamic stability derivatives for all composite cases, for jet-wings not in ground effect. The pressure coefficients from the basic run are retrieved from storage unit 1 and the aerodynamic coefficients are computed by calling SUMIT1 and SUMIT2. Finally, a table is printed by calling STABLE showing the components of the stability derivatives and the variation of the derivatives with angle of attack.

Composite Case Dynamic Stability Derivatives (GEDER)

This subroutine calculates the pitching, rolling, and yawing dynamic stability derivatives for all composite cases, for jet-wings in ground effect. The vorticity and onset perturbation velocities from the basic run are retrieved from storage units 1 and 9 and combined to get the pressure coefficients. The aerodynamic coefficients are then computed by calling SUMIT3 and SUMIT4.

Dynamic Stability Derivatives (SUMIT1)

This subroutine computes the dynamic stability derivatives of yawing moment due to rolling, and rolling moment due to yawing for jet-wings not in ground effect.

Dynamic Stability Derivatives (SUMIT2)

This subroutine computes the dynamic stability derivatives of yawing moment due to yawing for jet-wings not in ground effect.

Dynamic Stability Derivatives (SUMIT3)

This subroutine computes the dynamic stability derivatives due to pitching rate for jet-wings in ground effect.

Dynamic Stability Derivatives (SUMIT4)

This subroutine computes the dynamic stability derivatives due to rolling and yawing rates for jet-wings in ground effect.

Dynamic Stability Derivative Table (STABLE)

This subroutine prints, for jet-wing composite case not in ground effect, a summary of all dynamic stability derivatives and the terms used to compute each. An angle of attack table is then computed and printed, containing all the stability derivatives which depend on angle of attack.

OVERLAY SEGMENT 12

Aircraft Aerodynamic Coefficient Summary

Main Program (STAG12)

This program sums and prints tables of the variation with angle of attack of the aerodynamic characteristics of the jet-wing, horizontal tail, vertical tail, and fuselage. The basic coefficients have previously been stored on unit 9 at various points during the program execution. Total aircraft characteristics are also summed and printed.

APPENDIX II

EVD FUNCTIONS - NUMERICAL RESTRICTIONS

Because of the characteristic of all digital computers of defining numbers to only a limited number of significant digits, it sometimes happens that equations cannot be evaluated with the required accuracy. This is particularly a problem where the difference of two numbers of nearly the same magnitude is taken. As the two numbers approach each other, the difference loses accuracy, and if they are identical (to the number of digits the computer can hold), the difference loses all accuracy.

For the four EVD downwash influence functions (see Appendix I.1 of Volume I), the above type of problem has been encountered in their evaluation when the control point is in the far-field or, for certain unique locations, in the near-field region of any EVD element. These can generally occur along the leading and trailing edges of any EVD element, and at the apex of a Regular EVD.

The following restrictions and approximations have been adopted, without loss of generality, in order to prevent random inaccuracies from affecting the results. In the case of far-field cutoff, the true limiting values of each function are used, and the only approximations are that these limiting values are used somewhat "closer" than real infinity, where they correctly apply. For near-field cutoff, the limiting values are used in small regions near the points where the values correctly apply. The subscript i refers to the control point at which the downwash is being computed, due to the vorticity of element j.

EVD Influence Function	Region	Value Used
Regular*	$\frac{x_i-x_j}{\frac{1}{2}\left(\delta_j+\delta_{j-1}\right)} > 100$	$a\left(\infty,\left(y_i-y_j\right)\right) = -\frac{1}{4\pi}\left(\delta_j+\delta_{j-1}\right)\left(\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right)$
	$x_i-x_j = 0,\ y=0$	Not encountered

*Volume I: Appendix I.1.1

EVD Influence Function	Region	Value Used
	$x_i-x_j = -\delta_{j-1}, y=0$	Not encountered
	$x_i-x_j = \delta_j, y=0$	Not encountered
Leading* Edge	$(x_i-x_j)/\delta_j > 100$	$a\left(\infty,(y_i-y_j)\right) = -\frac{1}{2\pi}\left(\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right)$
	$\left.\begin{array}{l}\lvert(x_i-x_j)/\delta_j\rvert < 10^{-4}\\ \lvert[(x_i-x_j)/\delta_j]-1\rvert < 10^{-6}\end{array}\right\}$	$a\left((x_i-x_j),(y_i-y_j)\right) = a\left(0,(y_i-y_j)\right)$
Infinity+	$\left(\frac{x_i-x_j+d}{y_i-y_j-\Delta}\right) > 10^6$	$a\left((x_i-x_j),(y_i-y_j)\right) = -\frac{1}{2\pi}\left(\frac{\delta_{j-1}}{2}+d\right)\left(\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right)$
	$\left\lvert\frac{x_i-x_j+d}{d}\right\rvert < 10^{-2}$	$a\left((x_i-x_j),(y_i-y_j)\right) = a\left(-d,(y_i-y_j)\right)$
Hinge‡ (Singular Part)	$\left\lvert\frac{x_i-x_j}{\frac{1}{2}(\delta_j+\delta_{j-1})}\right\rvert > 7.5$	$b\left((x_i-x_j),(y_i-y_j)\right) = \frac{\beta}{\pi^2}\left[-(\delta_{j-1}+d_j) - (\delta_{j-1}\log\delta_{j-1}+\delta_j\log\delta_j)\right]\left[\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right]$
	$\left\lvert\frac{x_i-x_j}{\frac{1}{2}(\delta_j+\delta_{j-1})}\right\rvert < 10^{-4}$	$b\left((x_i-x_j),(y_i-y_j)\right) = b\left(0,(y_i-y_j)\right)$
	$\left\lvert\frac{x_i-x_j+\delta_{j-1}}{\delta_{j-1}}\right\rvert < 10^{-6}$	$b\left((x_i-x_j),(y_i-y_j)\right) = b\left(-\delta_{j-1},(y_i-y_j)\right)$
	$\left\lvert\frac{x_i-x_j-\delta_j}{\delta_j}\right\rvert < 10^{-6}$	$b\left((x_i-x_j),(y_i-y_j)\right) = b\left(\delta_j,(y_i-y_j)\right)$

* Volume I: Appendix I.1.2
\+ Volume I: Appendix I.1.4
‡ Volume I: Appendix I.1.3

EVD Influence Function	Region	Value Used
	$\left\lvert \frac{\delta_{j-1}}{\lvert x_i - x_j \rvert} - 1 \right\rvert < 10^{-4}$	$b\left((x_i - x_j), (y_i - y_j)\right) = b\left(\delta_{j-1}, (y_i - y_j)\right)$
	$\left\lvert \frac{\delta_j}{\lvert x_i - x_j \rvert} - 1 \right\rvert < 10^{-4}$	$b\left((x_i - x_j), (y_i - y_j)\right) = b\left(\delta_j, (y_i - y_j)\right)$
Hinge‡ (Singular Part, cont'd)	$\left\lvert \frac{\delta_j - x_i - x_j}{\delta_j} \right\rvert < 10^{-6}$	$b\left((x_i - x_j), (y_i - y_j)\right) = b\left(\delta_j, (y_i - y_j)\right)$
	$\left\lvert \frac{x_i - x_j + \delta_{j-1}}{\delta_{j-1}} \right\rvert < 10^{-6}$	$b\left((x_i - x_j), (y_i - y_j)\right) = b\left(-\delta_{j-1}, (y_i - y_j)\right)$

Note: The value of the Hinge EVD regular part is, in all cases, equivalent to a Regular EVD.

‡ Volume I: Appendix I.1.3

APPENDIX III

ERROR MESSAGES

In order to avoid wasting computing time, some types of errors or inconsistencies in input data can be checked by the program before beginning computation. Sometimes such errors make the program unable to continue execution, and sometimes the program can recover by ignoring the data or substituting correct data. In any case, an error message will be printed, enabling the user to identify the type of error or warning him that the input data may have been altered. Below is a list of the messages which may be printed and the subroutines in which they occur. At the end of each run a status message is always printed indicating whether the program has reached a normal or abnormal conclusion.

Message	Subroutine
** * THE PROGRAM HAS REACHED NORMAL TERMINATION * **	MAIN
******** ** * THE PROGRAM HAS REACHED ABNORMAL TERMINATION * ** ********	MAIN
AN INCONSISTENCY HAS BEEN FOUND IN THE GENERAL CONTROL FLAGS PLEASE RECHECK YOUR INPUT ON THE GENERAL CONTROL CARD	MAIN
NUMBER OF WING ROW TYPES ****	INPTS
*** WING ELEMENTS PRESCRIBED FOR ROW TYPE***	INPTS
NUMBER OF JET ROW TYPES ****	INPUTJ
*** JET ELEMENTS PRESCRIBED FOR ROW TYPE***	INPUTJ
3 ROW CONTINUITY RULE FAILURE	INPUTJ
AN INCONSISTENCY HAS BEEN FOUND IN THE SECTIONAL LEADING AND TRAILING EDGE INPUT	XLETRI
PLEASE CHECK YOUR SECTION LOCATION (Y) INPUT	BOXS
**** IS TOO MANY ELEMENTS	BOXS
A ZERO VALUE OF CMU HAS BEEN INPUT. THIS CMU CASE HAS BEEN IGNORED.	BOXJ
FUNDAMENTAL GEOMETRIC CASE*** AN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT DATA FOR WING ROW***, ROW TYPE***	BEECEE
AN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN FOUND. IT WILL BE IGNORED.	INCOMP
MORE THAN 24 COMPOSITE CASES HAVE BEEN INPUT. SUBSEQUENT INPUTS WILL BE IGNORED.	INCOMP
AN END OF FILE HAS BEEN READ DURING COMPOSITE CASE INPUT	INCOMP

Message	Subroutine
NUMBER OF TAIL ROW TYPES ****	INTAIL
NUMBER OF TAIL CAMBER TYPES ****	INTAIL
THERE IS AN ERROR IN HORIZONTAL TAIL Y INPUTS	INTAIL
THERE IS AN ERROR IN VERTICAL TAIL Z INPUTS	INTAIL
NO LAST FLAG FOUND WHILE READING CIRCULAR JET INPUT	INJETS
THTA ANGLE MUST BE IN 1 TO 179 OR -1 TO -179 DEGREE RANGE	INJETS
** WARNING ** ELEMENT**** HAS AN ITYPE VALUE OF*** AN EQUIVALENT TRIANGULAR DOWNWASH WAS USED	DWNWSH
0 0.0 0.0 0.0 CONTROL POINT IS ON BOUND VORTEX ELEMENT NUMBER 5	INDVEL
0 0.0 0.0 0.0 CONTROL POINT IS ON TRAILING VORTEX ELEMENT NUMBER 5	INDVEL
***** ERROR IN CALL TO ADAMS. NFUNCT = 5. PROGRAM HALT.	ADAMS
***** ERROR IN CALL TO ADAMS1. NFUNCT = 5. PROGRAM HALT.	ADAMS
***** WARNING ***** THE FOLLOWING**** FLOW FIELD POINTS FELL INSIDE THE JET ENVELOPE THE INDUCED VELOCITIES AT THESE POINTS ARE IN ERROR	JETVEL
NOT ENOUGH ROOM WAS RESERVED FOR MATRIX	STG8W
NOT ENOUGH ROOM WAS RESERVED FOR NEWSOL	STG9W
*** NO NEW RIGHT SIDES WERE SUPPLIED FOR SOLUTION BY EXISTING QUASI-INVERSE MATRIX ***	NEWSOL
*** AN INPUT ERROR HAS CAUSED TERMINATION IN SUBROUTINE ANGLES ***	ANGLES
*** AN INPUT ERROR HAS CAUSED TERMINATION IN SUBROUTINE CPRES ***	CPRES

APPENDIX IV

SAMPLE PROBLEMS

The input card images and program printout for two sample problems are given below. The first case is for a jet-wing with a fuselage and tail. The second case is for a wing with two vectored jets and a fuselage and tail.

It must be noted that these cases have been prepared only as brief examples of program input and output. No attempt has been made to represent the wing realistically in terms of element spacing. On the contrary, the crudest possible spacing was used in order to minimize computing time and printed output volume.

Since the element spacing is very rough, it should not be used as a model for preparation of other cases. Because of the crude spacing and few elements, the results shown will not accurately predict the aerodynamic characteristics of the aircraft. The fundamental and composite case capabilities of the program are far greater than those demonstrated by these cases, and the ground effect and dynamic stability derivatives and ground effects have not even been included. It is recommended that these sample cases be used as the first verification test cases any time the program is used at a new facility.

INPUT CARD IMAGES

```
See
Sect.

5.1     { *** STAMP SAMPLE CASE *  RECT WING + TAIL + FUSELAGE + JET SHEET CJ = 1.0  ***
        {  3.20        4.00        0.80        0.20        0.40        0.10        0.0
        {  4 3 0 0 0 2 0 0 0 0   3 4 1 2 1 4 2 1
        {  0.950       0.800       0.550       0.200
        {  1 1 1 1
        {  3
        {  0.00        0.200       0.750
        {  5.00        0.00        1.00
        {  1 1 1 1
        {  3 3 3 3
5.2     {  1.00        1.250       2.00
        {  0 0 0 0 1
        {  0 1 1 0
        {  0.750     0 1.00
        {  0 0 1 0 0
        {  1.00        1.00        1.00        1.00
        {  1 0.00  2 30.0  3 10.0
        { 9
        {  2.6666      1.00        0.00        1.600       2.00        0.600
5.3.1   {  0.8750      0.3750     -0.3750     -0.8750
        {  1 1 1 1
        {  1 1 1 1
        {  1.3333      1.00        0.00        0.800       2.00        0.00
5.3.2   {  0.8750      0.3750
        {  1 1
        {  1 1
        {  2
5.3.3   {  2
        {  0.00        0.50
        {  0.00        10.00
        { -1.00       -0.90       -0.60        2.00        2.60
5.4     { -0.150      -0.150      -0.150      -0.150      -0.150      -0.150
        {  4 4 4 4
        {  0.00        0.20        0.30        0.30        0.00
        { -1.000       0.250       0.250
5.5     {  0.0         0.250       0.250
        {  1.000       0.250       0.250
5.6.1   {  1.00        1.00        1.00        1.00
        { 9
```

```
5.1     { *** STAMP SAMPLE CASE *  RECT WING + TAIL + FUSELAGE + VECTORED JETS  ***
        {  3.20        4.00        0.80        0.20        0.40        0.10        0.0
        {  4 2 0 0 2 2 0 0 0 0   3 4 1 2 1 4 2 1
        {  0.950       0.800       0.550       0.200
        {  1 1 1 1
        {  3
        {  0.00        0.200       0.750
5.2     {  5.00        0.00        1.00
        {  0 0 0 0 1
        {  0 1 1 0
        {  0.750     0 1.00
        {  1 0.00  2 30.0
        { 9
        {  2.6666      1.00        0.00        1.600       2.00        0.600
5.3.1   {  0.8750      0.3750     -0.3750     -0.8750
        {  1 1 1 1
        {  1 1 1 1
        {  1.3333      1.00        0.00        0.800       2.00        0.00
5.3.2   {  0.8750      0.3750
        {  1 1
        {  1 1
        {  2
5.3.3   {  2
        {  0.00        0.50
        {  0.00        10.00
        { -1.00       -0.90       -0.60        2.00        2.60
5.4     { -0.150      -0.150      -0.150      -0.150      -0.150      -0.150
        {  4 4 4 4
        {  0.00        0.20        0.30        0.30        0.00
        { -1.000       0.250       0.250
5.5     {  0.0         0.250       0.250
        {  1.000       0.250       0.250
        {  0.50        0.10        40.0        0.0         0.20        0.250      -0.10        0.0    001
5.6.2   { 9
```

SAMPLE PROBLEM I
JET SHEET CASE

```
**********************************************
*  STOL  AERODYNAMIC  METHODS  PROGRAM  *
**********************************************

*** STAMP SAMPLE CASE *  RECT WING + TAIL + FUSELAGE + JET SHEET CJ = 1.0  ***

                   USED            INPUT
      AREA =       0.800000        3.200000
      SPAN =       2.000000        4.000000
      CREF =       0.400000        0.800000
       XMC =       0.100000        0.200000
      CMAC =       0.400000        0.799999
    ARATIO =       5.000001        5.000001
       XCG =       0.200000        0.400000

          NROWS  =  4          4
          NCASES =  3          3
          ISYMM  =  0          0
          IPRINT =  0          0
          JETFLG =  0          0
          IGTYPE =  2          2
          IHINGE =  0          0
          IDERIV =  0          0
          IGRND  =  0          0
          IFFLD  =  0          0
          NPOINT =  3          3
          NROWH  =  4          4
          ICAMH  =  1          1
          NROWV  =  2          2
          ICAMV  =  1          1
          NSEG   =  4          4
          ISECT  =  2          2
          ICAMF  =  1          1

          NUMBER OF WING ELEMENTS  =  12
          NUMBER OF JET ELEMENTS   =  12
          TOTAL NUMBER OF ELEMENTS =  24

**************************************************************************
*  ELEMENT GEOMETRY AND FUNDAMENTAL CASE DATA FOR ALL FUNDAMENTAL CASES  *
**************************************************************************

FUNDAMENTAL CASE  1 =   ALPHA CASE

FUNDAMENTAL CASE  2 =   T.E. HINGE

FUNDAMENTAL CASE  3 =   JET TURNING


*** SECTION  1 ***  Y =  0.950000  DELTA =  0.050000  XLEAD =  0.0       XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                                          ..........................................INPUT VALUES..........................................
                                       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
                             TWIST(DEG)  0.0       0.0       0.0
                             DJ   (DEG)  0.0       0.0       1.00000
                             HL   (Z/C)  0.0       0.0       0.0
WING ELEMENTS                             ...............................................EPS..............................................
NO   X/C          X       DEL       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
 1  0.0          0.0     0.20000   1.00000   0.0       0.0
 2  0.20000      0.08000 0.55000   1.00000   0.0       0.0
 3  0.75000      0.30000 0.25000   1.00000   0.0       0.0
JET ELEMENTS                              ..............................................THETA.............................................
NO   X/C          X       DEL       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
13  1.00000      0.40000 0.25000   1.00000   0.0       1.00000
14  1.25000      0.50000 0.75000
15  2.00000      0.80000 *******

*** SECTION  2 ***  Y =  0.800000  DELTA =  0.100000  XLEAD =  0.0       XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                                          ..........................................INPUT VALUES..........................................
                                       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
                             TWIST(DEG)  0.0       0.0       0.0
                             DJ   (DEG)  0.0       0.0       1.00000
                             HL   (Z/C)  0.0       0.0       0.0
WING ELEMENTS                             ...............................................EPS..............................................
NO   X/C          X       DEL       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
 4  0.0          0.0     0.20000   1.00000   0.0       0.0
 5  0.20000      0.08000 0.55000   1.00000   0.0       0.0
 6  0.75000      0.30000 0.25000   1.00000   1.00000   0.0
JET ELEMENTS                              ..............................................THETA.............................................
NO   X/C          X       DEL       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
16  1.00000      0.40000 0.25000   1.00000   1.00000   1.00000
17  1.25000      0.50000 0.75000
18  2.00000      0.80000 *******

*** SECTION  3 ***  Y =  0.550000  DELTA =  0.150000  XLEAD =  0.0       XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                                          ..........................................INPUT VALUES..........................................
                                       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
                             TWIST(DEG)  0.0       0.0       0.0
                             DJ   (DEG)  0.0       0.0       1.00000
                             HL   (Z/C)  0.0       0.0       0.0
WING ELEMENTS                             ...............................................EPS..............................................
NO   X/C          X       DEL       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
 7  0.0          0.0     0.20000   1.00000   0.0       0.0
 8  0.20000      0.08000 0.55000   1.00000   0.0       0.0
 9  0.75000      0.30000 0.25000   1.00000   1.00000   0.0
JET ELEMENTS                              ..............................................THETA.............................................
NO   X/C          X       DEL       FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
19  1.00000      0.40000 0.25000   1.00000   1.00000   1.00000
20  1.25000      0.50000 0.75000
21  2.00000      0.80000 *******
```

```
*** SECTION  4 ***  Y =  0.200000  DELTA =  0.200000  XLEAD =  0.0       XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                                    ............................................INPUT VALUES.........................................
                                      FC1        FC2        FC3        FC4        FC5        FC6        FC7        FC8        FC9        FC10
                        TWIST(DEG)   0.0        0.0        0.0
                        DJ   (DEG)   0.0        0.0        1.00000
                        HL   (Z/C)   0.0        0.0        0.0
WING ELEMENTS                       ..................................... ..........EPS.............................................
NO   X/C        X        DEL         FC1        FC2        FC3        FC4        FC5        FC6        FC7        FC8        FC9        FC10
10  0.0        0.0      0.20000     1.00000    0.0        0.0
11  0.20000    0.08000 0.55000      1.00000    0.0        0.0
12  0.75000    0.30000 0.25000      1.00000    0.0        0.0
JET ELEMENTS                        ..............................................THETA...........................................
NO   X/C        X        DEL         FC1        FC2        FC3        FC4        FC5        FC6        FC7        FC8        FC9        FC10
22  1.00000    0.40000 0.25000      1.00000    0.0        1.00000
23  1.25000    0.90000 0.75000
24  2.00000    0.80000 *******

 * STOL AERODYNAMIC METHODS PROGRAM *                    * DOUGLAS AIRCRAFT COMPANY *                        * DATE 11/21/73  *  TIME 16.16.59
 CMU CASE NUMBER  1                                           NO GROUND EFFECT                                      NO STABILITY DERIVATIVES


                                          ****************************************
                                          *  SECTIONAL JET BLOWING COEFFICIENTS  *
                                          ****************************************

                                                    ROW     CMU
                                                     1     1.000000
                                                     2     1.000000
                                                     3     1.000000
                                                     4     1.000000

* STOL AERODYNAMIC METHODS PROGRAM *                    * DOUGLAS AIRCRAFT COMPANY *                        * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                           NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                      ****************************************************
                                      *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                      ****************************************************

                                    SECTION  1     Y =  0.950000     CHORD =  0.400000
WING    I       XB         CASE 1    CASE 2    CASE 3    CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
        1   0.0          0.110625  0.016047  0.013449  0.0        0.0        0.0        0.0        0.0        0.0        0.0
        2   0.200000     0.046723  0.013141  0.009106  0.0        0.0        0.0        0.0        0.0        0.0        0.0
        3   0.750000     0.012297  0.017058  0.011973  0.0        0.0        0.0        0.0        0.0        0.0        0.0
 JET
       13   1.000000     0.006646  0.008015  0.046870  0.0        0.0        0.0        0.0        0.0        0.0        0.0
       14   1.250000     0.002609  0.002424  0.003426  0.0        0.0        0.0        0.0        0.0        0.0        0.0
       15   2.000000     0.001082  0.000550  0.000783  0.0        0.0        0.0        0.0        0.0        0.0        0.0
                                              DETAILED LEADING EDGE LOADING
        1   0.040000     0.159504  0.024410  0.020077
        2   0.080000     0.105797  0.017092  0.014232
        3   0.120000     0.078994  0.015277  0.011659
        4   0.160000     0.060833  0.013916  0.010136
        5   0.200000     0.046723  0.013141  0.009106

 * STOL AERODYNAMIC METHODS PROGRAM *                    * DOUGLAS AIRCRAFT COMPANY *                        * DATE 11/21/73  *  TIME 16.16.59
 CMU CASE NUMBER  1                                           NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                      ****************************************************
                                      *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                      ****************************************************

                                    SECTION  2     Y =  0.800000     CHORD =  0.400000
WING    I       XB         CASE 1    CASE 2    CASE 3    CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
        4   0.0          0.150733  0.025284  0.021240  0.0        0.0        0.0        0.0        0.0        0.0        0.0
        5   0.200000     0.077377  0.018149  0.015801  0.0        0.0        0.0        0.0        0.0        0.0        0.0
        6   0.750000     0.021920  0.065150  0.020256  0.0        0.0        0.0        0.0        0.0        0.0        0.0
 JET
       16   1.000000     0.011222  0.014976  0.054806  0.0        0.0        0.0        0.0        0.0        0.0        0.0
       17   1.250000     0.004356  0.004512  0.006208  0.0        0.0        0.0        0.0        0.0        0.0        0.0
       18   2.000000     0.001912  0.000744  0.001077  0.0        0.0        0.0        0.0        0.0        0.0        0.0
                                              DETAILED LEADING EDGE LOADING
        1   0.040000     0.220077  0.037952  0.032082
        2   0.080000     0.149642  0.027170  0.023091
        3   0.120000     0.115863  0.022938  0.019269
        4   0.160000     0.093860  0.019880  0.017146
        5   0.200000     0.077377  0.018149  0.015801

* STOL AERODYNAMIC METHODS PROGRAM *                    * DOUGLAS AIRCRAFT COMPANY *                        * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                           NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                      ****************************************************
                                      *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                      ****************************************************

                                    SECTION  3     Y =  0.550000     CHORD =  0.400000
WING    I       XB         CASE 1    CASE 2    CASE 3    CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
        7   0.0          0.175350  0.030028  0.028251  0.0        0.0        0.0        0.0        0.0        0.0        0.0
        8   0.200000     0.095480  0.022264  0.021320  0.0        0.0        0.0        0.0        0.0        0.0        0.0
        9   0.750000     0.030099  0.069057  0.024467  0.0        0.0        0.0        0.0        0.0        0.0        0.0
 JET
       19   1.000000     0.015293  0.016886  0.057448  0.0        0.0        0.0        0.0        0.0        0.0        0.0
       20   1.250000     0.005977  0.005143  0.007944  0.0        0.0        0.0        0.0        0.0        0.0        0.0
       21   2.000000     0.001898  0.000873  0.001384  0.0        0.0        0.0        0.0        0.0        0.0        0.0
                                              DETAILED LEADING EDGE LOADING
        1   0.040000     0.257112  0.045212  0.042611
        2   0.080000     0.176267  0.032551  0.030773
        3   0.120000     0.138065  0.027192  0.025806
        4   0.160000     0.113562  0.024178  0.023046
        5   0.200000     0.095480  0.022264  0.021320
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                        ******************************************************
                                        *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                        ******************************************************

                                      SECTION  4      Y =  0.200000     CHORD =  0.400000
WING   I      XB        CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
      10   0.0         0.187858   0.022853   0.032483   0.0        0.0        0.0        0.0        0.0        0.0        0.0
      11   0.200000    0.104140   0.016077   0.024360   0.0        0.0        0.0        0.0        0.0        0.0        0.0
      12   0.750000    0.034286   0.009153   0.026232   0.0        0.0        0.0        0.0        0.0        0.0        0.0
 JET
      22   1.000000    0.017522   0.005143   0.058473   0.0        0.0        0.0        0.0        0.0        0.0        0.0
      23   1.250000    0.006957   0.002387   0.008064   0.0        0.0        0.0        0.0        0.0        0.0        0.0
      24   2.000000    0.002184   0.000916   0.001582   0.0        0.0        0.0        0.0        0.0        0.0        0.0

                                             DETAILED LEADING EDGE LOADING
       1   0.040000    0.275821   0.034236   0.048964
       2   0.080000    0.189580   0.024426   0.035322
       3   0.120000    0.149023   0.020174   0.029580
       4   0.160000    0.123142   0.017707   0.026375
       5   0.200000    0.104140   0.016077   0.024360
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                        ***********************************************
                                        *  WING SPANWISE LOADING FOR FUNDAMENTAL CASE  1  *
                                        ***********************************************
                       ............ LIFT ............     .............................. INDUCED DRAG ..............................
SECTION      Y        CLG       CLMU       CL        *    CDG        CDMU       CS         CD         CMU        GAMMA      ALFIN
   1     0.950000  0.045395  0.017453  0.062849  *  0.0007923  0.0001523  0.0004272  0.0005174  1.0000000  0.0098117  0.0292922
   2     0.800000  0.069333  0.017453  0.086787  *  0.0012101  0.0001523  0.0007931  0.0005693  1.0000000  0.0149986  0.0153765
   3     0.550000  0.084826  0.017453  0.102279  *  0.0014805  0.0001523  0.0010733  0.0005595  1.0000000  0.0184672  0.0098859
   4     0.200000  0.092529  0.017453  0.109982  *  0.0016149  0.0001523  0.0012319  0.0005354  1.0000000  0.0202401  0.0076439
         --------- --------- --------- --------- *  ---------- ---------- ---------- ---------- ---------- ---------- ----------
     TOTAL         0.080865  0.017453  0.098319  *  0.0014114  0.0001523  0.0010161  0.0005476  0.9999998             0.0005453

                     ............ PITCHING MOMENT ............           ... LIFT CENTER ...
     SECTION     Y        CMG       CMMU       CMT        CM        *  *    XCP/C     XCL/C
        1     0.950000 -0.010567 -0.017453  0.017453 -0.010567   *  *   0.232770  0.445832
        2     0.800000 -0.017495 -0.017453  0.017453 -0.017495   *  *   0.252337  0.402696
        3     0.550000 -0.022406 -0.017453  0.017453 -0.022406   *  *   0.264139  0.389706
        4     0.200000 -0.024868 -0.017453  0.017453 -0.024868   *  *   0.268798  0.384801
              --------- --------- --------- --------- ---------   *  *   --------- ---------
          TOTAL        -0.021225 -0.017453  0.017453 -0.021225  (APEX)   0.262467  0.393392 (X/CREF)
                       -0.001008 -0.013090  0.013090 -0.001008  (XMC)    0.104987  0.157357 (X/B/2)
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                        ***********************************************
                                        *  WING SPANWISE LOADING FOR FUNDAMENTAL CASE  2  *
                                        ***********************************************
                       ............ LIFT ............     .............................. INDUCED DRAG ..............................
SECTION      Y        CLG       CLMU       CL        *    CDG        CDMU       CS         CD         CMU        GAMMA      ALFIN
   1     0.950000  0.015962  0.0       0.015962  *  0.0        0.0        0.0000090 -0.0000090  1.0000000  0.0038065  0.0030550
   2     0.800000  0.039795  0.017453  0.057248  *  0.0001748  0.0001523  0.0000223  0.0003048  1.0000000  0.0089892  0.0130596
   3     0.550000  0.044088  0.017453  0.061541  *  0.0001875  0.0001523  0.0000315  0.0003083  1.0000000  0.0099943  0.0126488
   4     0.200000  0.014904  0.0       0.014904  *  0.0        0.0        0.0000182 -0.0000182  1.0000000  0.0036000  0.0013759
         --------- --------- --------- --------- *  ---------- ---------- ---------- ---------- ---------- ---------- ----------
     TOTAL         0.028743  0.008727  0.037470  *  0.0000912  0.0000762  0.0000221  0.0001453  0.9999998             0.0001981

                     ............ PITCHING MOMENT ............           ... LIFT CENTER ...
     SECTION     Y        CMG       CMMU       CMT        CM        *  *    XCP/C     XCL/C
        1     0.950000 -0.007011  0.0        0.0       -0.007011   *  *   0.439223  0.439223
        2     0.800000 -0.020986 -0.017453  0.004363 -0.034076   *  *   0.527353  0.671449
        3     0.550000 -0.022736 -0.017453  0.004363 -0.035826   *  *   0.515692  0.653043
        4     0.200000 -0.005040  0.0        0.0       -0.005040   *  *   0.338176  0.338176
              --------- --------- --------- --------- ---------   *  *   --------- ---------
          TOTAL        -0.013735 -0.008727  0.002182 -0.020280  (APEX)   0.477857  0.599463 (X/CREF)
                       -0.006549 -0.006545  0.002182 -0.010913  (XMC)    0.191143  0.239785 (X/B/2)
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                        ***********************************************
                                        *  WING SPANWISE LOADING FOR FUNDAMENTAL CASE  3  *
                                        ***********************************************
                       ............ LIFT ............     .............................. INDUCED DRAG ..............................
SECTION      Y        CLG       CLMU       CL        *    CDG        CDMU       CS         CD         CMU        GAMMA      ALFIN
   1     0.950000  0.016752  0.017453  0.034206  *  0.0        0.0001523  0.0000063  0.0001460  1.0000000  0.0050801  0.0167288
   2     0.800000  0.025128  0.017453  0.042581  *  0.0        0.0001523  0.0000158  0.0001365  1.0000000  0.0073127  0.0078388
   3     0.550000  0.030613  0.017453  0.048066  *  0.0        0.0001523  0.0000279  0.0001244  1.0000000  0.0086937  0.0043966
   4     0.200000  0.033434  0.017453  0.050887  *  0.0        0.0001523  0.0000368  0.0001155  1.0000000  0.0093899  0.0031181
         --------- --------- --------- --------- *  ---------- ---------- ---------- ---------- ---------- ---------- ----------
     TOTAL         0.029258  0.017453  0.046711  *  0.0        0.0001523  0.0000269  0.0001254  0.9999998             0.0001254

                     ............ PITCHING MOMENT ............           ... LIFT CENTER ...
     SECTION     Y        CMG       CMMU       CMT        CM        *  *    XCP/C     XCL/C
        1     0.950000 -0.009663 -0.017453  0.0       -0.027116   *  *   0.576824  0.792748
        2     0.800000 -0.013565 -0.017453  0.0       -0.031019   *  *   0.539857  0.728462
        3     0.550000 -0.015663 -0.017453  0.0       -0.033116   *  *   0.511643  0.688970
        4     0.200000 -0.016630 -0.017453  0.0       -0.034083   *  *   0.497395  0.669779
              --------- --------- --------- --------- ---------   *  *   --------- ---------
          TOTAL        -0.015030 -0.017453  0.0       -0.032483  (APEX)   0.513709  0.699406 (X/CREF)
                       -0.007716 -0.013090  0.0       -0.020806  (XMC)    0.205483  0.278163 (X/B/2)
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                     NO GROUND EFFECT                             NO STABILITY DERIVATIVES

                                          ****************************************
                                          *  WING TOTAL AERODYNAMIC COEFFICIENTS  *
                                          ****************************************

                CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10

         CCLG  0.0808654  0.0287431  0.0292582  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCLJ  0.0174533  0.0087266  0.0174533  0.0        0.0        0.0        0.0        0.0        0.0        0.0

   **     CCL  0.0983187  0.0374697  0.0467115  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCDG  0.0014114  0.0000912  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCDJ  0.0001523  0.0000762  0.0001523  0.0        0.0        0.0        0.0        0.0        0.0        0.0

          CCS  0.0010161  0.0000221  0.0000269  0.0        0.0        0.0        0.0        0.0        0.0        0.0

          CCD  0.0005476  0.0001453  0.0001254  0.0        0.0        0.0        0.0        0.0        0.0        0.0

   ** CDITZ    0.0005453  0.0001981  0.0001254  0.0        0.0        0.0        0.0        0.0        0.0        0.0

   **     CCJ  0.9999998  0.9999998  0.9999998  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCMG -0.0212245 -0.0137351 -0.0150302  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCMJ -0.0174533 -0.0087266 -0.0174533  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCMT  0.0174533  0.0021817  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

          CCM -0.0212245 -0.0202801 -0.0324835  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CXCP  0.2624674  0.4778571  0.5137087  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CXCL  0.3933923  0.5994632  0.6954065  0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CXCPB  0.1049869  0.1911428  0.2054834  0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CXCLB  0.1573569  0.2397853  0.2781625  0.0        0.0        0.0        0.0        0.0        0.0        0.0

       CCMGMC -0.0010082 -0.0065493 -0.0077156  0.0        0.0        0.0        0.0        0.0        0.0        0.0

       CCMJMC -0.0130900 -0.0065450 -0.0130900  0.0        0.0        0.0        0.0        0.0        0.0        0.0

       CCMTMC  0.0130900  0.0021817  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

   **  CCMMC  -0.0010082 -0.0109126 -0.0208056  0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CLLG  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CLLJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

   *      CLL  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

          CNJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

   *    CNGMC  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

   *      CCY  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                     NO GROUND EFFECT                             NO STABILITY DERIVATIVES

                                          **************************************************
                                          *  WING CHORDWISE LOADING FOR COMPOSITE CASE  1  *
                                          **************************************************
                                                   FUNDAMENTAL CASE FACTORS
           A(1)        A(2)        A(3)        A(4)        A(5)        A(6)        A(7)        A(8)        A(9)        A(10)
           0.0      30.000000   10.000000     0.0         0.0         0.0         0.0         0.0         0.0         0.0

      *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                    DO NOT PLOT THESE LOADING POINTS DIRECTLY

                                 SECTION  1      Y =  0.950000     CHORD =  0.400000
WING
      XB     0.0          0.200000     0.750000
     CPO     0.615903     0.485301     0.631460
     CPA     0.110625     0.046723     0.012297
JET
      XB     1.000000     1.250000     2.000000
     CPO     0.709145     0.112973     0.024332
     CPA     0.006646     0.002689     0.001082

                                 SECTION  2      Y =  0.800000     CHORD =  0.400000
WING
      XB     0.0          0.200000     0.750000
     CPO     0.971045     0.702486     2.157058
     CPA     0.150733     0.077377     0.021920
JET
      XB     1.000000     1.250000     2.000000
     CPO     0.997339     0.197441     0.033089
     CPA     0.011222     0.004356     0 001512

                                 SECTION  3      Y =  0.550000     CHORD =  0.400000
WING
      XB     0.0          0.200000     0.750000
     CPO     1.183351     0.881134     2.316364
     CPA     0.175350     0.095480     0.030099
JET
      XB     1.000000     1.250000     2.000000
     CPO     1.081061     0.229728     0.040041
     CPA     0.015293     0.005977     0.001898

                                 SECTION  4      Y =  0.200000     CHORD =  0.400000
WING
      XB     0.0          0.200000     0.750000
     CPO     1.010428     0.725908     0.536907
     CPA     0.187858     0.104140     0.034286
JET
      XB     1.000000     1.250000     2.000000
     CPO     0.739037     0.152258     0.043304
     CPA     0.017522     0.006957     0.002184
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *                * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                      NO GROUND EFFECT                                 NO STABILITY DERIVATIVES

                                      **************************************************
                                      *  WING SPANWISE LOADING FOR COMPOSITE CASE  1  *
                                      **************************************************
                                               FUNDAMENTAL CASE FACTORS
          A( 1 )     A( 2 )     A( 3 )     A( 4 )     A( 5 )     A( 6 )     A( 7 )     A( 8 )     A( 9 )     A( 10 )
         0.0      30.000000 10.000000  0.0        0.0        0.0        0.0        0.0        0.0        0.0

                          ........... LIFT ............   *  *   ........... PITCHING MOMENT ...........   *  *    ... LIFT CENTER ...
SECTION       Y           CLG0       CLMU0      CL0       *  *     CMG0      CMMU0      CMT0      CM0      *  *     XCP0/C    XCL0/C
                          CLGA       CLMUA      CLA       *  *     CMGA      CMMUA      CMTA      CMA      *  *     XCPA/C    XCLA/C
   1       0.950000   0.646395  0.174533  0.820928  *  *  -0.306963 -0.174533  0.0       -0.481496  *  *    0.474885  0.586527
                      0.045395  0.017453  0.062849  *  *  -0.010567 -0.017453  0.017453 -0.010567  *  *    0.232770  0.445832
   2       0.800000   1.445130  0.698132  2.143262  *  *  -0.765236 -0.698132  0.130900 -1.332467  *  *    0.529527  0.682776
                      0.069333  0.017453  0.086787  *  *  -0.017495 -0.017453  0.017453 -0.017495  *  *    0.252337  0.402696
   3       0.550000   1.628773  0.698132  2.326904  *  *  -0.838706 -0.698132  0.130900 -1.405937  *  *    0.514931  0.660464
                      0.084826  0.017453  0.102279  *  *  -0.022406 -0.017453  0.017453 -0.022406  *  *    0.264135  0.389706
   4       0.200000   0.781444  0.174533  0.955976  *  *  -0.317498 -0.174533  0.0       -0.492031  *  *    0.406297  0.514690
                      0.092529  0.017453  0.109982  *  *  -0.024068 -0.017453  0.017453 -0.024068  *  *    0.268758  0.384801
          ---------- --------- --------- ----------  *  *  --------- --------- --------- ---------  *  *   --------- ---------
     TOTAL            1.154874  0.436332  1.591206  *  *  -0.562355 -0.436332  0.065450 -0.933237 (APEX)   0.486940  0.627629
                      0.080865  0.017453  0.098319  *  *  -0.021225 -0.017453  0.017453 -0.021225 (APEX)   0.262467  0.393392
                                                    *  *  -0.273636 -0.327249  0.065450 -0.535435 (XMC)    0.194776  0.251051
                                                    *  *  -0.001008 -0.013090  0.013090 -0.001008 (XMC)    0.104987  0.157357


                      ..................................... INDUCED DRAG ......................................
   SECTION       Y         CDG0        CDMU0       CS0         CD0        GAMMA0      ALFIN0      CT0         CMU
                           CDGA        CDMUA       CSA         CDA
                           CDGA2       CDMUA2      CSA2        CDA2       GAMMAA      ALFINA
      1       0.950000  0.0         0.0152309   0.0132414   0.0019895  0.1649964   0.2589407   0.9980105   1.0000000
                        0.0112817   0.0030462   0.0047567   0.0095712
                        0.0007923   0.0001523   0.0004272   0.0005174  0.0098117   0.0292922
      2       0.800000  0.2064547   0.2436939   0.0329144   0.4172341  0.3428028   0.4701773   0.5827659   1.0000000
                        0.0273914   0.0121847   0.0102104   0.0293576
                        0.0012101   0.0001523   0.0007931   0.0005693  0.0149986   0.0153769
      3       0.550000  0.2223610   0.2436939   0.0488803   0.4171745  0.3867651   0.4234283   0.5828255   1.0000000
                        0.0313984   0.0121847   0.0144863   0.0290968
                        0.0014805   0.0001523   0.0010733   0.0005595  0.0184672   0.0098859
      4       0.200000  0.0         0.0152309   0.0356384  -0.0204075  0.2010989   0.0724573   1.0204067   1.0000000
                        0.0136388   0.0030462   0.0132517   0.0034332
                        0.0016149   0.0001523   0.0012319   0.0005354  0.0202401   0.0076439
                        ----------  ----------  ----------  ---------- ----------  ----------  ----------  ----------
              TOTAL     0.1079992   0.1294623   0.0368265   0.2006350              0.2730447   0.7993648   0.9999998
                        0.0214814   0.0076154   0.0121659   0.0169310              0.0223941
                        0.0014114   0.0001523   0.0010161   0.0005476              0.0005453
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *                * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                      NO GROUND EFFECT                                 NO STABILITY DERIVATIVES

                               ****************************************************************
                               *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE   1  *
                               ****************************************************************

                         ALPHA=0        ALPHA        ALPHA**2

             CCLG       1.154874      0.080865

             CCLJ       0.436332      0.017453

       **     CCL       1.591206      0.098319

             CCDG       0.1079992     0.0214814    0.0014114

             CCDJ       0.1294623     0.0076154    0.0001523

              CCS       0.0368265     0.0121659    0.0010161

              CCD       0.2006350     0.0169310    0.0005476

       **  CDITZ        0.2730447     0.0223941    0.0005453

       **     CCJ       0.9999998

             CCMG      -0.562355     -0.021225

             CCMJ      -0.436332     -0.017453

             CCMT       0.065450      0.017453

              CCM      -0.933237     -0.021225

             CXCP       0.486940      0.262467

             CXCL       0.627629      0.393392

            CXCPB       0.194776      0.104987

            CXCLB       0.251051      0.157357

           CCMGMC      -0.273636     -0.001008

           CCMJMC      -0.327249     -0.013090

           CCMTMC       0.065450      0.013090

       **  CCMMC       -0.535435     -0.001008

             CLLG       0.0           0.0

             CLLJ       0.0           0.0

       *      CLL       0.0           0.0

              CNJ       0.0           0.0          0.0

       *    CNGMC       0.0           0.0          0.0

       *      CCY       0.0           0.0          0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                              NO STABILITY DERIVATIVES

                                  ************************************************************
                                  *  TABULATED WING TOTAL COEFFICIENTS FOR COMPOSITE CASE  1  *
                                  ************************************************************

ALPHA      CCL       CCL**2     CCM(MC)    CLLJ      CLL          CDITZ       CT          CNJ       CN        CCV
-10.0   0.6080186  0.3696866 -0.5253534  0.0      0.0         0.1036339  0.8963659    0.0       0.0       0.0
 -9.0   0.7063373  0.4989124 -0.5263615  0.0      0.0         0.1156672  0.8843326    0.0       0.0       0.0
 -8.0   0.8046560  0.6474713 -0.5273697  0.0      0.0         0.1287912  0.8712086    0.0       0.0       0.0
 -7.0   0.9029747  0.8153633 -0.5283780  0.0      0.0         0.1430058  0.8569940    0.0       0.0       0.0
 -6.0   1.0012932  1.0025873 -0.5293861  0.0      0.0         0.1583109  0.8416889    0.0       0.0       0.0
 -5.0   1.0996113  1.2091446 -0.5303943  0.0      0.0         0.1747067  0.8252931    0.0       0.0       0.0
 -4.0   1.1979303  1.4350367 -0.5314025  0.0      0.0         0.1921931  0.8078067    0.0       0.0       0.0
 -3.0   1.2962494  1.6802616 -0.5324107  0.0      0.0         0.2107701  0.7892298    0.0       0.0       0.0
 -2.0   1.3945679  1.9448175 -0.5334189  0.0      0.0         0.2304376  0.7695622    0.0       0.0       0.0
 -1.0   1.4928869  2.2287102 -0.5344271  0.0      0.0         0.2511958  0.7488040    0.0       0.0       0.0
  0.0   1.5912056  2.5319347 -0.5354353  0.0      0.0         0.2730447  0.7269551    0.0       0.0       0.0
  1.0   1.6895237  2.8544903 -0.5364435  0.0      0.0         0.2959840  0.7040158    0.0       0.0       0.0
  2.0   1.7878428  3.1963816 -0.5374517  0.0      0.0         0.3200141  0.6799858    0.0       0.0       0.0
  3.0   1.8861609  3.5576019 -0.5384599  0.0      0.0         0.3451347  0.6548651    0.0       0.0       0.0
  4.0   1.9844799  3.9381599 -0.5394681  0.0      0.0         0.3713459  0.6286539    0.0       0.0       0.0
  5.0   2.0827990  4.3380508 -0.5404763  0.0      0.0         0.3986478  0.6013520    0.0       0.0       0.0
  6.0   2.1811171  4.7572708 -0.5414845  0.0      0.0         0.4270402  0.5729597    0.0       0.0       0.0
  7.0   2.2794361  5.1958284 -0.5424926  0.0      0.0         0.4565232  0.5434766    0.0       0.0       0.0
  8.0   2.3777552  5.6537189 -0.5435008  0.0      0.0         0.4870969  0.5129029    0.0       0.0       0.0
  9.0   2.4760733  6.1309385 -0.5445091  0.0      0.0         0.5187612  0.4812387    0.0       0.0       0.0
 10.0   2.5743923  6.6274958 -0.5455172  0.0      0.0         0.5515161  0.4484838    0.0       0.0       0.0
 11.0   2.6727104  7.1433802 -0.5465254  0.0      0.0         0.5853614  0.4146384    0.0       0.0       0.0
 12.0   2.7710295  7.6786041 -0.5475336  0.0      0.0         0.6202976  0.3797023    0.0       0.0       0.0
 13.0   2.8693485  8.2331600 -0.5485418  0.0      0.0         0.6563242  0.3436756    0.0       0.0       0.0
 14.0   2.9676666  8.8070450 -0.5495500  0.0      0.0         0.6934415  0.3065583    0.0       0.0       0.0
 15.0   3.0659857  9.4002676 -0.5505582  0.0      0.0         0.7316494  0.2683504    0.0       0.0       0.0
 16.0   3.1643047 10.0128241 -0.5515664  0.0      0.0         0.7709479  0.2290519    0.0       0.0       0.0
 17.0   3.2626228 10.6447077 -0.5525746  0.0      0.0         0.8113369  0.1886629    0.0       0.0       0.0
 18.0   3.3609419 11.2959299 -0.5535828  0.0      0.0         0.8528166  0.1471832    0.0       0.0       0.0
 19.0   3.4592600 11.9664793 -0.5545910  0.0      0.0         0.8953870  0.1046128    0.0       0.0       0.0
 20.0   3.5575790 12.6563683 -0.5555992  0.0      0.0         0.9390479  0.0609519    0.0       0.0       0.0
 21.0   3.6558981 13.3655901 -0.5566074  0.0      0.0         0.9837995  0.0162004    0.0       0.0       0.0
 22.0   3.7542162 14.0941391 -0.5576156  0.0      0.0         1.0296412 -0.0296413    0.0       0.0       0.0
 23.0   3.8525352 14.8420277 -0.5586237  0.0      0.0         1.0765734 -0.0765736    0.0       0.0       0.0
 24.0   3.9508543 15.6092491 -0.5596319  0.0      0.0         1.1245975 -0.1245977    0.0       0.0       0.0
 25.0   4.0491724 16.3957825 -0.5606402  0.0      0.0         1.1737108 -0.1737110    0.0       0.0       0.0
 26.0   4.1474915 17.2016754 -0.5616483  0.0      0.0         1.2239161 -0.2239162    0.0       0.0       0.0
 27.0   4.2458096 18.0268860 -0.5626565  0.0      0.0         1.2752104 -0.2752106    0.0       0.0       0.0
 28.0   4.3441286 18.8714447 -0.5636647  0.0      0.0         1.3275967 -0.3275968    0.0       0.0       0.0
 29.0   4.4424477 19.7353363 -0.5646729  0.0      0.0         1.3810730 -0.3810732    0.0       0.0       0.0
 30.0   4.5407658 20.6185455 -0.5656811  0.0      0.0         1.4356403 -0.4356405    0.0       0.0       0.0


   * STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
   CMU CASE NUMBER  1                                    NO GROUND EFFECT                              NO STABILITY DERIVATIVES

                                                 *************************************
                                                 * HORIZONTAL TAIL GEOMETRY SUMMARY *
                                                 *************************************

                                                    ASPECT RATIO =      2.666599
                                                     TAPER RATIO =      1.000000
                                                           SWEEP =      0.0
                                                            SPAN =      0.800000
                                                            AREA =      0.240006
                                                             MAC =      0.300007
                                                             XMC =      1.075002
                                                             YMC =      0.0
                                                          LENGTH =      0.975002
                                                          HEIGHT =      0.300000
                                                          VOLUME =      0.731269
                                                           NROWH =      4

                                           ROW    YLEAD        XLEAD        XTRAIL       CHORD
                                            1     0.350000     1.000000     1.300007     0.300007
                                            2     0.150000     1.000000     1.300007     0.300007
                                            3    -0.150000     1.000000     1.300007     0.300007
                                            4    -0.350000     1.000000     1.300007     0.300007


                                                 ***********************************
                                                 * VERTICAL TAIL GEOMETRY SUMMARY *
                                                 ***********************************

                                                    ASPECT RATIO =      1.333300
                                                     TAPER RATIO =      1.000000
                                                           SWEEP =      0.0
                                                            SPAN =      0.400000
                                                            AREA =      0.120003
                                                             MAC =      0.300007
                                                             XMC =      1.075002
                                                             ZMC =      0.200000
                                                          LENGTH =      0.975002
                                                          HEIGHT =      0.0
                                                          VOLUME =      0.073127
                                                           NROWV =      2

                                           ROW    ZLEAD        XLEAD        XTRAIL       CHORD
                                            1     0.350000     1.000000     1.300007     0.300007
                                            2     0.150000     1.000000     1.300007     0.300007
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                  * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                         NO GROUND EFFECT                                   NO STABILITY DERIVATIVES

                                              *****************************************
                                              *          ISOLATED TAIL ANALYSIS       *
                                              *  CHORDWISE LOADING FOR ALL TAIL CASES *
                                              *****************************************

                                                   CASE 1 IS THE ALPHA  CASE
                                                   CASE 2 IS THE  BETA  CASE
                                                   CASE 3 IS THE CAMBER CASE

                                                       *******************
                                                       * HORIZONTAL TAIL *
                                                       *******************

                         SECTION  1      Y =   0.350000      CHORD =   0.300007
               ELEMENT      X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                  1         0.0          0.071704     0.013105     0.316012
                  2         0.500000     0.017510     0.006195     0.530445

                         SECTION  2      Y =   0.150000      CHORD =   0.300007
               ELEMENT      X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                  3         0.0          0.102980     0.025675     0.547442
                  4         0.500000     0.030074     0.012695     0.743646

                         SECTION  3      Y =  -0.150000      CHORD =   0.300007
               ELEMENT      X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                  5         0.0          0.102980    -0.025675     0.220755
                  6         0.500000     0.030074    -0.012695     0.941081

                         SECTION  4      Y =  -0.350000      CHORD =   0.300007
               ELEMENT      X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                  7         0.0          0.071706    -0.013105     0.146604
                  8         0.500000     0.017510    -0.006195     0.438058

                                                       *****************
                                                       * VERTICAL TAIL *
                                                       *****************

                         SECTION  1      Z =   0.350000      CHORD =   0.300007
               ELEMENT      X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                  1         0.0         -0.000000     0.054619     0.124678
                  2         0.500000    -0.000000     0.008603     0.414898

                         SECTION  2      Z =   0.150000      CHORD =   0.300007
               ELEMENT      X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                  3         0.0         -0.000000     0.110188     0.487885
                  4         0.500000    -0.000000     0.038005     0.696890

* STOL AERODYNAMIC METHODS PROGRAM *                  * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                         NO GROUND EFFECT                                   NO STABILITY DERIVATIVES

                                         ***************************************************
                                         *               ISOLATED TAIL ANALYSIS            *
                                         *  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES  *
                                         ***************************************************

                                                       *******************
                                                       * HORIZONTAL TAIL *
                                                       *******************

            CASE 1                         CASE 2                        CASE 3                         CASE 4                        CASE 5
K   CLH     CMH     CDIH      CLH      CMH     CDIH      CLH      CMH     CDIH      CLH      CMH      CDIH      CLH      CMH      CDIH
1  0.0446 -0.0100  0.0006   0.0096 -0.0028 -0.0000   0.4232 -0.1899  0.0362
2  0.0665 -0.0158  0.0007   0.0192 -0.0056 -0.0001   0.6455 -0.2666  0.0395
3  0.0665 -0.0158  0.0007  -0.0192  0.0056 -0.0001   0.3849 -0.1834  0.0429
4  0.0446 -0.0100  0.0006  -0.0096  0.0028 -0.0000   0.2924 -0.1461  0.0363
-------------------------------------------------------------------------------------------------------------------------------------
   0.0183 -0.0444  0.0002  -0.0000  0.0000 -0.0000   0.1428 -0.3685  0.0120

     CLLH =    0.000000       CLLH =  -0.000901       CLLH =  -0.006116       CLLH =

                                                       *****************
                                                       * VERTICAL TAIL *
                                                       *****************

            CASE 1                         CASE 2                        CASE 3                         CASE 4                        CASE 5
K   CYV     CNV     CDIV      CYV      CNV     CDIV      CYV      CNV     CDIV      CYV      CNV      CDIV      CYV      CNV      CDIV
1 -0.0000  0.0000  0.0000   0.0316 -0.0061  0.0004   0.2698 -0.1374  0.0338
2 -0.0000  0.0000 -0.0000   0.0781 -0.0193  0.0007   0.5924 -0.2483  0.0402
-------------------------------------------------------------------------------------------------------------------------------------
  -0.0000  0.0000 -0.0000   0.0090 -0.0121  0.0000   0.0384 -0.0988  0.0029

     CLLV =  -0.000000        CLLV =   0.000866       CLLV =   0.006769       CLLV =

        *** NOTE *** ALL TAIL COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF

* STOL AERODYNAMIC METHODS PROGRAM *                  * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                         NO GROUND EFFECT                                   NO STABILITY DERIVATIVES

                                            ***********************************************
                                            *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                            ***********************************************

                                                     ALPHA =   0.0  DEGREES

             X            Y            Z          U/UINF       V/UINF       W/UINF      DOWNWASH    SIDEWASH
                                                                                         ANGLE       ANGLE
                                                                                       (DEGREES)   (DEGREES)

    1     1.112502     0.350000     0.300000     0.002594    -0.057583    -0.100819     5.742283   -3.287108
    2     1.262506     0.350000     0.300000    -0.002200    -0.062206    -0.095322     5.457006   -3.567359
    3     1.112502     0.150000     0.300000     0.009943    -0.022372    -0.122772     6.931067   -1.269021
    4     1.262506     0.150000     0.300000     0.002505    -0.025898    -0.120878     6.875284   -1.479818
    5     1.112502    -0.150000     0.300000     0.009943     0.022372    -0.122772     6.931067    1.269021
    6     1.262506    -0.150000     0.300000     0.002505     0.025898    -0.120878     6.875284    1.479818
    7     1.112502    -0.350000     0.300000     0.002594     0.057583    -0.100819     5.742283    3.287108
    8     1.262506    -0.350000     0.300000    -0.002200     0.062206    -0.095322     5.457006    3.567359
    9     1.112502     0.0          0.350000     0.012040     0.0         -0.114205     6.438957    0.0
   10     1.262506     0.0          0.350000     0.004513     0.0         -0.113919     6.447265    0.0
   11     1.112502     0.0          0.150000     0.013178     0.0         -0.153100     8.592689    0.0
   12     1.262506     0.0          0.150000     0.003492     0.0         -0.154352     8.744393    0.0
   13    -0.475000     0.0         -0.075000    -0.010373     0.0          0.039025    -2.298241    0.0
   14    -0.375000     0.0         -0.075000    -0.013333     0.0          0.049842    -2.891883    0.0
   15     0.350000     0.0         -0.075000    -0.070370     0.0         -0.052318     3.221103    0.0
   16     1.150000     0.0         -0.075000     0.019994     0.0         -0.137312     7.667088    0.0
   17    -0.500000     0.125000     0.125000     0.006778    -0.000266     0.035539    -2.021684   -0.015120
   18     0.0          0.125000     0.125000     0.130169    -0.000329     0.081583    -4.099892   -0.016561
   19     0.500000     0.125000     0.125000     0.103538     0.018492    -0.111137     5.750865    0.960039
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *                * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                                NO STABILITY DERIVATIVES

                                          **********************************************
                                          *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                          **********************************************

                                                   ALPHA =   5.00 DEGREES

              X           Y           Z         U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                     ANGLE       ANGLE
                                                                                   (DEGREES)   (DEGREES)

   1      1.112502    0.350000    0.300000   -0.006351   -0.103751   -0.145784    8.346623   -5.960866
   2      1.262506    0.350000    0.300000   -0.011837   -0.111631   -0.139340    8.026280   -6.445293
   3      1.112502    0.150000    0.300000    0.000962   -0.040541   -0.188462   10.662901   -2.319354
   4      1.262506    0.150000    0.300000   -0.008179   -0.045618   -0.187149   10.685400   -2.633433
   5      1.112502   -0.150000    0.300000    0.000962    0.040541   -0.188462   10.662901    2.319352
   6      1.262506   -0.150000    0.300000   -0.008179    0.045618   -0.187149   10.685400    2.633432
   7      1.112502   -0.350000    0.300000   -0.006351    0.103751   -0.145784    8.346623    5.960866
   8      1.262506   -0.350000    0.300000   -0.011837    0.111631   -0.139340    8.026284    6.445293
   9      1.112502    0.0         0.350000    0.004492    0.0        -0.175217    9.894764    0.0
  10      1.262506    0.0         0.350000   -0.004861    0.0        -0.175106    9.979650    0.0
  11      1.112502    0.0         0.150000    0.002806    0.0        -0.237778   13.339205    0.0
  12      1.262506    0.0         0.150000   -0.009094    0.0        -0.240407   13.637230    0.0
  13     -0.475000    0.0        -0.075000   -0.009288    0.0         0.056095   -3.240655    0.0
  14     -0.375000    0.0        -0.075000   -0.012630    0.0         0.072607   -4.205709    0.0
  15      0.350000    0.0        -0.075000   -0.100329    0.0        -0.118399    7.497184    0.0
  16      1.150000    0.0        -0.075000    0.015212    0.0        -0.210029   11.600953    0.0
  17     -0.500000    0.125000    0.125000    0.019470   -0.000530    0.048603   -2.740236   -0.029932
  18      0.0         0.125000    0.125000    0.237657   -0.002501    0.083528   -3.860955   -0.115778
  19      0.500000    0.125000    0.125000    0.114337    0.015152   -0.197367   10.043854    0.778998
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *                * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                                NO STABILITY DERIVATIVES

                                          **********************************************
                                          *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                          **********************************************

                                                   ALPHA =  10.00 DEGREES

              X           Y           Z         U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                     ANGLE       ANGLE
                                                                                   (DEGREES)   (DEGREES)

   1      1.112502    0.350000    0.300000   -0.020997   -0.159879   -0.194393   11.230708   -9.274952
   2      1.262506    0.350000    0.300000   -0.027101   -0.174289   -0.188332   10.955711  -10.156436
   3      1.112502    0.150000    0.300000   -0.015611   -0.060849   -0.261098   14.855018   -3.537176
   4      1.262506    0.150000    0.300000   -0.026571   -0.067973   -0.262519   15.092750   -3.994403
   5      1.112502   -0.150000    0.300000   -0.015611    0.060849   -0.261098   14.855018    3.537174
   6      1.262506   -0.150000    0.300000   -0.026571    0.067973   -0.262519   15.092753    3.994403
   7      1.112502   -0.350000    0.300000   -0.020997    0.159879   -0.194393   11.230708    9.274952
   8      1.262506   -0.350000    0.300000   -0.027101    0.174289   -0.188332   10.955711   10.156436
   9      1.112502    0.0         0.350000   -0.010052    0.0        -0.242007   13.737336    0.0
  10      1.262506    0.0         0.350000   -0.021281    0.0        -0.244074   14.002867    0.0
  11      1.112502    0.0         0.150000   -0.016107    0.0        -0.327383   18.404495    0.0
  12      1.262506    0.0         0.150000   -0.030120    0.0        -0.332593   18.927933    0.0
  13     -0.475000    0.0        -0.075000   -0.004977    0.0         0.073076   -4.200330    0.0
  14     -0.375000    0.0        -0.075000   -0.007671    0.0         0.095379   -5.490164    0.0
  15      0.350000    0.0        -0.075000   -0.140320    0.0        -0.179290   11.780450    0.0
  16      1.150000    0.0        -0.075000    0.005709    0.0        -0.276641   15.380000    0.0
  17     -0.500000    0.125000    0.125000    0.026732   -0.000837    0.060232   -3.357347   -0.046709
  18      0.0         0.125000    0.125000    0.338598   -0.004819    0.068437   -2.926752   -0.206259
  19      0.500000    0.125000    0.125000    0.113130    0.011575   -0.286830   14.449589    0.595763
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *                * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                                NO STABILITY DERIVATIVES

                                          **********************************************
                                          *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                          **********************************************

                                                   ALPHA =  15.00 DEGREES

              X           Y           Z         U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                     ANGLE       ANGLE
                                                                                   (DEGREES)   (DEGREES)

   1      1.112502    0.350000    0.300000   -0.042018   -0.228221   -0.246995   14.457607  -13.399878
   2      1.262506    0.350000    0.300000   -0.048990   -0.254140   -0.243622   14.368589  -14.961610
   3      1.112502    0.150000    0.300000   -0.040853   -0.083088   -0.340835   19.562759   -4.950978
   4      1.262506    0.150000    0.300000   -0.054090   -0.092393   -0.347751   20.185272   -5.578731
   5      1.112502   -0.150000    0.300000   -0.040853    0.083088   -0.340835   19.562759    4.950974
   6      1.262506   -0.150000    0.300000   -0.054090    0.092393   -0.347751   20.185272    5.578727
   7      1.112502   -0.350000    0.300000   -0.042018    0.228221   -0.246995   14.457610   13.399878
   8      1.262506   -0.350000    0.300000   -0.048990    0.254140   -0.243622   14.368589   14.961610
   9      1.112502    0.0         0.350000   -0.032455    0.0        -0.314410   18.001907    0.0
  10      1.262506    0.0         0.350000   -0.045825    0.0        -0.320519   18.567596    0.0
  11      1.112502    0.0         0.150000   -0.044166    0.0        -0.420535   23.747864    0.0
  12      1.262506    0.0         0.150000   -0.060147    0.0        -0.428731   24.520950    0.0
  13     -0.475000    0.0        -0.075000    0.002506    0.0         0.089528   -5.103203    0.0
  14     -0.375000    0.0        -0.075000    0.001489    0.0         0.117579   -6.694101    0.0
  15      0.350000    0.0        -0.075000   -0.189551    0.0        -0.233927   16.100204    0.0
  16      1.150000    0.0        -0.075000   -0.007432    0.0        -0.336456   18.734146    0.0
  17     -0.500000    0.125000    0.125000    0.040346   -0.001184    0.070083   -3.853896   -0.065188
  18      0.0         0.125000    0.125000    0.438906   -0.007289    0.036097   -1.437045   -0.290250
  19      0.500000    0.125000    0.125000    0.099273    0.007717   -0.378274   18.989014    0.402226
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *                * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                                NO STABILITY DERIVATIVES

                                          **********************************************
                                          *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                          **********************************************

                                                   ALPHA =  20.00 DEGREES

              X           Y           Z         U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                     ANGLE       ANGLE
                                                                                   (DEGREES)   (DEGREES)

   1      1.112502    0.350000    0.300000   -0.070245   -0.311474   -0.304174   18.115738  -18.521179
   2      1.262506    0.350000    0.300000   -0.078986   -0.356476   -0.307556   18.465820  -21.158813
   3      1.112502    0.150000    0.300000   -0.075925   -0.106845   -0.427591   24.831070   -6.595442
   4      1.262506    0.150000    0.300000   -0.092318   -0.117722   -0.443282   26.029358   -7.389717
   5      1.112502   -0.150000    0.300000   -0.075925    0.106845   -0.427591   24.831070    6.595438
   6      1.262506   -0.150000    0.300000   -0.092318    0.117722   -0.443282   26.029358    7.389714
   7      1.112502   -0.350000    0.300000   -0.070245    0.311474   -0.304174   18.115753   18.521179
   8      1.262506   -0.350000    0.300000   -0.078986    0.356476   -0.307556   18.465820   21.158813
   9      1.112502    0.0         0.350000   -0.063607    0.0        -0.392029   22.717026    0.0
  10      1.262506    0.0         0.350000   -0.079610    0.0        -0.404157   23.707016    0.0
  11      1.112502    0.0         0.150000   -0.081881    0.0        -0.515608   29.318176    0.0
  12      1.262506    0.0         0.150000   -0.099431    0.0        -0.526079   30.291840    0.0
  13     -0.475000    0.0        -0.075000    0.013044    0.0         0.105033   -5.919284    0.0
  14     -0.375000    0.0        -0.075000    0.014715    0.0         0.138657   -7.781046    0.0
  15      0.350000    0.0        -0.075000   -0.247128    0.0        -0.281381   20.492889    0.0
  16      1.150000    0.0        -0.075000   -0.025444    0.0        -0.389821   21.801331    0.0
  17     -0.500000    0.125000    0.125000    0.056050   -0.001569    0.077852   -4.216207   -0.085112
  18      0.0         0.125000    0.125000    0.536543   -0.009918   -0.013560    0.505616   -0.369835
  19      0.500000    0.125000    0.125000    0.072150    0.003519   -0.470348   23.686890    0.188056
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                           NO STABILITY DERIVATIVES

                                           ********************************************
                                           *        INTERFERENCE TAIL ANALYSIS        *
                                           *  CHORDWISE LOADING FOR ALL TAIL CASES    *
                                           ********************************************

                                                 CASE 1 IS FOR ALPHA =  0.0
                                                 CASE 2 IS FOR ALPHA =  5.00
                                                 CASE 3 IS FOR ALPHA = 10.00
                                                 CASE 4 IS FOR ALPHA = 15.00
                                                 CASE 5 IS FOR ALPHA = 20.00

                                                       *******************
                                                       * HORIZONTAL TAIL *
                                                       *******************

                          SECTION  1      Y =   0.350000     CHORD =   0.300007
             ELEMENT       X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                1          0.0        -0.136435    -0.086723     0.093470     0.159053     0.184351
                2          0.500000    0.423073     0.448917     0.462466     0.460276     0.437517

                          SECTION  2      Y =   0.150000     CHORD =   0.300007
             ELEMENT       X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                3          0.0        -0.157083    -0.023165     0.059184     0.082867     0.041892
                4          0.500000    0.542001     0.577936     0.591765     0.579694     0.538580

                          SECTION  3      Y =  -0.150000     CHORD =   0.300007
             ELEMENT       X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                5          0.0        -0.475771    -0.341053    -0.259504    -0.235821    -0.276796
                6          0.500000    0.339436     0.375370     0.389200     0.377089     0.336015

                          SECTION  4      Y =  -0.350000     CHORD =   0.300007
             ELEMENT       X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                7          0.0        -0.305764    -0.176051    -0.075851    -0.010276     0.019022
                8          0.500000    0.330685     0.356529     0.370078     0.367889     0.349129

                                                       *****************
                                                       * VERTICAL TAIL *
                                                       *****************

                          SECTION  1      Z =   0.350000     CHORD =   0.300007
             ELEMENT       X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                1          0.0         0.124678     0.124678     0.124678     0.124678     0.124678
                2          0.500000    0.414898     0.414898     0.414898     0.414898     0.414898

                          SECTION  2      Z =   0.150000     CHORD =   0.300007
             ELEMENT       X/C          CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
                3          0.0         0.487885     0.487885     0.487885     0.487885     0.487885
                4          0.500000    0.696890     0.696890     0.696890     0.696890     0.696890
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                           NO STABILITY DERIVATIVES

                                    *****************************************************
                                    *              INTERFERENCE TAIL ANALYSIS           *
                                    *  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES    *
                                    *****************************************************

                                                       *******************
                                                       * HORIZONTAL TAIL *
                                                       *******************

              CASE 1                       CASE 2                       CASE 3                       CASE 4                       CASE 5
K   CLH     CMH     CDIH       CLH     CMH     CDIH       CLH     CMH     CDIH       CLH     CMH     CDIH       CLH     CMH     CDIH
1   0.1433 -0.1237  0.0024     0.2211 -0.1399  0.0144     0.2780 -0.1504  0.0295     0.3097 -0.1538  0.0436     0.3109 -0.1482  0.0522
2   0.1925 -0.1596 -0.0091     0.2774 -0.1792 -0.0059     0.3295 -0.1806 -0.0035     0.3313 -0.1863 -0.0038     0.2902 -0.1709 -0.0068
3  -0.0682 -0.0763  0.0055     0.0168 -0.0959  0.0071     0.0648 -0.1054  0.0086     0.0706 -0.1031  0.0082     0.0296 -0.0877  0.0060
4   0.0125 -0.0842  0.0070     0.0902 -0.1004  0.0150     0.1471 -0.1109  0.0261     0.1708 -0.1143  0.0363     0.1801 -0.1088  0.0414
------------------------------------------------------------------------------------------------------------------------------------------
    0.0198 -0.0704 -0.0001     0.0448 -0.1274  0.0012     0.0599 -0.1570  0.0026     0.0635 -0.1577  0.0035     0.0544 -0.1301  0.0034

      CLLH =  -0.006116          CLLH =  -0.006116          CLLH =  -0.006116          CLLH =  -0.006116          CLLH =  -0.006116

                                                       *****************
                                                       * VERTICAL TAIL *
                                                       *****************

              CASE 1                       CASE 2                       CASE 3                       CASE 4                       CASE 5
K   CYV     CNV     CDIV       CYV     CNV     CDIV       CYV     CNV     CDIV       CYV     CNV     CDIV       CYV     CNV.    CDIV
1   0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338
2   0.5924 -0.2483  0.0402     0.5924 -0.2483  0 0402     0.5924 -0.2483  0.0402     0.5924 -0.2483  0.0402     0.5924 -0.2483  0.0402
------------------------------------------------------------------------------------------------------------------------------------------
    0.0384 -0.0988  0.0029     0.0384 -0.0988  0.0029     0.0384 -0.0988  0.0029     0.0384 -0.0988  0.0029     0.0384 -0.0988  0.0029

      CLLV =   0.006769          CLLV =   0.006769          CLLV =   0.006769          CLLV =   0.006769          CLLV =   0.006769

         *** NOTE *** ALL TAIL COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                    NO GROUND EFFECT                           NO STABILITY DERIVATIVES

                               ************************************************************
                               * SECTIONAL AND TOTAL FUSELAGE LOADING FROM PRESSURE INTEGRATION *
                               *                    FOR COMPOSITE CASE   1                 *
                               ************************************************************

SEG      X          CM(0)     CN(0)       CM(5)     CN(5)       CM(10)    CN(10)      CM(15)    CN(15)      CM(20)    CN(20)

  1  -0.475000   0.000214   0.0       0.009358   0.0       0.017207   0.0       0.023494   0.0       0.028024   0.0
  2  -0.375000   0.004990   0.0       0.018840   0.0       0.031912   0.0       0.043731   0.0       0.053889   0.0
  3   0.350000  -0.000000   0.0       0.000000   0.0       0.000000   0.0      -0.000000   0.0      -0.000000   0.0
  4   1.150000  -0.059753   0.0      -0.051828   0.0      -0.041480   0.0      -0.029240   0.0      -0.016142   0.0
----------------------------------------------------------------------------------------------------------------------
TOTAL            -0.054548   0.0      -0.023630   0.0       0.007639   0.0       0.037985   0.0       0.065770   0.0

        *** NOTE *** ALL FUSELAGE COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME 16.16.59
CMU CASE NUMBER  1                                     NO GROUND EFFECT                              NO STABILITY DERIVATIVES

                   ***********************************************************************
                   *  SUMMARY OF AIRCRAFT AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                   ***********************************************************************

                                              *** WING ***
                ALPHA      CCL          CMMC        CLL              CDIT2        CN            CCY
                 0.0     1.591206    -0.535435    0.0              0.273049     0.0           0.0
                 5.0     2.082795    -0.540476    0.0              0.398648     0.0           0.0
                10.0     2.574392    -0.545517    0.0              0.551516     0.0           0.0
                15.0     3.065986    -0.550558    0.0              0.731649     0.0           0.0
                20.0     3.557579    -0.555599    0.0              0.939048     0.0           0.0

                                         *** HORIZONTAL TAIL ***
                ALPHA      CCL          CMMC        CLL              CCDI         CN            CCY
                 0.0     0.019825    -0.070358   -0.006116     -0.000053
                 5.0     0.044767    -0.127387   -0.006116      0.001228
                10.0     0.059853    -0.156988   -0.006116      0.002649
                15.0     0.063532    -0.157706   -0.006116      0.003499
                20.0     0.054397    -0.130110   -0.006116      0.003412

                                          *** VERTICAL TAIL ***
                ALPHA      CCL          CMMC        CLL              CCDI         CN            CCY
                 0.0                               0.006769      0.002893    -0.098765      0.038381
                 5.0                               0.006769      0.002893    -0.098765      0.038381
                10.0                               0.006769      0.002893    -0.098765      0.038381
                15.0                               0.006769      0.002893    -0.098765      0.038381
                20.0                               0.006769      0.002893    -0.098765      0.038381

                                            *** FUSELAGE ***
                ALPHA      CCL          CMMC        CLL              CCDI         CN            CCY
                 0.0                 -0.054548                                 0.0
                 5.0                 -0.023630                                 0.0
                10.0                  0.007639                                 0.0
                15.0                  0.037985                                 0.0
                20.0                  0.065770                                 0.0

                                             ********************
                                             * COMPLETE AIRCRAFT *
                                             ********************
                     ALPHA      CCL          CMMC         CLL          CDITZ         CN            CCY
                      0.0     1.611030    -0.660342    0.000653     0.275885    -0.098765      0.038381
                      5.0     2.127566    -0.691494    0.000653     0.402769    -0.098765      0.038381
                     10.0     2.634245    -0.694866    0.000653     0.557059    -0.098765      0.038381
                     15.0     3.129518    -0.670279    0.000653     0.738042    -0.098765      0.038381
                     20.0     3.611976    -0.619939    0.000653     0.945353    -0.098765      0.038381


                              ******************************************************
                              *  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  *
                              ******************************************************


                              ******************************************************
                              *  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  *
                              ******************************************************
```

SAMPLE PROBLEM II
VECTORED JET CASE

```
                 ****************************************
                 *  STOL  AERODYNAMIC  METHODS  PROGRAM  *
                 ****************************************

      ***  STAMP SAMPLE CASE *  RECT WING + TAIL + FUSELAGE + VECTORED JETS  ***


                                 USED            INPUT
                    AREA =       0.600000        3.200000
                    SPAN =       2.000000        4.000000
                    CREF =       0.400000        0.800000
                     XMC =       0.100000        0.200000
                    CMAC =       0.400000        0.799999
                  ARATIO =       5.000001        5.000001
                     XCG =       0.200000        0.400000

                         NROWS  =  4           4
                         NCASES =  2           2
                         ISYMM  =  0           0
                         IPRINT =  0           0
                         JETFLG =  2           2
                         IGTYPE =  2           2
                         IHINGE =  0           0
                         IDERIV =  0           0
                         IGRND  =  0           0
                         IFFLD  =  0           0
                         NPOINT =  3           3
                         NROWH  =  4           4
                         ICAMH  =  1           1
                         NROWV  =  2           2
                         ICAMV  =  1           1
                         NSEG   =  4           4
                         ISECT  =  2           2
                         ICAMF  =  1           1

                         NUMBER OF WING ELEMENTS  =   12
                         NUMBER OF JET ELEMENTS   =    0
                         TOTAL NUMBER OF ELEMENTS =   12

          **********************************************************************
          *  ELEMENT GEOMETRY AND FUNDAMENTAL CASE DATA FOR ALL FUNDAMENTAL CASES  *
          **********************************************************************

          FUNDAMENTAL CASE  1 =  ALPHA CASE

          FUNDAMENTAL CASE  2 =    T.E. HINGE


  *** SECTION  1 ***  Y =  0.950000  DELTA =  0.050000  XLEAD =  0.0      XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                      .........................................INPUT VALUES..............................................
                          FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
           TWIST(DEG)     0.0       0.0
           ML   (Z/C)     0.0       0.0
  WING ELEMENTS       ..............................................EPS..............................................
  NO   X/C       X       DEL      FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
   1  0.0       0.0     0.20000  1.00000   0.0
   2  0.20000   0.08000 0.55000  1.00000   0.0
   3  0.75000   0.30000 0.25000  1.00000   0.0

 THIS ROW HAS NO JET

  *** SECTION  2 ***  Y =  0.800000  DELTA =  0.100000  XLEAD =  0.0      XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                      .........................................INPUT VALUES..............................................
                          FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
           TWIST(DEG)     0.0       0.0
           ML   (Z/C)     0.0       0.0
  WING ELEMENTS       ..............................................EPS..............................................
  NO   X/C       X       DEL      FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
   4  0.0       0.0     0.20000  1.00000   0.0
   5  0.20000   0.08000 0.55000  1.00000   0.0
   6  0.75000   0.30000 0.25000  1.00000   1.00000

 THIS ROW HAS NO JET

*** SECTION  3 ***  Y =  0.550000  DELTA =  0.150000  XLEAD =  0.0      XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                    .........................................INPUT VALUES..............................................
                        FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
         TWIST(DEG)     0.0       0.0
         ML   (Z/C)     0.0       0.0
WING ELEMENTS       ..............................................EPS..............................................
NO   X/C       X       DEL      FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
 7  0.0       0.0     0.20000  1.00000   0.0
 8  0.20000   0.08000 0.55000  1.00000   0.0
 9  0.75000   0.30000 0.25000  1.00000   1.00000

THIS ROW HAS NO JET

*** SECTION  4 ***  Y =  0.200000  DELTA =  0.200000  XLEAD =  0.0      XTRAIL =  0.400000  CHORD =  0.400000  TANLE =  0.0
                    .........................................INPUT VALUES..............................................
                        FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
         TWIST(DEG)     0.0       0.0
         ML   (Z/C)     0.0       0.0
WING ELEMENTS       ..............................................EPS..............................................
NO   X/C       X       DEL      FC1       FC2       FC3       FC4       FC5       FC6       FC7       FC8       FC9       FC10
10  0.0       0.0     0.20000  1.00000   0.0
11  0.20000   0.08000 0.55000  1.00000   0.0
12  0.75000   0.30000 0.25000  1.00000   0.0

THIS ROW HAS NO JET
```

```
 * STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
 NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                   ****************************************************
                                   *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                   ****************************************************

                                  SECTION  1      Y =  0.950000     CHORD =  0.400000
 WING    I      XB         CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
         1   0.0         0.106373   0.012789   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
         2   0.200000    0.043927   0.010885   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
         3   0.750000    0.009363   0.013951   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

                                        DETAILED LEADING EDGE LOADING
         1   0.040000    0.153173   0.019537
         2   0.080000    0.101331   0.014425
         3   0.120000    0.075358   0.012423
         4   0.160000    0.057695   0.011420
         5   0.200000    0.043927   0.010885


 * STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
 NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                   ****************************************************
                                   *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                   ****************************************************

                                  SECTION  2      Y =  0.300000     CHORD =  0.400000
 WING    I      XB         CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
         4   0.0         0.143983   0.020200   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
         5   0.200000    0.072476   0.014289   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
         6   0.750000    0.016661   0.059653   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

                                        DETAILED LEADING EDGE LOADING
         1   0.040000    0.209934   0.030276
         2   0.080000    0.142366   0.021621
         3   0.120000    0.109813   0.017878
         4   0.160000    0.088508   0.015714
         5   0.200000    0.072476   0.014289


 * STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
 NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                   ****************************************************
                                   *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                   ****************************************************

                                  SECTION  3      Y =  0.550000     CHORD =  0.400000
 WING    I      XB         CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
         7   0.0         0.166102   0.023660   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
         8   0.200000    0.088594   0.017398   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
         9   0.750000    0.022685   0.062539   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

                                        DETAILED LEADING EDGE LOADING
         1   0.040000    0.243181   0.035596
         2   0.080000    0.166230   0.025590
         3   0.120000    0.129673   0.021338
         4   0.160000    0.106093   0.018935
         5   0.200000    0.088594   0.017398


 * STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
 NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                   ****************************************************
                                   *  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  *
                                   ****************************************************

                                  SECTION  4      Y =  0.200000     CHORD =  0.400000
 WING    I      XB         CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10
        10   0.0         0.176832   0.016621   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
        11   0.200000    0.095896   0.011584   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
        12   0.750000    0.025554   0.005492   0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

                                        DETAILED LEADING EDGE LOADING
         1   0.040000    0.259207   0.024876
         2   0.080000    0.177600   0.017722
         3   0.120000    0.138997   0.014607
         4   0.160000    0.114209   0.012791
         5   0.200000    0.095895   0.011584


 * STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
 NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                   ************************************************
                                   *  WING SPANWISE LOADING FOR FUNDAMENTAL CASE  1  *
                                   ************************************************
                ... ........ LIFT .. .........        ............................ INDUCED DRAG ....... .... ......... ..........
 SECTION      Y       CLG      CLMU        CL     *    CDG       CDMU          CS         CD        CMU          GAMMA      ALFIN
    1     0.950000  0.041492  0.0       0.041492  *  0.0007242  0.0        0.0003950  0.0003292  0.0        0.0082965  0.0254734
    2     0.800000  0.062639  0.0       0.062639  *  0.0010933  0.0        0.0007237  0.0003696  0.0        0.0125279  0.0132666
    3     0.550000  0.075517  0.0       0.075517  *  0.0013180  0.0        0.0009631  0.0003550  0.0        0.0151034  0.0086523
    4     0.200000  0.081549  0.0       0.081549  *  0.0014233  0.0        0.0010915  0.0003318  0.0        0.0163097  0.0057904
          --------- --------- --------- --------- *  ---------- ---------- ---------- ---------- ---------- ---------- ----------
      TOTAL       0.071952  0.0       0.071952  *  0.0012558  0.0        0.0004098  0.0003460  0.0                   0.0063715

                       ..... .. .. PITCHING MOMENT ........ ...          ... LIFT CENTER ...
    SECTION      Y        CMG       CMMU       CMT         CM         *  *     XCP/C     XCL/C
       1     0.950000 -0.008679  0.0        0.0        -0.008679   *  *   0.209160  0.209160
       2     0.800000 -0.014214  0.0        0.0        -0.014214   *  *   0.226917  0.226917
       3     0.550000 -0.017846  0.0        0.0        -0.017846   *  *   0.236315  0.236315
       4     0.200000 -0.019537  0.0        0.0        -0.019537   *  *   0.239574  0.239574
             --------- --------- --------- ---------  ---------   *  *   --------- ---------
         TOTAL      -0.016879  0.0        0.0        -0.016879   (APEX)   0.234590  0.234590 (X/C,REF)
                     0.001109  0.0        0.0         0.001109   (XMC)    0.093838  0.093838 (X/B22)


 * STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
 NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                      NO STABILITY DERIVATIVES

                                   ************************************************
                                   *  WING SPANWISE LOADING FOR FUNDAMENTAL CASE  2  *
                                   ************************************************
                . .. ..... . LIFT .............        .. .. .......................... INDUCED DRAG ............... ..............
 SECTION      Y       CLG      CLMU        CL     *    CDG       CDMU          CS         CD        CMU          GAMMA      ALFIN
    1     0.950000  0.012220  0.0       0.012220  *  0.0        0.0        0.0000057 -0.0000057  0.0        0.0024440 -0.0005162
    2     0.800000  0.033259  0.0       0.033259  *  0.0001301  0.0        0.0000142  0.0001159  0.0        0.0066519  0.0100575
    3     0.550000  0.036272  0.0       0.036272  *  0.0001364  0.0        0.0000195  0.0001165  0.0        0.0072544  0.0099431
    4     0.200000  0.009865  0.0       0.009865  *  0.0        0.0        0.0000096 -0.0000096  0.0        0.0039730  0.0007637
          --------- --------- --------- --------- *  ---------- ---------- ---------- ---------- ---------- ---------- ----------
      TOTAL       0.022701  0.0       0.022701  *  0.0000670  0.0        0.0000131  0.0000538  0.0                   0.0000887

                       ......... . PITCHING MOMENT ..........            ... LIFT CENTER ...
    SECTION      Y        CMG       CMMU       CMT         CM         *  *     XCP/C     XCL/C
       1     0.950000 -0.004994  0.0        0.0        -0.004994   *  *   0.408638  0.408638
       2     0.800000 -0.017349  0.0        0.0        -0.017349   *  *   0.521623  0.521623
       3     0.550000 -0.018487  0.0        0.0        -0.018487   *  *   0.509676  0.509676
       4     0.200000 -0.002922  0.0        0.0        -0.002922   *  *   0.296215  0.296215
             --------- --------- --------- ---------  ---------   *  *   --------- ---------
         TOTAL      -0.010684  0.0        0.0        -0.010684   (APEX)   0.470633  0.470633 (X/C,REF)
                    -0.005009  0.0        0.0        -0.005009   (XMC)    0.188253  0.188253 (X/B22)
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                          NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                        ****************************************
                                        *  WING TOTAL AERODYNAMIC COEFFICIENTS  *
                                        ****************************************

              CASE 1     CASE 2     CASE 3     CASE 4     CASE 5     CASE 6     CASE 7     CASE 8     CASE 9     CASE 10

        CCLG  0.0719517  0.0227015  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CCLJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  **     CCL  0.0719517  0.0227015  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CCDG  0.0012558  0.0000670  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CCDJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CDS  0.0009098  0.0000131  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCD  0.0003460  0.0000538  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  ** CDITZ    0.0003215  0.0000887  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  **     CCJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CCMG -0.0168791 -0.0106841  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CCMJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CCMT  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CCM -0.0168791 -0.0106841  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CXCP  0.2345900  0.4706329  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CXCL  0.2345900  0.4706329  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

       CXCPB  0.0938359  0.1882531  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

       CXCLB  0.0938359  0.1882531  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

      CCMGMC  0.0011088 -0.0050087  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

      CCMJMC  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

      CCMTMC  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  ** CCMMC    0.0011088 -0.0050087  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CLLG  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

        CLLJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  *      CLL  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

         CNJ  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  *    CNGMC  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

  *      CCY  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                          NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                  ***************************************************
                                  *  WING CHORDWISE LOADING FOR COMPOSITE CASE   1  *
                                  ***************************************************
                                             FUNDAMENTAL CASE FACTORS
       A(1)         A(2)          A(3)        A(4)        A(5)        A(6)        A(7)        A(8)        A(9)        A(10)
       0.0        30.000000      0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0

    *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                  DO NOT PLOT THESE LOADING POINTS DIRECTLY

                                   SECTION  1      Y =  0.950000     CHORD =  0.400000
WING
     XB     0.0          0.200000     0.750000
    CPO     0.383684     0.326546     0.418529
    CPA     0.106373     0.043927     0.009363

                                   SECTION  2      Y =  0.800000     CHORD =  0.400000
WING
     XB     0.0          0.200000     0.750000
    CPO     0.605986     0.428660     1.789586
    CPA     0.143983     0.072476     0.016661

                                   SECTION  3      Y =  0.550000     CHORD =  0.400000
WING
     XB     0.0          0.200000     0.750000
    CPO     0.709814     0.521950     1.876159
    CPA     0.166102     0.088594     0.022685

                                   SECTION  4      Y =  0.200000     CHORD =  0.400000
WING
     XB     0.0          0.200000     0.750000
    CPO     0.498641     0.347524     0.164762
    CPA     0.176832     0.095896     0.025554
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                          NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                                  ***************************************************
                                  *  WING SPANWISE LOADING FOR COMPOSITE CASE   1  *
                                  ***************************************************
                                             FUNDAMENTAL CASE FACTORS
      A(1)        A(2)       A(3)        A(4)       A(5)        A(6)       A(7)        A(8)       A(9)       A(10)
      0.0       30.000000   0.0          0.0        0.0         0.0        0.0         0.0        0.0        0.0

                       ............ LIFT ............  *  *   ............ PITCHING MOMENT ............  *  *   ... LIFT CENTER ...
SECTION       Y        CLGO       CLMUO      CLO       *  *   CMGO       CMMUO      CM10       CMO       *  *   XCPO/C     XCLO/C
                       CLGA       CLMUA      CLA       *  *   CMGA       CMMUA      CM1A       CMA       *  *   XCPA/C     XCLA/C
   1       0.950000  0.366603   0.0        0.366603    *  *  -0.149808   0.0        0.0       -0.149808  *  *   0.408638   0.408638
                     0.041492   0.0        0.041492    *  *  -0.008679   0.0        0.0       -0.008679  *  *   0.209160   0.209160
   2       0.800000  0.997779   0.0        0.997779    *  *  -0.520465   0.0        0.0       -0.520465  *  *   0.521624   0.521624
                     0.062639   0.0        0.062639    *  *  -0.014214   0.0        0.0       -0.014214  *  *   0.226917   0.226917
   3       0.550000  1.088157   0.0        1.088157    *  *  -0.554607   0.0        0.0       -0.554607  *  *   0.509676   0.509676
                     0.075517   0.0        0.075517    *  *  -0.017846   0.0        0.0       -0.017846  *  *   0.236315   0.236315
   4       0.200000  0.295955   0.0        0.295955    *  *  -0.087666   0.0        0.0       -0.087666  *  *   0.296214   0.296214
                     0.081549   0.0        0.081549    *  *  -0.019537   0.0        0.0       -0.019537  *  *   0.239574   0.239574
         ---------  ---------  ---------  ---------    *  *  ---------  ---------  ---------  ---------  *  *  ---------  ---------
   TOTAL             0.681045   0.0        0.681045    *  *  -0.320522   0.0        0.0       -0.320522 (APEX)  0.470633   0.470633
                     0.071952   0.0        0.071952    *  *  -0.016879   0.0        0.0       -0.016879 (APEX)  0.234590   0.234590
                                                       *  *  -0.150261   0.0        0.0       -0.150261 (MAC)   0.188253   0.188253
                                                       *  *   0.001109   0.0        0.0        0.001109 (MAC)   0.093836   0.093836
```

```
                          ..................................  INDUCED DRAG  .....................................
SECTION       Y        CDG0       CDMU0      CS0        CD0        GAMMA0     ALFIN0     CT0        CMU
                       CDGA       CDMUA      CSA        CDA
                       CDGA2      CDMUA2     CSA2       CDA2       GAMMAA     ALFINA
   1       0.950000  0.0        0.0        0.0051387 -0.0051387  0.0733206 -0.0154871  0.0051387  0.0
                     0.0063984  0.0        0.0028493  0.0035491
                     0.0007242  0.0        0.0003950  0.0003292  0.0082985  0.0254214
   2       0.800000  0.1171281  0.0        0.0128183  0.1043097  0.1995557  0.3017246 -0.1043097  0.0
                     0.0185050  0.0        0.0060913  0.0124137
                     0.0010933  0.0        0.0007237  0.0003696  0.0125279  0.0132666
   3       0.550000  0.1227943  0.0        0.0175872  0.1052070  0.2176313  0.2982925 -0.1052070  0.0
                     0.0204766  0.0        0.0082311  0.0122456
                     0.0013180  0.0        0.0009631  0.0003550  0.0151034  0.0080523
   4       0.200000  0.0        0.0        0.0086793 -0.0086793  0.0591909  0.0229112  0.0086793  0.0
                     0.0051654  0.0        0.0061558 -0.0009904
                     0.0014233  0.0        0.0010915  0.0003318  0.0163097  0.0057904
                     ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
            TOTAL    0.0602639  0.0        0.0118254  0.0484385             0.0798660 -0.0484385  0.0
                     0.0125500  0.0        0.0064348  0.0061151             0.0090572
                     0.0012558  0.0        0.0009098  0.0003460             0.0003215
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                 * DOUGLAS AIRCRAFT COMPANY *                   * DATE 11/21/73   *   TIME  1.28.13
NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                          ***************************************************************
                          *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE   1  *
                          ***************************************************************

                              ALPHA=0        ALPHA           ALPHA**2

                   CCLG       0.681045       0.071952

                   CCLJ       0.0            0.0

             **     CCL       0.681045       0.071952

                   CCDG       0.0602639      0.0125500       0.0012558

                   CCDJ       0.0            0.0             0.0

                    CCS       0.0118254      0.0064348       0.0009098

                    CCD       0.0484385      0.0061151       0.0003460

             **  CDITZ        0.0798660      0.0090572       0.0003215

             **     CCJ       0.0

                   CCMG      -0.320522      -0.016879

                   CCMJ       0.0            0.0

                   CCMT       0.0            0.0

                    CCM      -0.320522      -0.016879

                   CXCP       0.470633       0.234590

                   CXCL       0.470633       0.234590

                  CXCPB       0.188253       0.093836

                  CXCLB       0.188253       0.093836

                 CCMGMC      -0.150261       0.001109

                 CCMJMC       0.0            0.0

                 CCMTMC       0.0            0.0

             **  CCMMC       -0.150261       0.001109

                   CLLG       0.0            0.0

                   CLLJ       0.0            0.0

             *      CLL       0.0            0.0

                    CNJ       0.0            0.0             0.0

             *    CNGMC       0.0            0.0             0.0

             *      CCY       0.0            0.0             0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                 * DOUGLAS AIRCRAFT COMPANY *                   * DATE 11/21/73   *   TIME  1.28.13
NO VECTORED JET OR JET SHEET                              NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                          ***************************************************************
                          *  TABULATED WING TOTAL COEFFICIENTS FOR COMPOSITE CASE   1  *
                          ***************************************************************

ALPHA       CCL        CCL**2     CCM(MC)     CCLJ        CLL            CDITZ       CT           CNJ        CN         CCY
-10.0   -0.0384721  0.0014801 -0.1613486  0.0         0.0          0.0214434 -0.0214434      0.0        0.0        0.0
 -9.0    0.0334796  0.0011209 -0.1602398  0.0         0.0          0.0243922 -0.0243922      0.0        0.0        0.0
 -8.0    0.1054313  0.0111158 -0.1591311  0.0         0.0          0.0279840 -0.0279840      0.0        0.0        0.0
 -7.0    0.1773830  0.0314647 -0.1580223  0.0         0.0          0.0322188 -0.0322188      0.0        0.0        0.0
 -6.0    0.2493347  0.0621678 -0.1569135  0.0         0.0          0.0370966 -0.0370966      0.0        0.0        0.0
 -5.0    0.3212864  0.1032249 -0.1558048  0.0         0.0          0.0426173 -0.0426173      0.0        0.0        0.0
 -4.0    0.3932381  0.1546361 -0.1546960  0.0         0.0          0.0487811 -0.0487811      0.0        0.0        0.0
 -3.0    0.4651898  0.2164015 -0.1535872  0.0         0.0          0.0555878 -0.0555878      0.0        0.0        0.0
 -2.0    0.5371414  0.2885209 -0.1524785  0.0         0.0          0.0630375 -0.0630375      0.0        0.0        0.0
 -1.0    0.6090931  0.3709944 -0.1513697  0.0         0.0          0.0711302 -0.0711302      0.0        0.0        0.0
  0.0    0.6810448  0.4638220 -0.1502610  0.0         0.0          0.0798660 -0.0798660      0.0        0.0        0.0
  1.0    0.7529965  0.5670037 -0.1491522  0.0         0.0          0.0892446 -0.0892446      0.0        0.0        0.0
  2.0    0.8249482  0.6805395 -0.1480435  0.0         0.0          0.0992664 -0.0992664      0.0        0.0        0.0
  3.0    0.8968999  0.8044294 -0.1469347  0.0         0.0          0.1099309 -0.1099309      0.0        0.0        0.0
  4.0    0.9688516  0.9386733 -0.1458259  0.0         0.0          0.1212386 -0.1212386      0.0        0.0        0.0
  5.0    1.0408030  1.0832701 -0.1447172  0.0         0.0          0.1331893 -0.1331893      0.0        0.0        0.0
  6.0    1.1127548  1.2382231 -0.1436084  0.0         0.0          0.1457829 -0.1457829      0.0        0.0        0.0
  7.0    1.1847057  1.4035273 -0.1424996  0.0         0.0          0.1590195 -0.1590195      0.0        0.0        0.0
  8.0    1.2566576  1.5791874 -0.1413909  0.0         0.0          0.1728992 -0.1728992      0.0        0.0        0.0
  9.0    1.3286095  1.7652025 -0.1402821  0.0         0.0          0.1874217 -0.1874217      0.0        0.0        0.0
 10.0    1.4005613  1.9615717 -0.1391733  0.0         0.0          0.2025874 -0.2025874      0.0        0.0        0.0
 11.0    1.4725132  2.1682949 -0.1380646  0.0         0.0          0.2183959 -0.2183959      0.0        0.0        0.0
 12.0    1.5444651  2.3853722 -0.1369558  0.0         0.0          0.2348475 -0.2348475      0.0        0.0        0.0
 13.0    1.6164160  2.6128006 -0.1358470  0.0         0.0          0.2519420 -0.2519420      0.0        0.0        0.0
 14.0    1.6883678  2.8505859 -0.1347383  0.0         0.0          0.2696795 -0.2696795      0.0        0.0        0.0
 15.0    1.7603197  3.0987253 -0.1336295  0.0         0.0          0.2880601 -0.2880601      0.0        0.0        0.0
 16.0    1.8322716  3.3572187 -0.1325208  0.0         0.0          0.3070837 -0.3070837      0.0        0.0        0.0
 17.0    1.9042225  3.6260624 -0.1314120  0.0         0.0          0.3267501 -0.3267501      0.0        0.0        0.0
 18.0    1.9761744  3.9052649 -0.1303033  0.0         0.0          0.3470596 -0.3470596      0.0        0.0        0.0
 19.0    2.0481262  4.1948204 -0.1291945  0.0         0.0          0.3680121 -0.3680121      0.0        0.0        0.0
 20.0    2.1200781  4.4947309 -0.1280857  0.0         0.0          0.3896075 -0.3896075      0.0        0.0        0.0
 21.0    2.1920300  4.8049946 -0.1269770  0.0         0.0          0.4118460 -0.4118460      0.0        0.0        0.0
 22.0    2.2639809  5.1256084 -0.1258682  0.0         0.0          0.4347274 -0.4347274      0.0        0.0        0.0
 23.0    2.3359327  5.4565811 -0.1247594  0.0         0.0          0.4582519 -0.4582519      0.0        0.0        0.0
 24.0    2.4078846  5.7979078 -0.1236507  0.0         0.0          0.4824193 -0.4824193      0.0        0.0        0.0
 25.0    2.4798365  6.1495886 -0.1225419  0.0         0.0          0.5072297 -0.5072297      0.0        0.0        0.0
 26.0    2.5517883  6.5116234 -0.1214331  0.0         0.0          0.5326831 -0.5326831      0.0        0.0        0.0
 27.0    2.6237392  6.8840075 -0.1203244  0.0         0.0          0.5587794 -0.5587794      0.0        0.0        0.0
 28.0    2.6956911  7.2667503 -0.1192156  0.0         0.0          0.5855188 -0.5855188      0.0        0.0        0.0
 29.0    2.7676430  7.6598473 -0.1181068  0.0         0.0          0.6129012 -0.6129012      0.0        0.0        0.0
 30.0    2.8395948  8.0632982 -0.1169981  0.0         0.0          0.6409265 -0.6409265      0.0        0.0        0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                       NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                                   **********************************
                                   * HORIZONTAL TAIL GEOMETRY SUMMARY *
                                   **********************************

                                       ASPECT RATIO =      2.666599
                                        TAPER RATIO =      1.000000
                                              SWEEP =      0.0
                                               SPAN =      0.600000
                                               AREA =      0.240006
                                                MAC =      0.300007
                                                XMC =      1.075002
                                                YMC =      0.0
                                             LENGTH =      0.975002
                                             HEIGHT =      0.300000
                                             VOLUME =      0.731269
                                              NROWH =      4

                              ROW    YLEAD         XLEAD         XTRAIL        CHORD
                               1     0.350000      1.000000      1.300007      0.300007
                               2     0.150000      1.000000      1.300007      0.300007
                               3    -0.150000      1.000000      1.300007      0.300007
                               4    -0.350000      1.000000      1.300007      0.300007


                                   ********************************
                                   * VERTICAL TAIL GEOMETRY SUMMARY *
                                   ********************************

                                       ASPECT RATIO =      1.333300
                                        TAPER RATIO =      1.000000
                                              SWEEP =      0.0
                                               SPAN =      0.400000
                                               AREA =      0.120003
                                                MAC =      0.300007
                                                XMC =      1.075002
                                                ZMC =      0.200000
                                             LENGTH =      0.975002
                                             HEIGHT =      0.0
                                             VOLUME =      0.073127
                                              NROWV =      2

                              ROW    ZLEAD         XLEAD         XTRAIL        CHORD
                               1     0.350000      1.000000      1.300007      0.300007
                               2     0.150000      1.000000      1.300007      0.300007

* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                       NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                              ****************************************
                              *        ISOLATED TAIL ANALYSIS        *
                              *  CHORDWISE LOADING FOR ALL TAIL CASES  *
                              ****************************************

                                     CASE 1 IS THE ALPHA  CASE
                                     CASE 2 IS THE  BETA  CASE
                                     CASE 3 IS THE CAMBER CASE

                                        ******************
                                        * HORIZONTAL TAIL *
                                        ******************

                      SECTION  1       Y =   0.350000       CHORD =    0.300007
           ELEMENT      X/C          CASE 1        CASE 2        CASE 3        CASE 4        CASE 5
              1         0.0          0.071706      0.013105      0.316012
              2         0.500000     0.017510      0.006195      0.530445

                      SECTION  2       Y =   0.150000       CHORD =    0.300007
           ELEMENT      X/C          CASE 1        CASE 2        CASE 3        CASE 4        CASE 5
              3         0.0          0.102980      0.025675      0.547442
              4         0.500000     0.030074      0.012695      0.743646

                      SECTION  3       Y =  -0.150000       CHORD =    0.300007
           ELEMENT      X/C          CASE 1        CASE 2        CASE 3        CASE 4        CASE 5
              5         0.0          0.102980     -0.025675      0.228755
              6         0.500000     0.030074     -0.012695      0.541081

                      SECTION  4       Y =  -0.350000       CHORD =    0.300007
           ELEMENT      X/C          CASE 1        CASE 2        CASE 3        CASE 4        CASE 5
              7         0.0          0.071706     -0.013105      0.146684
              8         0.500000     0.017510     -0.006195      0.438058

                                        ****************
                                        * VERTICAL TAIL *
                                        ****************

                      SECTION  1       Z =   0.350000       CHORD =    0.300007
           ELEMENT      X/C          CASE 1        CASE 2        CASE 3        CASE 4        CASE 5
              1         0.0         -0.000000      0.054619      0.124678
              2         0.500000    -0.000000      0.008603      0.414898

                      SECTION  2       Z =   0.150000       CHORD =    0.300007
           ELEMENT      X/C          CASE 1        CASE 2        CASE 3        CASE 4        CASE 5
              3         0.0         -0.000000      0.116188      0.487885
              4         0.500000    -0.000000      0.038005      0.696890

* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                       NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                         ************************************************
                         *             ISOLATED TAIL ANALYSIS             *
                         *  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES  *
                         ************************************************

                                        ******************
                                        * HORIZONTAL TAIL *
                                        ******************

          CASE 1                     CASE 2                     CASE 3                      CASE 4                      CASE 5
K    CLH     CMH     CDIH      CLH     CMH     CDIH      CLH     CMH     CDIH      CLH      CMH      CDIH      CLH      CMH      CDIH
1  0.0446 -0.0100  0.0006   0.0096 -0.0028 -0.0000   0.4232 -0.1855  0.0362
2  0.0665 -0.0158  0.0007   0.0192 -0.0056 -0.0001   0.6455 -0.2666  0.0395
3  0.0665 -0.0158  0.0007  -0.0192  0.0056 -0.0001   0.3849 -0.1834  0.0429
4  0.0446 -0.0100  0.0006  -0.0096  0.0028 -0.0000   0.2924 -0.1461  0.0363
--------------------------------------------------------------------------------------------------------------------------------
   0.0183 -0.0444  0.0002  -0.0000  0.0000 -0.0000   0.1428 -0.3685  0.0120

     CLLH =   0.000000        CLLH =  -0.000901        CLLH =  -0.006116        CLLH =

                                        ****************
                                        * VERTICAL TAIL *
                                        ****************

          CASE 1                     CASE 2                     CASE 3                      CASE 4                      CASE 5
K    CYV     CNV     CDIV      CYV     CNV     CDIV      CYV     CNV     CDIV      CYV      CNV      CDIV      CYV      CNV      CDIV
1 -0.0000  0.0000  0.0000   0.0316 -0.0061  0.0004   0.2690 -0.1374  0.0338
2 -0.0000  0.0000 -0.0000   0.0781 -0.0193  0.0007   0.5924 -0.2483  0.0402
--------------------------------------------------------------------------------------------------------------------------------
  -0.0000  0.0000 -0.0000   0.0050 -0.0121  0.0000   0.0384 -0.0988  0.0029

     CLLV =  -0.000000        CLLV =   0.000866        CLLV =   0.006769        CLLV =

     *** NOTE *** ALL TAIL COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                      NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                              ***********************************************
                              *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                              ***********************************************

                                        ALPHA =   0.0  DEGREES

            X          Y          Z        U/U INF     V/U INF     W/U INF   DOWNWASH   SIDEWASH
                                                                              ANGLE      ANGLE
                                                                            (DEGREES)  (DEGREES)

  1     1.112502   0.350000   0.300000   0.000326   -0.001435   -0.044156   2.527516   -0.082172
  2     1.262506   0.350000   0.300000  -0.001010   -0.006476   -0.042073   2.411643   -0.371392
  3     1.112502   0.150000   0.300000   0.006222    0.003963   -0.035991   2.048498    0.225648
  4     1.262506   0.150000   0.300000   0.004038    0.001292   -0.037512   2.139629    0.073730
  5     1.112502  -0.150000   0.300000   0.006222   -0.003963   -0.035991   2.048499   -0.225648
  6     1.262506  -0.150000   0.300000   0.004038   -0.001292   -0.037512   2.139629   -0.073730
  7     1.112502  -0.350000   0.300000   0.000326    0.001435   -0.044156   2.527517    0.082172
  8     1.262506  -0.350000   0.300000  -0.001010    0.006476   -0.042073   2.411643    0.371392
  9     1.112502   0.0        0.350000   0.006963    0.0        -0.032787   1.864886    0.0
 10     1.262506   0.0        0.350000   0.004710    0.0        -0.034394   1.960602    0.0
 11     1.112502   0.0        0.150000   0.011247    0.0        -0.023951   1.356768    0.0
 12     1.262506   0.0        0.150000   0.008619    0.0        -0.028872   1.639633    0.0
 13    -0.475000   0.0       -0.075000  -0.004612    0.0         0.020086  -1.156031    0.0
 14    -0.375000   0.0       -0.075000  -0.006053    0.0         0.025570  -1.473643    0.0
 15     0.350000   0.0       -0.075000  -0.004126    0.0        -0.000739   0.042527    0.0
 16     1.150000   0.0       -0.075000   0.020724    0.0         0.020744  -1.164256    0.0
 17    -0.500000   0.125000   0.125000   0.004208   -0.000061    0.018340  -1.046292   -0.003497
 18     0.0        0.125000   0.125000   0.065225    0.002677    0.041891  -2.252047    0.143987
 19     0.500000   0.125000   0.125000   0.031261    0.027601   -0.008606   0.478119    1.255461
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                      NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                              ***********************************************
                              *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                              ***********************************************

                                        ALPHA =   5.00 DEGREES

            X          Y          Z        U/U INF     V/U INF     W/U INF   DOWNWASH   SIDEWASH
                                                                              ANGLE      ANGLE
                                                                            (DEGREES)  (DEGREES)

  1     1.112502   0.350000   0.300000  -0.003818   -0.035622   -0.082609   4.740434   -2.047944
  2     1.262506   0.350000   0.300000  -0.005540   -0.044563   -0.080425   4.623631   -2.565798
  3     1.112502   0.150000   0.300000   0.005117   -0.006477   -0.086002   4.890537   -0.369207
  4     1.262506   0.150000   0.300000   0.002025   -0.010228   -0.088965   5.073703   -0.584824
  5     1.112502  -0.150000   0.300000   0.005117    0.006477   -0.086002   4.890541    0.369201
  6     1.262506  -0.150000   0.300000   0.002025    0.010228   -0.088965   5.073703    0.584824
  7     1.112502  -0.350000   0.300000  -0.003818    0.035622   -0.082609   4.740434    2.047944
  8     1.262506  -0.350000   0.300000  -0.005540    0.044563   -0.080425   4.623634    2.565798
  9     1.112502   0.0        0.350000   0.006564    0.0        -0.078387   4.452945    0.0
 10     1.262506   0.0        0.350000   0.003354    0.0        -0.081070   4.619386    0.0
 11     1.112502   0.0        0.150000   0.012885    0.0        -0.078677   4.441613    0.0
 12     1.262506   0.0        0.150000   0.009319    0.0        -0.085046   4.816382    0.0
 13    -0.475000   0.0       -0.075000  -0.004492    0.0         0.034623  -1.991918    0.0
 14    -0.375000   0.0       -0.075000  -0.006678    0.0         0.045110  -2.600180    0.0
 15     0.350000   0.0       -0.075000  -0.023763    0.0        -0.066555   3.900099    0.0
 16     1.150000   0.0       -0.075000   0.028560    0.0        -0.012213   0.680295    0.0
 17    -0.500000   0.125000   0.125000   0.011255   -0.000295    0.029827  -1.689450   -0.016725
 18     0.0        0.125000   0.125000   0.153738    0.001204    0.046873  -2.327481    0.059815
 19     0.500000   0.125000   0.125000   0.041370    0.022070   -0.072441   3.979267    1.214074
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                      NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                              ***********************************************
                              *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                              ***********************************************

                                        ALPHA =  10.00 DEGREES

            X          Y          Z        U/U INF     V/U INF     W/U INF   DOWNWASH   SIDEWASH
                                                                              ANGLE      ANGLE
                                                                            (DEGREES)  (DEGREES)

  1     1.112502   0.350000   0.300000  -0.012780   -0.080780   -0.127335   7.349600   -4.677842
  2     1.262506   0.350000   0.300000  -0.014963   -0.097820   -0.127356   7.366908   -5.671184
  3     1.112502   0.150000   0.300000  -0.001127   -0.016584   -0.142745   8.132853   -0.951173
  4     1.262506   0.150000   0.300000  -0.005209   -0.021102   -0.149088   8.523419   -1.215191
  5     1.112502  -0.150000   0.300000  -0.001127    0.016584   -0.142745   8.132855    0.951172
  6     1.262506  -0.150000   0.300000  -0.005209    0.021102   -0.149088   8.523423    1.215187
  7     1.112502  -0.350000   0.300000  -0.012780    0.080780   -0.127335   7.349604    4.677838
  8     1.262506  -0.350000   0.300000  -0.014963    0.097819   -0.127356   7.366911    5.671181
  9     1.112502   0.0        0.350000   0.001543    0.0        -0.128791   7.327618    0.0
 10     1.262506   0.0        0.350000  -0.002608    0.0        -0.133726   7.636455    0.0
 11     1.112502   0.0        0.150000   0.010174    0.0        -0.133277   7.515871    0.0
 12     1.262506   0.0        0.150000   0.005973    0.0        -0.140629   7.958051    0.0
 13    -0.475000   0.0       -0.075000  -0.001464    0.0         0.049127  -2.816646    0.0
 14    -0.375000   0.0       -0.075000  -0.003471    0.0         0.064746  -3.717343    0.0
 15     0.350000   0.0       -0.075000  -0.053502    0.0        -0.128063   7.705405    0.0
 16     1.150000   0.0       -0.075000   0.034434    0.0        -0.035911   1.988238    0.0
 17    -0.500000   0.125000   0.125000   0.020671   -0.000573    0.040047  -2.246909   -0.032166
 18     0.0        0.125000   0.125000   0.242845   -0.000362    0.036586  -1.686157   -0.016690
 19     0.500000   0.125000   0.125000   0.043112    0.022066   -0.137953   7.533731    1.211859
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                      NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                              ***********************************************
                              *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                              ***********************************************

                                        ALPHA =  15.00 DEGREES

            X          Y          Z        U/U INF     V/U INF     W/U INF   DOWNWASH   SIDEWASH
                                                                              ANGLE      ANGLE
                                                                            (DEGREES)  (DEGREES)

  1     1.112502   0.350000   0.300000  -0.027890   -0.141836   -0.181886  10.597734   -8.301155
  2     1.262506   0.350000   0.300000  -0.031360   -0.174758   -0.189825  11.087752  -10.227026
  3     1.112502   0.150000   0.300000  -0.013295   -0.024634   -0.206527  11.821868   -1.430117
  4     1.262506   0.150000   0.300000  -0.018678   -0.028396   -0.218406  12.547436   -1.657483
  5     1.112502  -0.150000   0.300000  -0.013295    0.024633   -0.206527  11.821872    1.430114
  6     1.262506  -0.150000   0.300000  -0.018678    0.028396   -0.218406  12.547436    1.657476
  7     1.112502  -0.350000   0.300000  -0.027890    0.141836   -0.181886  10.597734    8.301155
  8     1.262506  -0.350000   0.300000  -0.031360    0.174758   -0.189825  11.087756   10.227022
  9     1.112502   0.0        0.350000  -0.008576    0.0        -0.183619  10.492706    0.0
 10     1.262506   0.0        0.350000  -0.013739    0.0        -0.191863  11.008556    0.0
 11     1.112502   0.0        0.150000   0.003659    0.0        -0.184806  10.433144    0.0
 12     1.262506   0.0        0.150000  -0.000526    0.0        -0.191216  10.830767    0.0
 13    -0.475000   0.0       -0.075000   0.004440    0.0         0.063187  -3.599605    0.0
 14    -0.375000   0.0       -0.075000   0.003547    0.0         0.083942  -4.781398    0.0
 15     0.350000   0.0       -0.075000  -0.092701    0.0        -0.184011  11.464758    0.0
 16     1.150000   0.0       -0.075000   0.038951    0.0        -0.051136   2.817770    0.0
 17    -0.500000   0.125000   0.125000   0.032271   -0.000895    0.048670  -2.699384   -0.049674
 18     0.0        0.125000   0.125000   0.337195   -0.002023    0.010806  -0.464722   -0.086993
 19     0.500000   0.125000   0.125000   0.036336    0.022672   -0.204119  11.142468    1.253280
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                        NO GROUND EFFECT                        NO STABILITY DERIVATIVES

                              ***********************************************
                              *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                              ***********************************************

                                           ALPHA =  20.00 DEGREES

            X            Y            Z          U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                      ANGLE       ANGLE
                                                                                    (DEGREES)   (DEGREES)

    1    1.112502     0.350000     0.300000    -0.051847   -0.226772   -0.253860   14.988873  -13.450906
    2    1.262506     0.350000     0.300000    -0.059899   -0.290312   -0.285185   16.875458  -17.161194
    3    1.112502     0.150000     0.300000    -0.032125   -0.027835   -0.277072   15.974759   -1.647336
    4    1.262506     0.150000     0.300000    -0.039229   -0.027236   -0.296077   17.127502   -1.623774
    5    1.112502    -0.150000     0.300000    -0.032125    0.027835   -0.277072   15.974766    1.647333
    6    1.262506    -0.150000     0.300000    -0.039229    0.027236   -0.296077   17.127502    1.623770
    7    1.112502    -0.350000     0.300000    -0.051847    0.226772   -0.253860   14.988873   13.450902
    8    1.262506    -0.350000     0.300000    -0.059899    0.290312   -0.285185   16.875473   17.161194
    9    1.112502     0.0          0.350000    -0.024133    0.0         -0.242043   13.929907    0.0
   10    1.262506     0.0          0.350000    -0.030362    0.0         -0.254119   14.685565    0.0
   11    1.112502     0.0          0.150000    -0.005600    0.0         -0.230018   13.074230    0.0
   12    1.262506     0.0          0.150000    -0.008501    0.0         -0.232011   13.170239    0.0
   13   -0.475000     0.0         -0.075000     0.013138    0.0          0.076405   -4.312776    0.0
   14   -0.375000     0.0         -0.075000     0.014281    0.0          0.102180   -5.752671    0.0
   15    0.350000     0.0         -0.075000    -0.140543    0.0         -0.233212   15.181535    0.0
   16    1.150000     0.0         -0.075000     0.042218    0.0         -0.059861    3.287243    0.0
   17   -0.500000     0.125000     0.125000     0.045836   -0.001267    0.055342   -3.031808   -0.069134
   18    0.0          0.125000     0.125000     0.419490   -0.003780   -0.030572    1.233813   -0.152563
   19    0.500000     0.125000     0.125000     0.021010    0.023980   -0.269882   14.806288    1.345434
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                        NO GROUND EFFECT                        NO STABILITY DERIVATIVES

                                  *****************************************
                                  *        INTERFERENCE TAIL ANALYSIS     *
                                  *  CHORDWISE LOADING FOR ALL TAIL CASES *
                                  *****************************************

                                       CASE 1 IS FOR ALPHA =  0.0
                                       CASE 2 IS FOR ALPHA =  5.00
                                       CASE 3 IS FOR ALPHA = 10.00
                                       CASE 4 IS FOR ALPHA = 15.00
                                       CASE 5 IS FOR ALPHA = 20.00

                                          ******************
                                          * HORIZONTAL TAIL *
                                          ******************

                   SECTION  1       Y =   0.350000      CHORD =    0.300007
         ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
            1        0.0           0.150849     0.326232     0.470304     0.571661     0.609945
            2        0.500000      0.495017     0.533424     0.558632     0.562556     0.525454

                   SECTION  2       Y =   0.150000      CHORD =    0.300007
         ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
            3        0.0           0.331088     0.553389     0.729950     0.853843     0.920047
            4        0.500000      0.675532     0.736157     0.777301     0.795924     0.791343

                   SECTION  3       Y =  -0.150000      CHORD =    0.300007
         ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
            5        0.0           0.012400     0.234702     0.411262     0.535155     0.601360
            6        0.500000      0.472967     0.533593     0.574736     0.593359     0.588779

                   SECTION  4       Y =  -0.350000      CHORD =    0.300007
         ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
            7        0.0          -0.018479     0.156904     0.300976     0.402333     0.440617
            8        0.500000      0.402629     0.441036     0.466245     0.470168     0.433066

                                          *****************
                                          * VERTICAL TAIL *
                                          *****************

                   SECTION  1       Z =   0.350000      CHORD =    0.300007
         ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
            1        0.0           0.124678     0.124678     0.124678     0.124678     0.124678
            2        0.500000      0.414898     0.414898     0.414898     0.414898     0.414898

                   SECTION  2       Z =   0.150000      CHORD =    0.300007
         ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
            3        0.0           0.487885     0.487885     0.487885     0.487885     0.487885
            4        0.500000      0.696890     0.696890     0.696890     0.696890     0.696890
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *          * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                        NO GROUND EFFECT                        NO STABILITY DERIVATIVES

                           ***********************************************
                           *            INTERFERENCE TAIL ANALYSIS       *
                           *  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES  *
                           ***********************************************

                                          ******************
                                          * HORIZONTAL TAIL *
                                          ******************

          CASE 1                    CASE 2                    CASE 3                    CASE 4                    CASE 5
K   CLH     CMH     CDIH      CLH     CMH     CDIH      CLH     CMH     CDIH      CLH     CMH     CDIH      CLH     CMH     CDIH
1  0.3229 -0.1641  0.0150    0.4298 -0.1871  0.0398    0.5145 -0.2040  0.0667    0.5671 -0.2115  0.0862    0.5677 -0.2023  0.0812
2  0.5033 -0.2318  0.0176    0.6448 -0.2646  0.0391    0.7536 -0.2885  0.0599    0.8249 -0.3021  0.0757    0.8557 -0.3048  0.0844
3  0.2427 -0.1486  0.0241    0.3841 -0.1814  0.0424    0.4930 -0.2053  0.0606    0.5643 -0.2189  0.0744    0.5951 -0.2216  0.0834
4  0.1921 -0.1247  0.0182    0.2990 -0.1476  0.0392    0.3836 -0.1645  0.0628    0.4363 -0.1721  0.0799    0.4368 -0.1629  0.0751
------------------------------------------------------------------------------------------------------------------------------------
   0.1032 -0.2725  0.0059    0.1431 -0.3584  0.0121    0.1739 -0.4148  0.0184    0.1939 -0.4385  0.0231    0.2009 -0.4279  0.0245

     CLLH =  -0.006116         CLLH =  -0.006116         CLLH =  -0.006116         CLLH =  -0.006116         CLLH =  -0.006116

                                          *****************
                                          * VERTICAL TAIL *
                                          *****************

          CASE 1                    CASE 2                    CASE 3                    CASE 4                    CASE 5
K   CYV     CNV     CDIV      CYV     CNV     CDIV      CYV     CNV     CDIV      CYV     CNV     CDIV      CYV     CNV     CDIV
1  0.2698 -0.1374  0.0338    0.2698 -0.1374  0.0338    0.2698 -0.1374  0.0338    0.2698 -0.1374  0.0338    0.2698 -0.1374  0.0338
2  0.5924 -0.2483  0.0402    0.5924 -0.2483  0.0402    0.5924 -0.2483  0.0402    0.5924 -0.2483  0.0402    0.5924 -0.2483  0.0402
------------------------------------------------------------------------------------------------------------------------------------
   0.0384 -0.0988  0.0029    0.0384 -0.0988  0.0029    0.0384 -0.0988  0.0029    0.0384 -0.0988  0.0029    0.0384 -0.0988  0.0029

     CLLV =   0.006769         CLLV =   0.006769         CLLV =   0.006769         CLLV =   0.006769         CLLV =   0.006769

      *** NOTE *** ALL TAIL COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                          NO GROUND EFFECT                              NO STABILITY DERIVATIVES

                               ****************************************************************
                               * SECTIONAL AND TOTAL FUSELAGE LOADING FROM PRESSURE INTEGRATION *
                               *                     FOR COMPOSITE CASE    1                    *
                               ****************************************************************

 SEG      X         CM( 0 )    CN( 0 )     CM( 5 )    CN( 5 )     CM( 10 )   CN( 10 )    CM( 15 )   CN( 15 )    CM( 20 )   CN( 20 )

   1  -0.475000  -0.001589    0.0        0.007546   0.0         0.015471   0.0         0.021916   0.0         0.026679   0.0
   2  -0.375000   0.001819    0.0        0.015365   0.0         0.028231   0.0         0.039955   0.0         0.050128   0.0
   3   0.350000   0.000000    0.0        0.000000   0.0         0.000000   0.0        -0.000000   0.0        -0.000000   0.0
   4   1.150000   0.011876    0.0        0.037487   0.0         0.066466   0.0         0.097328   0.0         0.128026   0.0
---------------------------------------------------------------------------------------------------------------------------------
 TOTAL            0.012106    0.0        0.060398   0.0         0.110168   0.0         0.159199   0.0         0.204832   0.0

      *** NOTE *** ALL FUSELAGE COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF


                                             ************************************
                                             *  POWER ON VECTORED JET CASE   1  *
                                             ************************************


                                             ************************************
                                             *  VECTORED JET CHARACTERISTICS  *
                                             ************************************
             CJ          UJO        DO/B/2      THO         DIHED       ISY    XO/B/2      YO/B/2      ZO/B/2      DELZ
          0.500000    5.046270    0.100000   40.000000    0.0           0    0.200000    0.250000   -0.100000    0.009487


   * STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
   VECTORED JET CASE NUMBER   1                          NO GROUND EFFECT                              NO STABILITY DERIVATIVES
                                          ********************************************
                                          *  CHORDWISE LOADING FOR COMPOSITE CASE  1  *
                                          *  WITH VECTORED THRUST AT ALPHA =     0.0  *
                                          ********************************************
                                                   FUNDAMENTAL CASE FACTORS
             A( 1 )       A( 2 )       A( 3 )       A( 4 )       A( 5 )       A( 6 )       A( 7 )       A( 8 )       A( 9 )       A( 10 )
          0.0         30.000000    0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0

      *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                    DO NOT PLOT THESE LOADING POINTS DIRECTLY

                                       SECTION   1      Y =  0.950000      CHORD =  0.400000
 WING
       XB     0.0          0.200000     0.750000
       CP     0.358204     0.307415     0.407241

                                       SECTION   2      Y =  0.800000      CHORD =  0.400000
 WING
       XB     0.0          0.200000     0.750000
       CP     0.559178     0.390819     1.766351

                                       SECTION   3      Y =  0.550000      CHORD =  0.400000
 WING
       XB     0.0          0.200000     0.750000
       CP     0.619960     0.441497     1.811483

                                       SECTION   4      Y =  0.200000      CHORD =  0.400000
 WING
       XB     0.0          0.200000     0.750000
       CP     0.303243     0.057624    -0.054405


                                    ********************************************
                                    *  SPANWISE LOADING FOR COMPOSITE CASE  1  *
                                    *  WITH VECTORED THRUST AT ALPHA =    0.0  *
                                    ********************************************
                                             FUNDAMENTAL CASE FACTORS
                                      A( 4 )      A( 5 )      A( 6 )      A( 7 )      A( 8 )      A( 9 )      A( 10 )
           A( 1 )      A( 2 )      A( 3 )     0.0         0.0         0.0         0.0         0.0         0.0         0.0
         0.0        30.000000    0.0
                                                       * *                                                         * *     ... LIFT CENTER ...
                   ............ LIFT .............     * *     ........... PITCHING MOMENT ...........             * *       XCP/C      XCL/C
  SECTION    Y          CLG         CLMU       CL       * *      CMG        CMMU       CMT        CM                * *     0.412877   0.412877
     1     0.950000   0.349817   0.0        0.349817    * *    -0.144431   0.0        0.0       -0.144431            * *     0.528164   0.528164
     2     0.800000   0.964933   0.0        0.964933    * *    -0.509642   0.0        0.0       -0.509642            * *     0.520393   0.520393
     3     0.550000   1.014146   0.0        1.014146    * *    -0.527755   0.0        0.0       -0.527755            * *    -0.062344  -0.062344
     4     0.200000   0.060495   0.0        0.060495    * *     0.003772   0.0        0.0        0.003772            * *    ---------  ---------
          ---------  ---------  ---------  ---------   * *    ---------  ---------  ---------  ---------            * *
             TOTAL    0.556410   0.642788   1.199197    * *    -0.273189  -0.321394   0.191511  -0.403072  (APEX)          0.490986   0.495818
                                                        * *    -0.134087  -0.160697   0.191511  -0.103273  (XMC)           0.196394   0.198327


                          .......................................... INDUCED DRAG ...............
  SECTION      Y          COG          COMU           CS          CD           GAMMA        ALFIN          CT            CMU
     1      0.950000   0.0          0.0          0.0044789  -0.0044789   0.0699635  -0.0128068
     2      0.800000   0.1156073    0.0          0.0109146   0.1046927   0.1929865   0.3118570
     3      0.550000   0.1185613    0.0          0.0134164   0.1051449   0.2028291   0.3128834
     4      0.200000   0.0          0.0          0.0032099  -0.0032099   0.0120991  -0.0004357
                       ----------  ----------   ----------  ----------  ----------  ----------  ----------  ----------
             TOTAL     0.0586898   0.2339555    0.0079397   0.2847056               0.0774592   0.7152944   1.0000000
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *           * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                   .    NO GROUND EFFECT                         NO STABILITY DERIVATIVES

                    **********************************************************
                    *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                    *            WITH VECTORED THRUST AT ALPHA =   0.0              *
                    **********************************************************

                                      CCLG      0.5564100

                                      CCLJ      0.6427876

                                 **    CCL      1.1991968

                                      CCDG      0.0586898

                                      CCDJ      0.2339555

                                       CCS      0.0079397

                                       CCD      0.2847056

                                 ** CDITZ       0.0774592

                                 **    CCJ      1.0000000

                                      CCMG     -0.2731894

                                      CCMJ     -0.3213937

                                      CCMT      0.1915109

                                       CCM     -0.4030722

                                      CXCP      0.4909857

                                      CXCL      0.4958177

                                     CXCPB      0.1963943

                                     CXCLB      0.1983271

                                    CCMGMC     -0.1340870

                                    CCMJMC     -0.1606968

                                    CCMTMC      0.1915109

                                 ** CCMMC      -0.1032729

                                      CLLG      0.0

                                      CLLJ      0.0

                                 *     CLL      0.0

                                       CNJ      0.0

                                 * . CNGMC      0.0

                                 *     CCY      0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *            * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                           NO GROUND EFFECT                            NO STABILITY DERIVATIVES
                                          ******************************************
                                          *  CHORDWISE LOADING FOR COMPOSITE CASE  1  *
                                          *  WITH VECTORED THRUST AT ALPHA =     5.0  *
                                          ******************************************
                                                 FUNDAMENTAL CASE FACTORS
          A( 1)       A( 2)       A( 3)       A( 4)       A( 5)       A( 6)       A( 7)       A( 8)       A( 9)       A( 10)
        5.000000   30.000000    0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0

     *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                   DO NOT PLOT THESE LOADING POINTS DIRECTLY

                                   SECTION  1      Y =  0.950000     CHORD =  0.400000
WING
     XB    0.0          0.200000    0.750000
     CP    0.888457     0.524641    0.451749

                                   SECTION  2      Y =  0.800000     CHORD =  0.400000
WING
     XB    0.0          0.200000    0.750000
     CP    1.275485     0.748279    1.845021

                                   SECTION  3      Y =  0.550000     CHORD =  0.400000
WING
     XB    0.0          0.200000    0.750000
     CP    1.443123     0.873079    1.914041

                                   SECTION  4      Y =  0.200000     CHORD =  0.400000
WING
     XB    0.0          0.200000    0.750000
     CP    1.176293     0.516708    0.088000
```

```
                                          *******************************************
                                          *  SPANWISE LOADING FOR COMPOSITE CASE  1  *
                                          *  WITH VECTORED THRUST AT ALPHA =    5.0  *
                                          *******************************************
                                                 FUNDAMENTAL CASE FACTORS
           A( 1)     A( 2)      A( 3)     A( 4)     A( 5)     A( 6)     A( 7)     A( 8)     A( 9)     A( 10)
        5.000000 30.000000  0.0        0.0       0.0       0.0       0.0       0.0       0.0       0.0

                    ........... LIFT ...........   *  *    ........... PITCHING MOMENT ...........   *  *   ... LIFT CENTER ...
SECTION      Y          CLB       CLMU      CL        *  *      CMG       CMMU      CMT       CM         *  *     XCP/C     XCL/C
   1      0.950000  0.555131  0.0       0.555131   *  *   -0.186938  0.0       0.0       -0.186938   *  *   0.336746  0.336746
   2      0.800000  1.273708  0.0       1.273708   *  *   -0.578918  0.0       0.0       -0.578918   *  *   0.454514  0.454514
   3      0.550000  1.381644  0.0       1.381644   *  *   -0.612809  0.0       0.0       -0.612809   *  *   0.443536  0.443536
   4      0.200000  0.464224  0.0       0.464224   *  *   -0.095330  0.0       0.0       -0.095330   *  *   0.205353  0.205353
         ---------- --------- --------- ---------- * *  *   --------- --------- --------- ---------   *  *  --------- ---------
           TOTAL    0.910437  0.707107  1.617543   *  *   -0.356452 -0.336801  0.176104 -0.517149  (APEX)   0.391517  0.428684
                                                   *  *   -0.128843 -0.160697  0.191511 -0.098029  (XMC )   0.156607  0.171433
```

```
SECTION       Y      .................................... INDUCED DRAG ...............
                     CDG        CDMU         CS          CD          GAMMA       ALFIN         CJ          CMU
   1      0.950000  0.0484443  0.0         0.0275537  0.0208906  0.1110262  0.1137443
   2      0.800000  0.2319084  0.0         0.0567882  0.1751202  0.2547416  0.3778918
   3      0.550000  0.2458448  0.0         0.0726965  0.1731483  0.2763288  0.3502491
   4      0.200000  0.0405112  0.0         0.0482990 -0.0077879  0.0928447  0.0270554
                    ---------- ----------  ---------- ----------  ---------- ---------- ---------- ----------
          TOTAL     0.1411840  0.2928932   0.0552415  0.3788356              0.1263894  0.6211644  1.0000000
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                 * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                               NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                         ***************************************************************
                         *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                         *            WITH VECTORED THRUST AT ALPHA =  5.0             *
                         ***************************************************************

                                              CCLG       0.9104372

                                              CCLJ       0.7071067

                                        **     CCL       1.6175432

                                              CCDG       0.1411840

                                              CCDJ       0.2928932

                                               CCS       0.0552415

                                               CCD       0.3788356

                                        ** CDII2         0.1263894

                                        **     CCJ       1.0000000

                                              CCMG      -0.3564521

                                              CCMJ      -0.3368006

                                              CCMT       0.1761037

                                               CCM      -0.5171490

                                              CXCP       0.3915175

                                              CXCL       0.4285837

                                             CXCPB       0.1566070

                                             CXCLB       0.1714334

                                            CCMGMC      -0.1288429

                                            CCMJMC      -0.1606968

                                            CCMTMC       0.1915107

                                        **  CCMMC       -0.0480290

                                              CLLG       0.0

                                              CLLJ       0.0

                                        *      CLL       0.0

                                               CNJ       0.0

                                        *    CNGMC       0.0

                                        *      CCY       0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                 * DOUGLAS AIRCRAFT COMPANY *                    * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                               NO GROUND EFFECT                                    NO STABILITY DERIVATIVES
                                          ******************************************
                                          *  CHORDWISE LOADING FOR COMPOSITE CASE  1  *
                                          *  WITH VECTORED THRUST AT ALPHA =    10.0  *
                                          ******************************************
                                                  FUNDAMENTAL CASE FACTORS
          A(1)         A(2)        A(3)        A(4)        A(5)        A(6)        A(7)        A(8)        A(9)        A(10)
      10.000000    30.000000    0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0

    *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                  DO NOT PLOT   USE LOADING POINTS DIRECTLY

                                  SECTION  1      Y =  0.950000     CHORD =  0.400000
WING
       XB     0.0          0.200000    0.750000
       CP     1.417365     0.741219    0.496288

                                  SECTION  2      Y =  0.800000     CHORD =  0.400000
WING
       XB     0.0          0.200000    0.750000
       CP     1.989733     1.104625    1.923962

                                  SECTION  3      Y =  0.550000     CHORD =  0.400000
WING
       XB     0.0          0.200000    0.750000
       CP     2.263344     1.302982    2.019238

                                  SECTION  4      Y =  0.200000     CHORD =  0.400000
WING
       XB     0.0          0.200000    0.750000
       CP     2.047189     0.981087    0.227888
```

```
                                          ******************************************
                                          *  SPANWISE LOADING FOR COMPOSITE CASE  1  *
                                          *  WITH VECTORED THRUST AT ALPHA =    10.0  *
                                          ******************************************
                                                  FUNDAMENTAL CASE FACTORS
        A(1)      A(2)      A(3)       A(4)      A(5)       A(6)       A(7)       A(8)       A(9)       A(10)
    10.000000 30.000000  0.0        0.0       0.0        0.0        0.0        0.0        0.0        0.0

                    ............ LIFT ............   * *   ........... PITCHING MOMENT ...........   * *    ... LIFT CENTER ...
SECTION     Y         CLG       CLMU       CL        * *     CMG       CMMU       CMT        CM        * *      XCP/C     XCL/C
   1     0.950000  0.759945  0.0        0.759945     * *   -0.229364  0.0        0.0       -0.229364   * *     0.301817  0.301817
   2     0.800000  1.581764  0.0        1.581764     * *   -0.648115  0.0        0.0       -0.648115   * *     0.409742  0.409742
   3     0.550000  1.748981  0.0        1.748981     * *   -0.698327  0.0        0.0       -0.698327   * *     0.399276  0.399276
   4     0.200000  0.868500  0.0        0.868500     * *   -0.194330  0.0        0.0       -0.194330   * *     0.223753  0.223753
        ---------- --------- --------- ---------     * *   --------- --------- --------- ---------    * *    --------- ---------
         TOTAL     1.264441  0.766044  2.030484      * *   -0.439789 -0.343948  0.158255 -0.625481  (APEX)    0.347813  0.385985
                                                     * *   -0.123679 -0.155346  0.186160 -0.092865  (XMC)     0.139125  0.154394


   SECTION       Y        .................. ................... INDUCED DRAG ...............
                             CDG         CDMU           CS          CD         GAMMA       ALFIN        CT           CMU
      1       0.950000   0.1326354    0.0          0.0701246   0.0625108   0.1519890   0.2398977
      2       0.800000   0.4019935    0.0          0.1381963   0.2637966   0.3163528   0.4435713
      3       0.550000   0.4374137    0.0          0.1788167   0.2585970   0.3497962   0.3875524
      4       0.200000   0.1515818    0.0          0.1462927   0.0052890   0.1747000   0.0547377
                         ----------  ----------   ----------  ----------  ----------  ----------  ----------  ----------
          TOTAL          0.2855189   0.3572124    0.1968138   0.4959175               0.1904591   0.5040825   1.0000000
```

```
* STOL AERODYNAMIC METHODS PROGRAM *             * DOUGLAS AIRCRAFT COMPANY *                     * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                         **************************************************************
                         *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                         *           WITH VECTORED THRUST AT ALPHA = 10.0              *
                         **************************************************************

                                             CCLG      1.2644405

                                             CCLJ      0.7660445

                                        **   CCL       2.0304842

                                             CCDG      0.2855189

                                             CCDJ      0.3572124

                                             CCS       0.1468138

                                             CCD       0.4959175

                                        ** CDITZ       0.1904591

                                        **   CCJ       1.0000000

                                             CCMG     -0.4397891

                                             CCMJ     -0.3439475

                                             CCMT      0.1582554

                                             CCM      -0.6254812

                                             CXCP      0.3478131

                                             CXCL      0.3859850

                                            CXCPB      0.1391252

                                            CXCLB      0.1543940

                                           CCMGMC     -0.1236791

                                           CCMJMC     -0.1553460

                                           CCMTMC      0.1861600

                                        ** CCMMC      -0.0928651

                                             CLLG      0.0

                                             CLLJ      0.0

                                        *    CLL       0.0

                                             CNJ       0.0

                                        *  CNGMC       0.0

                                        *    CCY       0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *             * DOUGLAS AIRCRAFT COMPANY *                     * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                                    NO STABILITY DERIVATIVES
                                        ********************************************
                                        *  CHORDWISE LOADING FOR COMPOSITE CASE  1  *
                                        *  WITH VECTORED THRUST AT ALPHA =    15.0  *
                                        ********************************************
                                                  FUNDAMENTAL CASE FACTORS
          A( 1 )        A( 2 )        A( 3 )       A( 4 )       A( 5 )       A( 6 )       A( 7 )       A( 8 )       A( 9 )       A( 10 )
       15.000000     30.000000     0.0          0.0          0.0          0.0          0.0          0.0          0.0          0.0

   *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                 DO NOT PLOT THESE LOADING POINTS DIRECTLY

                                  SECTION  1      Y =  0.950000     CHORD =  0.400000
WING
      XB    0.0           0.200000     0.750000
      CP    1.944681      0.956344     0.539421

                                  SECTION  2      Y =  0.800000     CHORD =  0.400000
WING
      XB    0.0           0.200000     0.750000
      CP    2.700939      1.458198     2.001329

                                  SECTION  3      Y =  0.550000     CHORD =  0.400000
WING ,
      XB    0.0           0.200000     0.750000
      CP    3.077871      1.727713     2.122483

                                  SECTION  4      Y =  0.200000     CHORD =  0.400000
WING
      XB    0.0           0.200000     0.750000
      CP    2.909163      1.440322     0.358246
```

```
                                  ********************************************
                                  *  SPANWISE LOADING FOR COMPOSITE CASE  1  *
                                  *  WITH VECTORED THRUST AT ALPHA =    15.0  *
                                  ********************************************
                                            FUNDAMENTAL CASE FACTORS
       A( 1 )    A( 2 )     A( 3 )      A( 4 )      A( 5 )      A( 6 )      A( 7 )      A( 8 )      A( 9 )      A( 10 )
    15.000000 30.000000  0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0

                      ............ LIFT .............    *  *    ........... PITCHING MOMENT ..........     *  *     ... LIFT CENTER ...
SECTION       Y          CLG       CLMU       CL         *  *       CMG       CMMU       CMT        CM         *  *      XCP/C      XCL/C
   1      0.950000  0.963533  0.0         0.963533       *  *    -0.271373  0.0        0.0       -0.271373     *  *    0.281644   0.281644
   2      0.800000  1.887543  0.0         1.887543       *  *    -0.716557  0.0        0.0       -0.716557     *  *    0.379624   0.379624
   3      0.550000  2.112459  0.0         2.112459       *  *    -0.782692  0.0        0.0       -0.782692     *  *    0.370512   0.370512
   4      0.200000  1.265250  0.0         1.265250       *  *    -0.290168  0.0        0.0       -0.290168     *  *    0.229336   0.229336
         --------- --------- ---------  ---------       *  *    --------- ---------  ---------  ---------     *  *   ---------  ---------
     TOTAL       1.613699  0.819152  2.432851            *  *    -0.521323 -0.342617  0.138508 -0.725432   (APEX)    0.323061   0.355114
                                                         *  *    -0.117899 -0.144807  0.175621 -0.087085   (XMC)     0.129224   0.142046


      SECTION       Y      ............................................ INDUCED DRAG ...............
                           CDG        CDMU           CS          CD         GAMMA        ALFIN          CT          CMU
         1      0.950000  0.2522522  0.0          0.1320091   0.1202431   0.1927066   0.3657689
         2      0.800000  0.6251444  0.0          0.2546459   0.3704984   0.3775085   0.5092149
         3      0.550000  0.6919567  0.0          0.3306801   0.3612766   0.4224918   0.4240642
         4      0.200000  0.3312419  0.0          0.2954224   0.0358195   0.2530500   0.0816485
                         ---------- ----------  ----------  ----------  ----------  ----------  ----------  ----------
              TOTAL      0.4903377  0.4264237   0.2815030   0.6352583               0.2690253   0.3647417   1.0000000
```

```
* STOL AERODYNAMIC METHODS PROGRAM *            * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME  1.26.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                            NO STABILITY DERIVATIVES

                   ************************************************************
                   *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                   *           WITH VECTORED THRUST AT ALPHA = 15.0               *
                   ************************************************************

                                      CCLG        1.6136990

                                      CCLJ        0.8191520

                                  **   CCL        2.4328508

                                      CCDG        0.4903377

                                      CCDJ        0.4264237

                                       CCS        0.2815030

                                       CCD        0.6352583

                                  ** CDITZ        0.2690253

                                  **   CCJ        1.0000000

                                      CCMG       -0.5213233

                                      CCMJ       -0.3426169

                                      CCMT        0.1385080

                                       CCM       -0.7254322

                                      CXCP        0.3230610

                                      CXCL        0.3551143

                                     CXCPB        0.1292244

                                     CXCLB        0.1420457

                                    CCMGMC       -0.1178488

                                    CCMJMC       -0.1448070

                                    CCMTMC        0.1756210

                                  ** CCMMC       -0.0870848

                                      CLLG        0.0

                                      CLLJ        0.0

                                  *    CLL        0.0

                                       CNJ        0.0

                                  *  CNGMC        0.0

                                  *    CCY        0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *            * DOUGLAS AIRCRAFT COMPANY *              * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                            NO STABILITY DERIVATIVES
                              **********************************************
                              *  CHORDWISE LOADING FOR COMPOSITE CASE  1  *
                              *  WITH VECTORED THRUST AT ALPHA =    20.0  *
                              **********************************************
                                          FUNDAMENTAL CASE FACTORS
          A(1)         A(2)        A(3)        A(4)        A(5)        A(6)        A(7)        A(8)        A(9)        A(10)
       20.000000    30.000000    0.0         0.0         0.0         0.0         0.0         0.0         0.0         0.0

     *** NOTE ***  EACH LEADING EDGE CP VALUE IS THE AVERAGE VALUE OF THE SINGULAR DISTRIBUTION
                   DO NOT PLOT THESE LOADING POINTS DIRECTLY

                              SECTION  1     Y =  0.950000     CHORD =  0.400000
WING
      XB    0.0          0.200000    0.750000
      CP    2.471186     1.171170    0.583811

                              SECTION  2     Y =  0.800000     CHORD =  0.400000
WING
      XB    0.0          0.200000    0.750000
      CP    3.410890     1.811330    2.079445

                              SECTION  3     Y =  0.550000     CHORD =  0.400000
WING
      XB    0.0          0.200000    0.750000
      CP    3.890359     2.152319    2.229266

                              SECTION  4     Y =  0.200000     CHORD =  0.400000
WING
      XB    0.0          0.200000    0.750000
      CP    3.768069     1.902119    0.492969
```

```
                                 **********************************************
                                 *  SPANWISE LOADING FOR COMPOSITE CASE  1   *
                                 *  WITH VECTORED THRUST AT ALPHA =    20.0  *
                                 **********************************************
                                           FUNDAMENTAL CASE FACTORS
        A(1)      A(2)      A(3)       A(4)       A(5)       A(6)       A(7)       A(8)       A(9)       A(10)
     20.000000 30.000000  0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0

                   ............ LIFT ............   *  *   ............ PITCHING MOMENT ............   *  *    ... LIFT CENTER ...
SECTION     Y        CLG       CLMU       CL        *  *      CMG       CMMU       CMT        CM        *  *      XCP/C     XCL/C
   1     0.950000 1.166949  0.0        1.166949     *  *   -0.313406  0.0        0.0       -0.313406    *  *    0.268569  0.268569
   2     0.800000 2.193203  0.0        2.193203     *  *   -0.785129  0.0        0.0       -0.785129    *  *    0.357983  0.357983
   3     0.550000 2.476896  0.0        2.476896     *  *   -0.867936  0.0        0.0       -0.867936    *  *    0.350413  0.350413
   4     0.200000 1.664095  0.0        1.664095     *  *   -0.387381  0.0        0.0       -0.387381    *  *    0.232788  0.232788
        --------- --------- --------- ---------     *  *   --------- --------- --------- ---------    *  *   --------- ---------
   TOTAL          1.964041  0.866025  2.830066      *  *   -0.603700 -0.332849  0.117461 -0.819087  (APEX)    0.307376  0.330928
                                                    *  *   -0.112690 -0.129400  0.160214 -0.081876  (XMC)     0.122950  0.132371


                     ..................................... INDUCED DRAG ...............
SECTION      Y        CDG        CDMU        CS         CD         GAMMA      ALFIN        CT          CMU
   1      0.950000  0.4073423  0.0        0.2131660  0.1941763  0.2333898  0.4913366
   2      0.800000  0.9016716  0.0        0.4061089  0.4955627  0.4386405  0.5745215
   3      0.550000  1.0105047  0.0        0.5283072  0.4821975  0.4953792  0.4622276
   4      0.200000  0.5808788  0.0        0.4956155  0.0852633  0.3328189  0.1089515
                    ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
        TOTAL       0.7565714  0.4999999  0.4592765  0.7972948             0.3676669  0.2027052  1.0000000
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                     * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                              NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                                 **************************************************************
                                 *  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                                 *            WITH VECTORED THRUST AT ALPHA = 20.0            *
                                 **************************************************************

                                                  CCLG        1.9640408

                                                  CCLJ        0.8660252

                                             **    CCL        2.8300657

                                                  CCDG        0.7565714

                                                  CCDJ        0.4999999

                                                   CCS        0.4592765

                                                   CCD        0.7972948

                                             ** CDITZ        0.3626669

                                             **    CCJ        1.0000000

                                                  CCMG       -0.6036996

                                                  CCMJ       -0.3328490

                                                  CCMT        0.1174614

                                                   CCM       -0.8190872

                                                  CXCP        0.3073763

                                                  CXCL        0.3309282

                                                 CXCPB        0.1229505

                                                 CXCLB        0.1323712

                                                CCMGMC       -0.1126897

                                                CCMJMC       -0.1294000

                                                CCMTMC        0.1602140

                                             ** CCMMC        -0.0818756

                                                  CLLG        0.0

                                                  CLLJ        0.0

                                             *     CLL        0.0

                                                   CNJ        0.0

                                             *   CNGMC        0.0

                                             *     CCY        0.0
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                     * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                              NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                                          **********************************************
                                          *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                          **********************************************

                                                  ALPHA =    0.0  DEGREES

             X           Y           Z          U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                     ANGLE       ANGLE
                                                                                  (DEGREES)   (DEGREES)

     1    1.112502    0.350000    0.300000    0.009949    0.003094   -0.062760    3.555867    0.175506
     2    1.262506    0.350000    0.300000    0.006784   -0.001321   -0.061102    3.473014   -0.075154
     3    1.112502    0.150000    0.300000    0.015265    0.005030   -0.050681    2.857785    0.283872
     4    1.262506    0.150000    0.300000    0.011147    0.002644   -0.052143    2.952035    0.149841
     5    1.112502   -0.150000    0.300000    0.015265   -0.005030   -0.050681    2.857786   -0.283872
     6    1.262506   -0.150000    0.300000    0.011147   -0.002644   -0.052143    2.952035   -0.149841
     7    1.112502   -0.350000    0.300000    0.009949   -0.003094   -0.062760    3.555867   -0.175506
     8    1.262506   -0.350000    0.300000    0.006784    0.001321   -0.061102    3.473014    0.075154
     9    1.112502    0.0         0.350000    0.015803    0.0        -0.047339    2.668217    0.0
    10    1.262506    0.0         0.350000    0.011750    0.0        -0.049018    2.773728    0.0
    11    1.112502    0.0         0.150000    0.019974    0.0        -0.042594    2.391253    0.0
    12    1.262506    0.0         0.150000    0.014789    0.0        -0.046535    2.625586    0.0
    13   -0.475000    0.0        -0.075000    0.006666    0.0         0.019285   -1.097478    0.0
    14   -0.375000    0.0        -0.075000    0.007476    0.0         0.024012   -1.365335    0.0
    15    0.350000    0.0        -0.075000    0.088049    0.0         0.053121   -2.795102    0.0
    16    1.150000    0.0        -0.075000    0.026663    0.0        -0.020968    1.170017    0.0
    17   -0.500000    0.125000    0.125000    0.013444   -0.000498    0.015902   -0.898958   -0.078129
    18    0.0         0.125000    0.125000    0.059585    0.002699    0.031627   -1.709682    0.144965
    19    0.500000    0.125000    0.125000    0.057246    0.024580   -0.010942    0.592987    1.331860
```

```
* STOL AERODYNAMIC METHODS PROGRAM *                * DOUGLAS AIRCRAFT COMPANY *                     * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                              NO GROUND EFFECT                                    NO STABILITY DERIVATIVES

                                          **********************************************
                                          *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                          **********************************************

                                                  ALPHA =    5.00 DEGREES

             X           Y           Z          U/UINF      V/UINF      W/UINF     DOWNWASH    SIDEWASH
                                                                                     ANGLE       ANGLE
                                                                                  (DEGREES)   (DEGREES)

     1    1.112502    0.350000    0.300000    0.005737   -0.032450   -0.106473    6.043139   -1.848028
     2    1.262506    0.350000    0.300000    0.001894   -0.040163   -0.104447    5.951567   -2.295570
     3    1.112502    0.150000    0.300000    0.014032   -0.006976   -0.106817    6.013322   -0.394140
     4    1.262506    0.150000    0.300000    0.008713   -0.010434   -0.109280    6.183063   -0.592656
     5    1.112502   -0.150000    0.300000    0.014032    0.006976   -0.106817    6.013322    0.394140
     6    1.262506   -0.150000    0.300000    0.008713    0.010434   -0.109280    6.183063    0.592656
     7    1.112502   -0.350000    0.300000    0.005737    0.032450   -0.106473    6.043142    1.848028
     8    1.262506   -0.350000    0.300000    0.001894    0.040163   -0.104447    5.951570    2.295570
     9    1.112502    0.0         0.350000    0.015333    0.0        -0.099112    5.575309    0.0
    10    1.262506    0.0         0.350000    0.010017    0.0        -0.101579    5.743017    0.0
    11    1.112502    0.0         0.150000    0.020765    0.0        -0.107405    6.006539    0.0
    12    1.262506    0.0         0.150000    0.014331    0.0        -0.112580    6.333292    0.0
    13   -0.475000    0.0        -0.075000    0.007052    0.0         0.034168   -1.943254    0.0
    14   -0.375000    0.0        -0.075000    0.007182    0.0         0.043999   -2.501384    0.0
    15    0.350000    0.0        -0.075000    0.088570    0.0        -0.014937    0.786133    0.0
    16    1.150000    0.0        -0.075000    0.032065    0.0        -0.070463    3.905734    0.0
    17   -0.500000    0.125000    0.125000    0.020958   -0.000747    0.027563   -1.546426   -0.041942
    18    0.0         0.125000    0.125000    0.149429    0.001054    0.039007   -1.943660    0.052553
    19    0.500000    0.125000    0.125000    0.073141    0.023625   -0.084989    4.578155    1.261148
```

```
* STOL AERODYNAMIC METHODS PROGRAM *            * DOUGLAS AIRCRAFT COMPANY *             * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                                      ***********************************************
                                      *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                      ***********************************************

                                                ALPHA =   10.00 DEGREES

            X           Y           Z         U/UINF       V/UINF       W/UINF      DOWNWASH     SIDEWASH
                                                                                     ANGLE        ANGLE
                                                                                   (DEGREES)    (DEGREES)

  1     1.112502    0.350000    0.300000   -0.004031    -0.078200    -0.154622     8.824579    -4.489462
  2     1.262506    0.350000    0.300000   -0.008431    -0.093197    -0.154412     8.851293    -5.369430
  3     1.112502    0.150000    0.300000    0.006849    -0.018759    -0.168059     9.476224    -1.067401
  4     1.262506    0.150000    0.300000    0.000451    -0.023175    -0.173600     9.844068    -1.326990
  5     1.112502   -0.150000    0.300000    0.006849     0.018759    -0.168059     9.476228     1.067400
  6     1.262506   -0.150000    0.300000    0.000451     0.023175    -0.173600     9.844070     1.326989
  7     1.112502   -0.350000    0.300000   -0.004031     0.078200    -0.154622     8.824579     4.489462
  8     1.262506   -0.350000    0.300000   -0.008431     0.093197    -0.154412     8.851295     5.369430
  9     1.112502    0.0         0.350000    0.009404     0.0          -0.154262     8.689007     0.0
 10     1.262506    0.0         0.350000    0.003021     0.0          -0.158876     9.000747     0.0
 11     1.112502    0.0         0.150000    0.016212     0.0          -0.170457     9.522041     0.0
 12     1.262506    0.0         0.150000    0.008991     0.0          -0.177035     9.951674     0.0
 13    -0.475000    0.0        -0.075000    0.009906     0.0           0.048811    -2.767071     0.0
 14    -0.375000    0.0        -0.075000    0.010218     0.0           0.063835    -3.615669     0.0
 15     0.350000    0.0        -0.075000    0.075391     0.0          -0.079738     4.240598     0.0
 16     1.150000    0.0        -0.075000    0.035419     0.0          -0.104644     5.770990     0.0
 17    -0.500000    0.125000    0.125000    0.030364    -0.001020     0.037844    -2.102877    -0.056722
 18     0.0         0.125000    0.125000    0.240113    -0.000618     0.030964    -1.430316    -0.028538
 19     0.500000    0.125000    0.125000    0 077225     0.073593    -0.159490     8.421847     1.254681
```

```
* STOL AERODYNAMIC METHODS PROGRAM *            * DOUGLAS AIRCRAFT COMPANY *             * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                                      ***********************************************
                                      *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                      ***********************************************

                                                ALPHA =   15.00 DEGREES

            X           Y           Z         U/UINF       V/UINF       W/UINF      DOWNWASH     SIDEWASH
                                                                                     ANGLE        ANGLE
                                                                                   (DEGREES)    (DEGREES)

  1     1.112502    0.350000    0.300000   -0.019699    -0.140625    -0.215336    12.389003    -8.163421
  2     1.262506    0.350000    0.300000   -0.025593    -0.170151    -0.222176    12.844518    -9.905129
  3     1.112502    0.150000    0.300000   -0.006310    -0.029700    -0.240273    13.593162    -1.711955
  4     1.262506    0.150000    0.300000   -0.014319    -0.033872    -0.251185    14.296637    -1.968154
  5     1.112502   -0.150000    0.300000   -0.006310     0.029699    -0.240273    13.593162     1.711951
  6     1.262506   -0.150000    0.300000   -0.014319     0.033872    -0.251185    14.296637     1.968150
  7     1.112502   -0.350000    0.300000   -0.019699     0.140625    -0.215336    12.389006     8.163417
  8     1.262506   -0.350000    0.300000   -0.025593     0.170151    -0.222176    12.844518     9.905129
  9     1.112502    0.0         0.350000   -0.001678     0.0          -0.217732    12.303403     0.0
 10     1.262506    0.0         0.350000   -0.009436     0.0          -0.225767    12.839419     0.0
 11     1.112502    0.0         0.150000    0.007190     0.0          -0.236342    13.205825     0.0
 12     1.262506    0.0         0.150000   -0.000565     0.0          -0.242863    13.658181     0.0
 13    -0.475000    0.0        -0.075000    0.016469     0.0           0.063120    -3.553328     0.0
 14    -0.375000    0.0        -0.075000    0.018050     0.0           0.083335    -4.679672     0.0
 15     0.350000    0.0        -0.075000    0.060745     0.0          -0.139491     7.491576     0.0
 16     1.150000    0.0        -0.075000    0.037759     0.0          -0.132053     7.251802     0.0
 17    -0.500000    0.125000    0.125000    0.042715    -0.001373     0.046446    -2.550451    -0.075464
 18     0.0         0.125000    0.125000    0.331686    -0.002493     0.007030    -0.302451    -0.107260
 19     0.500000    0.125000    0.125000    0.074994     0.024052    -0.238087    12.488115     1.281740
```

```
* STOL AERODYNAMIC METHODS PROGRAM *            * DOUGLAS AIRCRAFT COMPANY *             * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                          NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                                      ***********************************************
                                      *  FLOW FIELD SOLUTION FOR COMPOSITE CASE  1  *
                                      ***********************************************

                                                ALPHA =   20.00 DEGREES

            X           Y           Z         U/UINF       V/UINF       W/UINF      DOWNWASH     SIDEWASH
                                                                                     ANGLE        ANGLE
                                                                                   (DEGREES)    (DEGREES)

  1     1.112502    0.350000    0.300000   -0.044630    -0.226343    -0.291993    16.994888   -13.328553
  2     1.262506    0.350000    0.300000   -0.054897    -0.284289    -0.320060    18.708710   -16.741394
  3     1.112502    0.150000    0.300000   -0.026964    -0.036538    -0.319350    18.169815    -2.150478
  4     1.262506    0.150000    0.300000   -0.036942    -0.037341    -0.337697    19.323242    -2.220457
  5     1.112502   -0.150000    0.300000   -0.026964     0.036538    -0.319350    18.169815     2.150476
  6     1.262506   -0.150000    0.300000   -0.036942     0.037341    -0.337697    19.323242     2.220453
  7     1.112502   -0.350000    0.300000   -0.044630     0.226343    -0.291993    16.994888    13.328553
  8     1.262506   -0.350000    0.300000   -0.054897     0.284289    -0.320060    18.708710    16.741394
  9     1.112502    0.0         0.350000   -0.019043     0.0          -0.284936    16.196873     0.0
 10     1.262506    0.0         0.350000   -0.028188     0.0          -0.297337    17.012100     0.0
 11     1.112502    0.0         0.150000   -0.005673     0.0          -0.295855    16.570007     0.0
 12     1.262506    0.0         0.150000   -0.012715     0.0          -0.299384    16.869354     0.0
 13    -0.475000    0.0        -0.075000    0.025587     0.0           0.076436    -4.262324     0.0
 14    -0.375000    0.0        -0.075000    0.029325     0.0           0.101690    -5.642132     0.0
 15     0.350000    0.0        -0.075000    0.034552     0.0          -0.194119    10.627190     0.0
 16     1.150000    0.0        -0.075000    0.042465     0.0          -0.136777     7.474805     0.0
 17    -0.500000    0.125000    0.125000    0.056758    -0.001764     0.053064    -2.874630    -0.095628
 18     0.0         0.125000    0.125000    0.420881    -0.004436    -0.032521     1.311152    -0.178878
 19     0.500000    0.125000    0.125000    0.061932     0.025484    -0.315867    16.564865     1.274729
```

```
* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *             * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                        NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                                       ****************************************
                                       *       INTERFERENCE TAIL ANALYSIS      *
                                       *  CHORDWISE LOADING FOR ALL TAIL CASES *
                                       ****************************************

                                            CASE 1 IS FOR ALPHA =' 0.0
                                            CASE 2 IS FOR ALPHA =  5.00
                                            CASE 3 IS FOR ALPHA = 10.00
                                            CASE 4 IS FOR ALPHA = 15.00
                                            CASE 5 IS FOR ALPHA = 20.00


                                                 ******************
                                                 * HORIZONTAL TAIL *
                                                 ******************

                     SECTION  1      Y =    0.350000     CHORD =    0.300007
           ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
             '1        0.0           0.084827     0.239469     0.369684     0.444501     0.459989
              2        0.500000      0.479004     0.512583     0.534750     0.532932     0.493080

                     SECTION  2      Y =    0.150000     CHORD =    0.300007
           ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
              3        0.0           0.246364     0.437203     0.591687     0.672228     0.695549
              4        0.500000      0.650427     0.702759     0.737990     0.744054     0.726090

                     SECTION  3      Y =   -0.150000     CHORD =    0.300007
           ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
              5        0.0          -0.072323     0.118517     0.272999     0.353540     0.376861
              6        0.500000      0.447863     0.500194     0.535426     0.541490     0.523529

                     SECTION  4      Y =   -0.350000     CHORD =    0.300007
           ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
              7        0.0          -0.084501     0.070140     0.200356     0.275173     0.290662
              8        0.500000      0.386617     0.420196     0.442362     0.440544     0.400693


                                                  *****************
                                                  * VERTICAL TAIL *
                                                  *****************

                     SECTION  1      Z =    0.350000     CHORD =    0.300007
           ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
              1        0.0           0.124678     0.124678     0.124678     0.124678     0.124678
              2        0.500000      0.414898     0.414898     0.414898     0.414898     0.414898

                     SECTION  2      Z =    0.150000     CHORD =    0.300007
           ELEMENT     X/C           CASE 1       CASE 2       CASE 3       CASE 4       CASE 5
              3        0.0           0.487885     0.487885     0.487885     0.487885     0.487885
              4        0.500000      0.696890     0.696890     0.696890     0.696890     0.696890


* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *             * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                        NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                                   ****************************************************
                                   *             INTERFERENCE TAIL ANALYSIS            *
                                   *  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES    *
                                   ****************************************************


                                                 ******************
                                                 * HORIZONTAL TAIL *
                                                 ******************

          CASE 1                     CASE 2                     CASE 3                     CASE 4                     CASE 5
K   CLH     CMH     CDIH       CLH     CMH     CDIH       CLH     CMH     CDIH       CLH     CMH     CDIH       CLH     CMH     CDIH
1   0.2819 -0.1550  0.0079     0.3760 -0.1751  0.0271     0.4522 -0.1902  0.0488     0.4887 -0.1943  0.0623     0.4765 -0.1828  0.0563
2   0.4484 -0.2187  0.0110     0.5700 -0.2469  0.0269     0.6648 -0.2676  0.0424     0.7081 -0.2745  0.0504     0.7108 -0.2704  0.0518
3   0.1870 -0.1354  0.0186     0.3094 -0.1637  0.0318     0.4042 -0.1844  0.0451     0.4475 -0.1913  0.0518     0.4502 -0.1872  0.0523
4   0.1511 -0.1155  0.0125     0.2452 -0.1357  0.0281     0.3214 -0.1508  0.0466     0.3579 -0.1549  0.0579     0.3457 -0.1434  0.0521
-----------------------------------------------------------------------------------------------------------------------------------
    0.0878 -0.2351  0.0041     0.1222 -0.3094  0.0087     0.1493 -0.3591  0.0134     0.1618 -0.3693  0.0160     0.1615 -0.3477  0.0158

      CLLH =  -0.006116          CLLH =  -0.006116          CLLH =  -0.006116          CLLH =  -0.006116          CLLH =  -0.006116


                                                  *****************
                                                  * VERTICAL TAIL *
                                                  *****************

          CASE 1                     CASE 2                     CASE 3                     CASE 4                     CASE 5
K   CYV     CNV     CDIV       CYV     CNV     CDIV       CYV     CNV     CDIV       CYV     CNV     CDIV       CYV     CNV     CDIV
1   0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338     0.2698 -0.1374  0.0338
2   0.5924 -0.2483  0.0402     0.5924 -0.2483  0.0402     0.5924 -0.2483  0.0402     0.5924 -0.2483  0.0402     0.5924 -0.2483  0.0402
-----------------------------------------------------------------------------------------------------------------------------------
    0.0384 -0.0988  0.0029     0.0384 -0.0988  0.0029     0.0364 -0.0988  0.0029     0.0384 -0.0988  0.0029     0.0384 -0.0988  0.0029

      CLLV =    0.006769         CLLV =    0.006769         CLLV =    0.006769         CLLV =    0.006769         CLLV =    0.006769

          *** NOTE *** ALL TAIL COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF



* STOL AERODYNAMIC METHODS PROGRAM *          * DOUGLAS AIRCRAFT COMPANY *             * DATE 11/21/73  *  TIME  1.28.14
VECTORED JET CASE NUMBER  1                        NO GROUND EFFECT                          NO STABILITY DERIVATIVES

                         ****************************************************************
                         * SECTIONAL AND TOTAL FUSELAGE LOADING FROM PRESSURE INTEGRATION *
                         *                    FOR COMPOSITE CASE   1                     *
                         ****************************************************************

 SEG      X         CM( 0 )    CN( 0 )     CM( 5 )    CN( 5 )     CM( 10 )   CN( 10 )    CM( 15 )   CN( 15 )    CM( 20 )   CN( 20 )

   1  -0.475000   -0.001686    0.0         0.007475   0.0         0.015408   0.0         0.021866   0.0         0.026635   0.0
   2  -0.375000    0.001576    0.0         0.015150   0.0         0.028019   0.0         0.039753   0.0         0.049921   0.0
   3   0.350000    0.000000    0.0         0.000000   0.0         0.000000   0.0        -0.000000   0.0        -0.000006   0.0
   4   1.150000   -0.007180    0.0         0.011112   0.0         0.035677   0.0         0.061652   0.0         0.094927   0.0
-------------------------------------------------------------------------------------------------------------------------------
TOTAL             -0.007290    0.0         0.033737   0.0         0.079104   0.0         0.123277   0.0         0.171483   0.0

        *** NOTE *** ALL FUSELAGE COEFFICIENTS AND STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA AND CREF
```

```
* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
NO VECTORED JET OR JET SHEET                          NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                 *******************************************************************
                 *  SUMMARY OF AIRCRAFT AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                 *******************************************************************

                                        *** WING ***
         ALPHA      CCL          CMMC         CLL           CDITZ         CN           CCY
          0.0     0.681045    -0.150261     0.0           0.079866      0.0          0.0
          5.0     1.040803    -0.144717     0.0           0.133189      0.0          0.0
         10.0     1.400561    -0.139173     0.0           0.202587      0.0          0.0
         15.0     1.760320    -0.133630     0.0           0.288060      0.0          0.0
         20.0     2.120078    -0.128086     0.0           0.389608      0.0          0.0

                                   *** HORIZONTAL TAIL ***
         ALPHA      CCL          CMMC         CLL           CCDI          CN           CCY
          0.0     0.103239    -0.272507    -0.006116      0.005936
          5.0     0.143087    -0.358419    -0.006116      0.012139
         10.0     0.173927    -0.414773    -0.006116      0.018407
         15.0     0.193909    -0.438536    -0.006116      0.023105
         20.0     0.200886    -0.427891    -0.006116      0.024522

                                    *** VERTICAL TAIL ***
         ALPHA      CCL          CMMC         CLL           CCDI          CN           CCY
          0.0                                0.006769      0.002893    -0.098765     0.038381
          5.0                                0.006769      0.002893    -0.098765     0.038381
         10.0                                0.006769      0.002893    -0.098765     0.038381
         15.0                                0.006769      0.002893    -0.098765     0.038381
         20.0                                0.006769      0.002893    -0.098765     0.038381

                                      *** FUSELAGE ***
         ALPHA      CCL          CMMC         CLL           CCDI          CN           CCY
          0.0                   0.012106                                 0.0
          5.0                   0.060398                                 0.0
         10.0                   0.110168                                 0.0
         15.0                   0.159199                                 0.0
         20.0                   0.204832                                 0.0


                                     *********************
                                     * COMPLETE AIRCRAFT *
                                     *********************
             ALPHA      CCL          CMMC         CLL          CDITZ         CN           CCY
              0.0     0.784284    -0.410662     0.000653     0.088695    -0.098765     0.038381
              5.0     1.183889    -0.442738     0.000653     0.148222    -0.098765     0.038381
             10.0     1.574489    -0.443778     0.000653     0.223887    -0.098765     0.038381
             15.0     1.954228    -0.412966     0.000653     0.314058    -0.098765     0.038381
             20.0     2.320964    -0.351145     0.000653     0.417023    -0.098765     0.038381
                          1


* STOL AERODYNAMIC METHODS PROGRAM *              * DOUGLAS AIRCRAFT COMPANY *               * DATE 11/21/73  *  TIME  1.28.13
VECTORED JET CASE NUMBER  1                           NO GROUND EFFECT                               NO STABILITY DERIVATIVES

                 *******************************************************************
                 *  SUMMARY OF AIRCRAFT AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE  1  *
                 *******************************************************************

                                        *** WING ***
         ALPHA      CCL          CMMC         CLL           CDITZ         CN           CCY
          0.0     1.199197    -0.103273     0.0           0.077459      0.0          0.0
          5.0     1.617543    -0.098029     0.0           0.126389      0.0          0.0
         10.0     2.030484    -0.092865     0.0           0.190459      0.0          0.0
         15.0     2.432851    -0.087085     0.0           0.269025      0.0          0.0
         20.0     2.830066    -0.081876     0.0           0.362667      0.0          0.0

                                   *** HORIZONTAL TAIL ***
         ALPHA      CCL          CMMC         CLL           CCDI          CN           CCY
          0.0     0.087807    -0.235053    -0.006116      0.004092
          5.0     0.122223    -0.309359    -0.006116      0.008675
         10.0     0.149281    -0.359066    -0.006116      0.013423
         15.0     0.161762    -0.369291    -0.006116      0.016000
         20.0     0.161451    -0.347723    -0.006116      0.015771

                                    *** VERTICAL TAIL ***
         ALPHA      CCL          CMMC         CLL           CCDI          CN           CCY
          0.0                                0.006769      0.002893    -0.098765     0.038381
          5.0                                0.006769      0.002893    -0.098765     0.038381
         10.0                                0.006769      0.002893    -0.098765     0.038381
         15.0                                0.006769      0.002893    -0.098765     0.038381
         20.0                                0.006769      0.002893    -0.098765     0.038381

                                      *** FUSELAGE ***
         ALPHA      CCL          CMMC         CLL           CCDI          CN           CCY
          0.0                  -0.007290                                 0.0
          5.0                   0.033737                                 0.0
         10.0                   0.079104                                 0.0
         15.0                   0.123272                                 0.0
         20.0                   0.171483                                 0.0


                                     *********************
                                     * COMPLETE AIRCRAFT *
                                     *********************
             ALPHA      CCL          CMMC         CLL          CDITZ         CN           CCY
              0.0     1.287004    -0.345616     0.000653     0.084444    -0.098765     0.038381
              5.0     1.739766    -0.373651     0.000653     0.137956    -0.098765     0.038381
             10.0     2.179765    -0.372827     0.000653     0.206775    -0.098765     0.038381
             15.0     2.594612    -0.333104     0.000653     0.287919    -0.098765     0.038381
             20.0     2.991516    -0.258116     0.000653     0.381331    -0.098765     0.038381


                        ****************************************************
                        *  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  *
                        ****************************************************


                        ****************************************************
                        *  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  *
                        ****************************************************
```

APPENDIX V

PROGRAM LISTING

```
*DECK STAMP                                                             MAIN0010
      OVERLAY(OVLY,0,0)                                                 MAIN0020
      PROGRAM STAMP(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,        MAIN0030
     1        TAPE7=PUNCH,TAPE1=400,TAPE2=750,TAPE3=750,                MAIN0040
     1        TAPE4=750,TAPE8=600,TAPE9=400,TAPE10=750,TAPE11=750,      MAIN0050
     2        TAPE12=750,TAPE13=750)                                    MAIN0060
C                                                                       MAIN0070
C***********************************************************************MAIN0080
C***** DOUGLAS STOL AERODYNAMICS METHODS PROGRAM (STAMP) *****          MAIN0090
C***********************************************************************MAIN0100
C                                                                       MAIN0110
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IA,ISTAB                MAIN0120
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            MAIN0130
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) MAIN0140
      COMMON/LUKE/TITLE(8)                                              MAIN0150
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     MAIN0160
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI,IGRMAIN0170
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  MAIN0180
     1            D(40),KK(400),ITYPE(400)                              MAIN0190
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             MAIN0200
      COMMON/FCASE2/IST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     MAIN0210
     1            XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTMAIN0220
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    MAIN0230
      COMMON/CASES/INT(10),INH(10),IND(10),INC(10),INB(10),IHLE(10),    MAIN0240
     1      IHTE(10)                                                    MAIN0250
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           MAIN0260
      COMMON/COMPOS/FACTOR(10,24),NCC                                   MAIN0270
      COMMON/DERIV/UD(40),CLQ,CMQ,CMQMC                                 MAIN0280
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          MAIN0290
      COMMON/SOLV1/GAMMA(400,10)                                        MAIN0300
      COMMON/SOLV3/CPREAD(400)                                          MAIN0310
      COMMON/LOAD6/ COEFF(27)                                           MAIN0320
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         MAIN0330
     1            IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),                MAIN0340
     2            XTR(40),XTL(40),CHR(40),CHL(40)                       MAIN0350
      COMMON /FFLD5/ GAMJ(400,5)                                        MAIN0360
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HX,HH,VL,VH,     MAIN0370
     1            NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,     MAIN0380
     2            ICAMV,NKTYPE,NCTYPE,ISENSE,KCASE,ICASES               MAIN0390
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),   MAIN0400
     1    IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NCA(5),XBT(10,5),       MAIN0410
     2            CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),         MAIN0420
     3            CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),          MAIN0430
     4    SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)MAIN0440
      COMMON/TAIL3/XH(80),XV(20),XCH(80),XCV(20),XLH(20),XTH(20),       MAIN0450
     1            XLV(10),XTV(10),YACH,ZACV                             MAIN0460
      COMMON /TAIL4/ B(100,5)                                           MAIN0470
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XF(41),ZCL(41),    MAIN0480
     1             RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                MAIN0490
     2             V(40,7),W(40,7)                                      MAIN0500
      COMMON/JET12/UJ(10),DD0(10),THTA(10),H (10),ZMX(10),X0(10),Y0(10),MAIN0510
     1          Z0(10),DIHE(10),IS (10),JETRE(10),NJETS                 MAIN0520
      COMMON/JET13/CJ(10),TH0(10)                                       MAIN0530
      COMMON/DATTIM/MDY,HMS                                             MAIN0540
      COMMON/SLIPIT/SSLIP                                               MAIN0550
      COMMON/KEEPEM/PUT(102)                                            MAIN0560
      DIMENSION INDEX1(1301),INDEX2(1516),INDEX3(889)                   MAIN0570
      DIMENSION PUT1(38)                                                MAIN0580
      LOGICAL VT,STAB,GE,FLOFLD,TAIL,FSLG,NOTEVD,NOWING,SUMARY          MAIN0590
      EQUIVALENCE (PUT(1),PUT1(1))                                      MAIN0600
      DATA CHECK/1OHY                                                   MAIN0610
C                                                                       MAIN0620
      CALL OPENMS(1,INDEX1,1301,0)                                      MAIN0630
      CALL OPENMS(8,INDEX2,1516,0)                                      MAIN0640
      CALL OPENMS(9,INDEX3,889,0)                                       MAIN0650
C  READ THE TITLE FOR THIS CASE                                         MAIN0660
   20 READ(5, 30 ) TITLE                                                MAIN0670
   30 FORMAT(8A10)                                                      MAIN0680
      IF(EOF(5)) 480 , 40                                               MAIN0690
C  READ GENERAL GEOMETRY CONTROL DATA                                   MAIN0700
   40 READ(5, 50 )AREA,SPAN,CREF,XMC,XCG,TCR,SSLIP                      MAIN0710
   50 FORMAT(7F10.6)                                                    MAIN0720
      READ(5, 60 )NROWS,NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE,IDERIV,MAIN0730
     1           IGRND,IFFLD,NPOINT,NROWH,ICAMH,NROWV,ICAMV,NSEG,ISECT, MAIN0740
     2            ICAMF                                                 MAIN0750
   60 FORMAT(10I2,I4,7I2)                                               MAIN0760
      ARE = AREA                                                        MAIN0770
      SPA = SPAN                                                        MAIN0780
      CRE = CREF                                                        MAIN0790
      XM  = XMC                                                         MAIN0800
      XC  = XCG                                                         MAIN0810
      NRO = NROWS                                                       MAIN0820
      NC  = NCASES                                                      MAIN0830
      ISY = ISYMM                                                       MAIN0840
      IPR = IPRINT                                                      MAIN0850
      JET = JETFLG                                                      MAIN0860
      IGT = IGTYPE                                                      MAIN0870
      IHI = IHINGE                                                      MAIN0880
      IGR = IGRND                                                       MAIN0890
C  DEFINE LOGICALS                                                      MAIN0900
      VT = JETFLG .EQ. 2                                                MAIN0910
      STAB = IDERIV .NE. 0                                              MAIN0920
      GE = IGRND .NE. 0                                                 MAIN0930
      TAIL = (NROWH+NROWV) .GT. 0                                       MAIN0940
      FSLG = NSEG .GT. 0                                                MAIN0950
      FLOFLD = (NROWH+NROWV+NSEG+NPOINT) .GT. 0                         MAIN0960
      NOTEVD = IFFLD .LT. 0                                             MAIN0970
      NOWING = NROWS .EQ. 0                                             MAIN0980
      SUMARY = (NROWS .GT. 0) .AND. (TAIL .OR. FSLG)                    MAIN0990
C  DEFINE FLAGS                                                         MAIN1000
      IA = 1                                                            MAIN1010
      ISYM = ISYMM                                                      MAIN1020
      NEWCMU = 0                                                        MAIN1030
      NOALFA = 1                                                        MAIN1040
      IF(ISYMM .LT. 0) NOALFA = 0                                       MAIN1050
C  CHECK CONSISTENCY OF SOME BASIC CONTROL FLAGS                        MAIN1060
      IF(NOWING) GO TO 70                                               MAIN1070
      IF((NROWS.LT.3) .OR. (NROWS.GT.40)) GO TO 430                     MAIN1080
      IF((NCASES .LT. 1) .OR. (NCASES .GT. 10)) GO TO 430               MAIN1090
      IF((IGTYPE .LT. 1) .OR. (IGTYPE .GT. 2)) GO TO 430                MAIN1100
      IF((JETFLG .LT. 0) .OR. (JETFLG .GT. 2)) GO TO 430                MAIN1110
      IF(STAB .AND. (ISYMM .LT. 0)) GO TO 430                           MAIN1120
      IF((GE .OR. STAB) .AND. (IHINGE .NE. 0)) GO TO 430                MAIN1130
      GO TO 80                                                          MAIN1140
   70 NMAX = 0                                                          MAIN1150
      NJT = 0                                                           MAIN1160
      NWT = 0                                                           MAIN1170
      CMAC = CREF * 2.0/SPAN                                            MAIN1180
   80 IF(NPOINT .GT. 200) GO TO 430                                     MAIN1190
      IF((NROWH .GT. 20) .OR. (NROWV .GT. 10) .OR. (NSEG .GT. 40))      MAIN1200
     1    GO TO 430                                                     MAIN1210
      CALL DATE(MDY)                                                    MAIN1220
      CALL TIME(HMS)                                                    MAIN1230
      DO 90 I = 1,102                                                   MAIN1240
   90 PUT(I) = 0.0                                                      MAIN1250
C                                                                       MAIN1260
C  CYCLE THE CMU CASES                                                  MAIN1270
  100 NEWCMU = NEWCMU + 1                                               MAIN1280
      ISTAB = 0                                                         MAIN1290
      ISYMM = ISYM                                                      MAIN1300
      IF((NEWCMU .GT. 1) .AND. (JETFLG .EQ. 1))GO TO 450                MAIN1310
      NHITE = 1                                                         MAIN1320
C  READ ALL THE INPUT DATA AND PRINT IF REQUIRED AT THIS TIME           MAIN1330
      LOGIC=1                                                           MAIN1340
      IF(STAB)LOGIC=2                                                   MAIN1350
      IF(NOWING) LOGIC = 4                                              MAIN1360
      CALL OVERLAY(4HOVLY,1,0)                                          MAIN1370
      GO TO ( 110 , 450 , 480 , 490 ),IA                                MAIN1380
  110 IF(NOWING) GO TO 230                                              MAIN1390
      IF((NCC .EQ. 0) .AND. (VT .OR. FLOFLD)) GO TO 430                 MAIN1400
C                                                                       MAIN1410
C  CYCLE THE GROUND HEIGHTS                                             MAIN1420
      IGND = 0                                                          MAIN1430
      IF((NEWCMU .EQ. 1) .OR. STAB) GO TO 140                           MAIN1440
      IF(JETFLG .EQ. 0) WRITE(6, 120 ) NEWCMU                           MAIN1450
  120 FORMAT(1H1 ///////////// 49X,5(4H****),3H*** /                    MAIN1460
     1        49X,17H*  JET SHEET CASE, I3, 3H  * /                     MAIN1470
     2        49X, 5(4H****), 3H***)                                    MAIN1480
      NCM1 = NEWCMU - 1                                                 MAIN1490
      IF(VT) WRITE(6, 130 ) NCM1                                        MAIN1500
  130 FORMAT(1H1 ///////////// 43X,7(5H*****) /                         MAIN1510
     1        43X, 29H*  POWER ON VECTORED JET CASE, I3, 3H  * /        MAIN1520
     2        43X, 7(5H*****))                                          MAIN1530
  140 IF(.NOT. GE)GO TO 180                                             MAIN1540
      NHITE = 0                                                         MAIN1550
  150 ISTAB = 0                                                         MAIN1560
      ISYMM = ISYM                                                      MAIN1570
      NHITE = NHITE + 1                                                 MAIN1580
      IF((NHITE .EQ. 1) .AND. (.NOT. STAB)) WRITE(6, 160 )              MAIN1590
  160 FORMAT(1H1 ///////////// 45X,7(4H****), 2H** /                    MAIN1600
     1        45X, 30H*  GROUND HEIGHT = INFINITY  * /                  MAIN1610
     2        45X, 7(4H****), 2H**)                                     MAIN1620
      IF(NHITE .EQ. 1) GO TO 180                                        MAIN1630
      IF(GHITE(NHITE) .LT. 0.001) GO TO 100                             MAIN1640
      IF(.NOT. STAB) WRITE(6, 170 ) GHITE(NHITE)                        MAIN1650
  170 FORMAT(1H1 ///////////// 45X, 7(4H****), 3H*** /                  MAIN1660
     1        45X, 18H*  GROUND HEIGHT =, F10.6, 3H  * /                MAIN1670
     2        45X, 7(4H****), 3H***)                                    MAIN1680
      IGND = 1                                                          MAIN1690
      IF((NEWCMU .EQ. 2) .AND. VT) NEWCMU = 1                           MAIN1700
C  ADD THE EXTRA FUNDAMENTAL CASE FOR PITCHING OR ROLLING RATE DERIV.   MAIN1710
  180 IF(.NOT. STAB)GO TO 200                                           MAIN1720
      IF((NEWCMU .EQ. 1) .AND. (NHITE .EQ. 1))NCASES = NCASES + 1       MAIN1730
      IF(NCASES .GT. 10)GO TO 430                                       MAIN1740
      LOGIC = 1                                                         MAIN1750
      CALL OVERLAY(4HOVLY,4,0)                                          MAIN1760
C  PRINT GEOMETRY DATA FOR ALL FUNDAMENTAL CASES                        MAIN1770
      LOGIC=3                                                           MAIN1780
      IF((NEWCMU .EQ. 1) .AND. (NHITE .EQ. 1)) CALL OVERLAY(4HOVLY,1,0) MAIN1790
      LOGIC = -1                                                        MAIN1800
      IF(NEWCMU .EQ. 1) GO TO 190                                       MAIN1810
      IF(JETFLG .EQ. 0) WRITE(6, 120 ) NEWCMU                           MAIN1820
      NCM1 = NEWCMU - 1                                                 MAIN1830
      IF(VT) WRITE(6, 130 ) NCM1                                        MAIN1840
  190 IF(NHITE .EQ. 1) WRITE(6, 160 )                                   MAIN1850
      IF(NHITE .GT. 1) WRITE(6, 170 ) GHITE(NHITE)                      MAIN1860
      GO TO 210                                                         MAIN1870
C                                                                       MAIN1880
C  SET UP THE DOWNWASH MATRIX.  IF LOGIC = -1, A STABILITY MATRIX WILL  MAIN1890
C  ALSO BE FORMED.                                                      MAIN1900
  200 LOGIC = 1                                                         MAIN1910
  210 IF(NOTEVD) GO TO 230                                              MAIN1920
      IF(NHITE .GT. 1) GO TO 220                                        MAIN1930
      IF(VT .AND. (NEWCMU .GT. 1)) GO TO 310                            MAIN1940
  220 CALL OVERLAY(4HOVLY,2,0)                                          MAIN1950
C  SOLVE THE JET-WING MATRIX                                            MAIN1960
      LOGIC = 1                                                         MAIN1970
      CALL OVERLAY(4HOVLY,8,0)                                          MAIN1980
      IF(IFFLD .GT. 0)CALL UNLOAD(NEWMAX,1)                             MAIN1990
C  SAVE THE JET-WING GAMMA SOLUTION FOR LATER USE                       MAIN2000
      CALL SAVECP(GAMMA,CPREAD,NMAX,NJT,NEWMAX,NCASES,1)                MAIN2010
C  INTEGRATE THE WING LOADINGS TO FORM AERODYNAMIC PARAMETERS           MAIN2020
      LOGIC = 3                                                         MAIN2030
      IF(STAB)LOGIC = 2                                                 MAIN2040
      CALL OVERLAY(4HOVLY,3,0)                                          MAIN2050
C  SET UP THE TAIL GEOMETRY AND MATRIX                                  MAIN2060
  230 IF((.NOT.TAIL) .OR. (NEWCMU.GT.1) .OR. (NHITE.GT.1)) GO TO 250    MAIN2070
      ISENSE = 1                                                        MAIN2080
      LOGIC=1                                                           MAIN2090
      IF(STAB)LOGIC=2                                                   MAIN2100
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2110
C  SOLVE THE TAIL WITHOUT WING, JET-WING, OR JET INFLUENCES             MAIN2120
      IF(.NOT. STAB) GO TO 240                                          MAIN2130
      ISTAB = 1                                                         MAIN2140
      ISENSE = 5                                                        MAIN2150
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2160
      LOGIC = 2                                                         MAIN2170
      CALL OVERLAY(4HOVLY,8,0)                                          MAIN2180
      ISENSE = 6                                                        MAIN2190
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2200
  240 ISENSE = 2                                                        MAIN2210
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2220
      LOGIC = 2                                                         MAIN2230
      ISTAB = 0                                                         MAIN2240
      CALL OVERLAY(4HOVLY,8,0)                                          MAIN2250
      ISENSE = 3                                                        MAIN2260
      IF(STAB) ISTAB = 1                                                MAIN2270
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2280
      ISTAB = 0                                                         MAIN2290
C  RETRIEVE THE JET-WING GAMMA SOLUTION FROM DISK                       MAIN2300
  250 IF(NOWING) GO TO 400                                              MAIN2310
      IF(NOTEVD) GO TO 260                                              MAIN2320
      CALL SAVECP(GAMMA,CPREAD,NMAX,NJT,NEWMAX,NCASES,2)                MAIN2330
      GO TO 270                                                         MAIN2340
  260 CALL UNLOAD(NEWMAX,2)                                             MAIN2350
C  COMPUTE THE NON-LINEAR TAIL AND FUSELAGE LOADING FOR ALL WING        MAIN2360
C  COMPOSITE CASES WITH VECTORED JETS OFF                               MAIN2370
C                                                                       MAIN2380
  270 IF(.NOT. FLOFLD) GO TO 310                                        MAIN2390
      ISENSE = 4                                                        MAIN2400
```

```
      IWRITE = 2*NMAX + NSEG + 1                                        MAIN2410
      DO 300 KLASE = 1,ICASES                                           MAIN2420
C   COMPUTE THE WING FLOW FIELD INFLUENCE                               MAIN2430
      LOGIC = 1                                                         MAIN2440
      CALL OVERLAY(4HOVLY,5,0)                                          MAIN2450
C   READ THE VT POWER-OFF WING TOTAL COEFFICIENTS                       MAIN2460
      IF(SUMARY) CALL READMS(9,POT,38,IWRITE)                           MAIN2470
C   COMPUTE THE NON-LINEAR TAIL LOADING                                 MAIN2480
      LOGIC = 2                                                         MAIN2490
      IF(.NOT. TAIL) GO TO 280                                          MAIN2500
      CALL OVERLAY(4HOVLY,9,0)                                          MAIN2510
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2520
C   COMPUTE THE NON-LINEAR FUSELAGE LOADING                             MAIN2530
  280 IF(.NOT FSLG) GO TO 290                                           MAIN2540
      LOGIC = 1                                                         MAIN2550
      IF(ISTAB .AND. (KLASE EQ.1)) LOGIC = 2                            MAIN2560
      CALL OVERLAY(4HOVLY,10,0)                                         MAIN2570
C   STORE THE VT POWER-OFF WING AND TAIL AND FUSELAGE TOTAL COEFFICIENTS MAIN2580
  290 IF(SUMARY) CALL WRITMS(9,POT,102,IWRITE)                          MAIN2590
      IWRITE = IWRITE + 1                                               MAIN2600
  300 CONTINUE                                                          MAIN2610
C                                                                       MAIN2620
C   COMPUTE THE PATHS OF THE CIRCULAR JETS AND THEIR FLOW FIELD INFLUENCEMAIN2630
  310 IF(.NOT. VT)GO TO 360                                             MAIN2640
      IF(NEWCMU .EQ. 1)NEWCMU = 2                                       MAIN2650
      NCM1 = NEWCMU - 1                                                 MAIN2660
      IF(NEWCMU .EQ. 2) WRITE(6, 130 ) NCM1                             MAIN2670
      IF(NEWCMU .EQ. 2) WRITE(6, 320 )(CJ(N),UJ(N),DDO(N),THO(N),DIHE(N)MAIN2680
     1,               IS(N),XO(N),YO(N),ZO(N),H(N),N=1,NJETS)           MAIN2690
  320 FORMAT(1HO// 43X, 8(4H****),2H** /                                MAIN2700
     1       43X,34H*  VECTORED JET CHARACTERISTICS   * / 43X,8(4H****),  MAIN2710
     2       2H** / 13X, 2HCJ,10X, 3HUJO, 9X, 6HDO/B/2, 6X, 3HTHO, 9X,   MAIN2720
     3       5HDIHED, 6X, 3HISY, 4X, 6HXO/B/2, 6X, 6HYO/B/2,             MAIN2730
     4       6X, 6HZO/B/2, 6X, 4HDELZ ,10(/ 8X,5F12.6, I6,4F12.6))       MAIN2740
      CALL OVERLAY(4HOVLY,7,0)                                          MAIN2750
C   BACK SOLVE THE WING MATRIX FOR CIRCULAR JET INFLUENCE               MAIN2760
      LOGIC = 1                                                         MAIN2770
      CALL OVERLAY(4HOVLY,9,0)                                          MAIN2780
C                                                                       MAIN2790
C   COMPUTE THE NON-LINEAR WING, TAIL AND FUSELAGE LOADING FOR ALL WING MAIN2800
C   COMPOSITE CASES WITH VECTORED JETS ON                               MAIN2810
      ISENSE = 4                                                        MAIN2820
      IWRITE = 2*NMAX + NSEG + NCC + 1                                  MAIN2830
      DO 350 KCASE = 1,ICASES                                           MAIN2840
C   COMPUTE THE NEW NON-LINEAR WING LOADING                             MAIN2850
      LOGIC = 1                                                         MAIN2860
      CALL OVERLAY(4HOVLY,3,0)                                          MAIN2870
C   COMPUTE THE WING FLOW FIELD INFLUENCE                               MAIN2880
      LOGIC = 2                                                         MAIN2890
      IF(FLOFLD) CALL OVERLAY(4HOVLY,5,0)                               MAIN2900
C   COMPUTE THE NON-LINEAR TAIL LOADING                                 MAIN2910
      IF(.NOT. TAIL) GO TO 330                                          MAIN2920
      LOGIC = 2                                                         MAIN2930
      CALL OVERLAY(4HOVLY,9,0)                                          MAIN2940
      CALL OVERLAY(4HOVLY,6,0)                                          MAIN2950
C   COMPUTE THE NON-LINEAR FUSELAGE LOADING                             MAIN2960
  330 IF(.NOT. FSLG) GO TO 340                                          MAIN2970
      LOGIC = 1                                                         MAIN2980
      CALL OVERLAY(4HOVLY,10,0)                                         MAIN2990
C   STORE THE VT POWER-ON WING AND TAIL AND FUSELAGE TOTAL COEFFICIENTS MAIN3000
```

```
  340 IF(SUMARY) CALL WRITMS(9,POT,102,IWRITE)                          MAIN3010
      IWRITE = IWRITE + 1                                               MAIN3020
  350 CONTINUE                                                          MAIN3030
C                                                                       MAIN3040
C   EXECUTE THE SECOND RUN FOR STABILITY DERIVATIVES                    MAIN3050
  360 IF(NOTEVU) GO TO 390                                              MAIN3060
      IF(.NOT  STAB) GO TO 380                                          MAIN3070
      WRITE(6, 370 )                                                    MAIN3080
  370 FORMAT(1H1///////////// 37X,11(4H****),2H** /                     MAIN3090
     1       37X,46H*  SECOND RUN FOR STABILITY DERIVATIVE CASE   * /    MAIN3100
     2       37X,11(4H****),2H**)                                       MAIN3110
      IF(ISYMM .EQ. 0) ISYMM = -1                                       MAIN3120
      ISTAB = 1                                                         MAIN3130
C   DEFINE THE NEW CASES FOR ROLLING AND YAWING RATES                   MAIN3140
      LOGIC = 2                                                         MAIN3150
      CALL OVERLAY(4HOVLY,4,0)                                          MAIN3160
C   PRINT GEOMETRY DATA FOR ALL FUNDAMENTAL CASES                       MAIN3170
      LOGIC=3                                                           MAIN3180
      IF((NEWCMU EQ.1).AND.(NHITE EQ.1).AND.(IPRINT.LT.0))              MAIN3190
     1     CALL OVERLAY(4HOVLY,1,0)                                     MAIN3200
      IF((NEWCMU.EQ.2).AND.(NHITE.EQ.1).AND.(IPRINT.LT.0).AND.VT)       MAIN3210
     1     CALL OVERLAY(4HOVLY,1,0)                                     MAIN3220
C   DEFINE NEW RIGHT HAND SIDES FOR STABILITY RUN                       MAIN3230
      LOGIC = 2                                                         MAIN3240
      CALL OVERLAY(4HOVLY,2,0)                                          MAIN3250
C   SOLVE THE JET-WING DOWNWASH MATRIX FOR STABILITY (ANTI-SYMETRIC)    MAIN3260
      LOGIC = 1                                                         MAIN3270
      CALL OVERLAY(4HOVLY,8,0)                                          MAIN3280
C   CALCULATE AND PRINT THE DERIVATIVES FOR ALL THE FUNDAMENTAL AND     MAIN3290
C   COMPOSITE CASES                                                     MAIN3300
      LOGIC = 1                                                         MAIN3310
      IF(VT) LOGIC = 2                                                  MAIN3320
      IF(.NOT. SUMARY) LOGIC = 3                                        MAIN3321
      CALL OVERLAY(4HOVLY,11,0)                                         MAIN3330
C   SUMMARIZE THE TOTAL AIRCRAFT STATIC AND DYNAMIC COEFFICIENTS        MAIN3340
  380 LOGIC = 1                                                         MAIN3350
      IF(VT) LOGIC = 2                                                  MAIN3360
      IF(SUMARY) CALL OVERLAY(4HOVLY,12,0)                              MAIN3370
C                                                                       MAIN3380
C   THE PROGRAM HAS BEEN SUCCESSFULLY EXECUTED                          MAIN3390
C   GO BACK FOR A NEW JET OR GROUND HEIGHT CASE                         MAIN3400
  390 IF(.NOT.GE) GO TO 100                                             MAIN3410
      GO TO 150                                                         MAIN3420
C   SOLVE THE FUSELAGE WITH NO WING                                     MAIN3430
  400 IF(.NOT. FSLG) GO TO 450                                          MAIN3440
      DO 420 J = 1,5                                                    MAIN3450
      DO 410 K = 1,NSEG                                                 MAIN3460
      V(K,J)-= 0.0                                                      MAIN3470
  410 W(K,J) = 0.0                                                      MAIN3480
  420 CONTINUE                                                          MAIN3490
      KCASE = 0                                                         MAIN3500
      LOGIC = 1                                                         MAIN3510
      IF(STAB) LOGIC = 2                                                MAIN3520
      CALL OVERLAY(4HOVLY,10,0)                                         MAIN3530
      GO TO 450                                                         MAIN3540
C                                                                       MAIN3550
C   AN INCONSISTENCY HAS BEEN FOUND IN THE BASIC CONTROL FLAGS          MAIN3560
  430 WRITE(6, 440 )                                                    MAIN3570
  440 FORMAT(1HO////////// 30X,                                         MAIN3580
     1  60HAN INCONSISTENCY HAS BEEN FOUND IN THE GENERAL CONTROL FLAGS/MAIN3590
```

```
     2   33X,53HPLEASE RECHECK YOUR INPUT ON THE GENERAL CONTROL CARDS) MAIN3600
      GO TO 490                                                         MAIN3610
C   THIS RUN HAS BEEN COMPLETED.  THANK GOD FOR SMALL BLESSINGS.        MAIN3620
C   PRINT COMPLETION MESSAGE FOR THIS RUN AND GO BACK TO BEGIN A NEW RUN MAIN3630
  450 WRITE(6, 460 )                                                    MAIN3640
  460 FORMAT(1HO//// 32X,10(5H*****), 3H***/ 32X,                       MAIN3650
     1        53H*  THE  PROGRAM  HAS  REACHED  NORMAL  TERMINATION  */  MAIN3660
     2        32X,10(5H*****),3H***)                                    MAIN3670
C   READ TO SEE IF THE NEXT CARD IS A TITLE OR AN OLD END OF CMU CARD   MAIN3680
      READ(5, 30 ) TITLE                                                MAIN3690
      IF(EOF(5)) 480 , 470                                              MAIN3700
  470 IF(TITLE(1) .EQ. CHECK) GO TO 20                                  MAIN3710
      GO TO 40                                                          MAIN3720
C   PRINT COMPLETION MESSAGE AND STOP EXECUTION                         MAIN3730
  480 WRITE(6, 460 )                                                    MAIN3740
      GO TO 510                                                         MAIN3750
C   A FATAL ERROR HAS OCCURED.  PRINT FINAL MESSAGE AND STOP EXECUTION. MAIN3760
  490 WRITE(6, 500 )                                                    MAIN3770
  500 FORMAT(1HO///62X,2(4H****)/31X,11(5H*****)/                       MAIN3780
     1   31X,55H*  THE  PROGRAM  HAS  REACHED  ABNORMAL  TERMINATION  */MAIN3790
     2   31X,11(5H*****)/62X,2(4H****))                                 MAIN3800
  510 STOP                                                              MAIN3810
      END                                                               MAIN3820
```

```
*DECK HEADER                                                            HEAD0010
      SUBROUTINE HEADER(ILINES)                                         HEAD0020
C                                                                       HEAD0030
C   THIS SUBROUTINE PRINTS BASIC INFORMATION AT THE TOP OF A PAGE       HEAD0040
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                HEAD0050
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          HEAD0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            HEAD0070
      COMMON/DATTIM/MDY,HMS                                             HEAD0080
      ILINES = 2                                                        HEAD0090
      WRITE(6, 20 )MDY,HMS                                              HEAD0100
   20 FORMAT(1H1,21X,32HSTOL AERODYNAMIC METHODS PROGRAM,               HEAD0110
     1         39H  *  DOUGLAS AIRCRAFT COMPANY  *  DATE ,A10,          HEAD0120
     2         10H  *  TIME ,A10)                                       HEAD0130
      JF1 = JETFLG + 1                                                  HEAD0140
      GO TO ( 30 , 50 , 70 ),JF1                                        HEAD0150
   30 WRITE(6, 40 )NEWCMU                                               HEAD0160
   40 FORMAT(1H ,15HCMU CASE NUMBER, I3)                                HEAD0170
      GO TO 90                                                          HEAD0180
   50 WRITE(6, 60 )                                                     HEAD0190
   60 FORMAT(1H ,28HNO VECTORED JET OR JET SHEET)                       HEAD0200
      GO TO 90                                                          HEAD0210
   70 NCM1 = NEWCMU - 1                                                 HEAD0220
      IF(NEWCMU .EQ. 1) WRITE(6, 60 )                                   HEAD0230
      IF(NEWCMU .GT. 1) WRITE(6, 80 )NCM1                               HEAD0240
   80 FORMAT(1H ,24HVECTORED JET CASE NUMBER,I3)                        HEAD0250
   90 IF(NHITE .LT. 2) WRITE(6, 100 )                                   HEAD0260
  100 FORMAT(1H+,57X,16HNO GROUND EFFECT)                               HEAD0270
      IF(NHITE .GT. 1) WRITE(6, 110 ) GHITE(NHITE)                      HEAD0280
  110 FORMAT(1H+,52X,15HGROUND HEIGHT =,F10.6)                          HEAD0290
      IF(ISTAB .EQ. 0) WRITE(6, 120 )                                   HEAD0300
  120 FORMAT(1H+,105X,24HNO STABILITY DERIVATIVES)                      HEAD0310
      IF(ISTAB .NE. 0) WRITE(6, 130 )                                   HEAD0320
  130 FORMAT(1H+,105X,24HSTABILITY DERIVATIVE RUN)                      HEAD0330
```

```
      RETURN                                                            HEAD0340
      END                                                               HEAD0350
```

```
*DECK SAVECP                                                            SAVC0010
      SUBROUTINE SAVECP(CP,DUMMY,NMAX,NJT,NEWMAX,NCASES,IGOTO)          SAVC0020
C                                                                       SAVC0030
C   THIS SUBROUTINE SAVES THE CP SOLUTION FOR ALL FUNDAMENTAL CASES     SAVC0040
C                                                                       SAVC0050
      DIMENSION CP(400,10),DUMMY(NEWMAX)                                SAVC0060
C                                                                       SAVC0070
C   FIND THE PROPER PLACE TO WRITE THE OLD SOLUTION                     SAVC0080
   20 IWRITE = 2 * (NMAX+NJT)                                           SAVC0090
      GO TO ( 30 , 60 ),IGOTO                                           SAVC0100
C   DEFINE THE DUMMY ARRAY                                              SAVC0110
   30 DO 50 N = 1,NCASES                                                SAVC0120
      IWRITE = IWRITE + 1                                               SAVC0130
      DO 40 I = 1,NEWMAX                                                SAVC0140
      DUMMY(I) = CP(I,N)                                                SAVC0150
   40 CONTINUE                                                          SAVC0160
C   SAVE THE DATA                                                       SAVC0170
      CALL WRITMS(1,DUMMY,NEWMAX,IWRITE)                                SAVC0180
   50 CONTINUE                                                          SAVC0190
      RETURN                                                            SAVC0200
   60 DO 80 N = 1,NCASES                                                SAVC0210
      IWRITE = IWRITE + 1                                               SAVC0220
      CALL READMS(1,DUMMY,NEWMAX,IWRITE)                                SAVC0230
      DO 70 I = 1,NEWMAX                                                SAVC0240
      CP(I,N) = DUMMY(I)                                                SAVC0250
   70 CONTINUE                                                          SAVC0260
   80 CONTINUE                                                          SAVC0270
      RETURN                                                            SAVC0280
      END                                                               SAVC0290
```

```
*DECK UNLOAD                                                            UNL00010
      SUBROUTINE UNLOAD(NEWMAX,IGOTO)                                   UNL00020
C                                                                       UNL00030
C   THIS SUBROUTINE EITHER READS OR WRITES THE LOADING DATA FOR USE IN A UNL00040
C   FLOW FIELD ONLY RUN                                                 UNL00050
C                                                                       UNL00060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            UNL00070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) UNL00080
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           UNL00090
      COMMON/SOLV1/CP(400,10)                                           UNL00100
      COMMON/SOLV3/DUMMY(400)                                           UNL00110
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         UNL00120
     1              IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),              UNL00130
     2              XTR(40),XTL(40),CHR(40),CHL(40)                     UNL00140
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          UNL00150
      GO TO ( 20 , 80 ),IGOTO                                           UNL00160
   20 IF(JETFLG .EQ. 1) GO TO 60                                        UNL00170
      DO 30 K = 1,NROWSJ                                                UNL00180
   30 DUMMY(K) = 0.0                                                    UNL00190
      KK = 0                                                            UNL00200
      DO 40 K = 1,NROWS                                                 UNL00210
```

```
      IF(CMU(K) .LT. 0.01)GO TO 40                                      UNL00220
      KK = KK + 1                                                       UNL00230
      DUMMY(KK) = CMU(K)                                                UNL00240
   40 CONTINUE                                                          UNL00250
      WRITE(7, 50 )(DUMMY(K),K = 1,NROWSJ)                              UNL00260
   50 FORMAT(8F10.6)                                                    UNL00270
   60 DO 70 N = 1,NCASES                                                UNL00280
      WRITE(7, 50 )(CP(I,N),I = 1,NEWMAX)                               UNL00290
   70 CONTINUE                                                          UNL00300
      RETURN                                                            UNL00310
   80 DO 90 N = 1,NCASES                                                UNL00320
      READ(5, 50 )(CP(I,N),I = 1,NEWMAX)                                UNL00330
   90 CONTINUE                                                          UNL00340
      IF(IGND .GT. 0) RETURN                                            UNL00350
      DO 100 N = 1,NCASES                                               UNL00360
      READ(5, 50 )(AINF(I,N),I = 1,NROWS)                               UNL00370
  100 CONTINUE                                                          UNL00380
      RETURN                                                            UNL00390
      END                                                               UNL00400
```

```
*DECK SAVE                                                              SAVE0010
      SUBROUTINE  SAVE(IU, IT, N, N1, A1, N2, A2)                       SAVE0020
C                                                                       SAVE0030
      DIMENSION  A1(N1), A2(N2)                                         SAVE0040
C                                                                       SAVE0050
      GO TO ( 20 , 30 , 40 , 50 ), IT                                   SAVE0060
C                                                                       SAVE0070
C  WRITE A1                                                             SAVE0080
   20 WRITE(IU) A1                                                      SAVE0090
      RETURN                                                            SAVE0100
C                                                                       SAVE0110
C  WRITE N AND A1                                                       SAVE0120
   30 WRITE(IU)  N, A1                                                  SAVE0130
      RETURN                                                            SAVE0140
C                                                                       SAVE0150
C  WRITE A1 AND A2                                                      SAVE0160
   40 WRITE(IU) A1, A2                                                  SAVE0170
      RETURN                                                            SAVE0180
C                                                                       SAVE0190
C  WRITE N, A1, AND A2                                                  SAVE0200
   50 WRITE(IU) N, A1, A2                                               SAVE0210
      RETURN                                                            SAVE0220
      END                                                               SAVE0230
```

```
*DECK GETT                                                              GETT0010
      SUBROUTINE GETT(IU, IT, N1, A1, N2, A2)                           GETT0020
C                                                                       GETT0030
      DIMENSION  A1(N1), A2(N2)                                         GETT0040
C                                                                       GETT0050
      GO TO ( 20 , 30 , 40 , 50 ), IT                                   GETT0060
C                                                                       GETT0070
C  READ A1                                                              GETT0080
   20 READ(IU) A1                                                       GETT0090
      RETURN                                                            GETT0100
C                                                                       GETT0110
C  READ N1 AND A1                                                       GETT0120
   30 READ(IU) N1, A1                                                   GETT0130
      RETURN                                                            GETT0140
C                                                                       GETT0150
C  READ A1 AND A2                                                       GETT0160
   40 READ(IU) A1, A2                                                   GETT0170
      RETURN                                                            GETT0180
C                                                                       GETT0190
C  READ IDUM AND A1                                                     GETT0200
   50 READ(IU) IDUM, A1                                                 GETT0210
      RETURN                                                            GETT0220
      END                                                               GETT0230
```

```
*DECK STAGE1                                                            STG10010
      OVERLAY(OVLY,1,0)                                                 STG10020
      PROGRAM STAGE1                                                    STG10030
C                                                                       STG10040
C  THIS SUBROUTINE READS THE GENERAL GEOMETRY PARAMETERS AND FLAGS, AND STG10050
C  CONTROLS THE CALLING OF THE SPECIALIZED GEOMETRY SUBROUTINES         STG10060
C                                                                       STG10070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                STG10080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            STG10090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) STG10100
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             STG10110
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         STG10120
     1             IFFLD,ICR,AINF(40,10),XLR(40),XLL(40),               STG10130
     2             XTR(40),XTL(40),CHR(40),CHL(40)                      STG10140
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          STG10150
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     STG10160
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    STG10170
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES              STG10180
      COMMON/COMPOS/FACTOR(10,24),NCC                                   STG10190
      COMMON/FUSLG1/PI,NSEG,NSEGL,NF(40),ISECT,ICAMF,XB(41),ZCL(41),    STG10200
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),               STG10210
     2              V(40,7),W(40,7)                                     STG10220
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           STG10230
      COMMON/JET12/UJ(10),DDO(10),THTA(10),DZ(10),ZMX(10),X0(10),Y0(10),STG10240
     1            Z0(10),DIHE(10),ISY(10),JETRE(10),NJETS               STG10250
      COMMON/JET13/CJ(10),THO(10)                                       STG10260
C                                                                       STG10270
C  CHECK WHETHER THIS IS THE FIRST CMU CASE                             STG10280
      IF((NEWCMU .GT. 1) .AND. (JETFLG .EQ. 0)) GO TO 90                STG10290
      IF((NEWCMU .GT. 2) .AND. (JETFLG .EQ. 2)) GO TO 100               STG10300
      IF(LOGIC .EQ. 3) GO TO 30                                         STG10310
      IF(LOGIC .EQ. 4) GO TO 50                                         STG10320
C  SECTIONAL INPUT                                                      STG10330
   20 CALL SGMAIN(NOALFA,IR)                                            STG10340
      IF(IR .EQ. 3) GO TO 190                                           STG10350
   30 IF(LOGIC .NE. 2) CALL OUT1                                        STG10360
      IF(LOGIC .EQ. 3) GO TO 130                                        STG10370
C  READ THE COMPOSITE CASE REQUIREMENTS                                 STG10380
   40 CALL INCOMP(NCASES,IR)                                            STG10390
      ICASES = NCC                                                      STG10400
      IF(IR .EQ. 2) GO TO 190                                           STG10410
   50 IF((NROWH .NE. 0) .OR. (NROWV .NE. 0))CALL INTAIL                 STG10420
      IF(IR .EQ. 2)GO TO 190                                            STG10430
      IF((NROWH .EQ. 0) .AND. (NROWV .EQ. 0))NTOTAL = 0                 STG10440
      IF(NSEG .NE. 0)CALL INFUS                                         STG10450
      NTPNP = NTOTAL + NSEG + NPOINT                                    STG10460
C  IF THERE IS NO WING, NORMALIZE PARAMETERS NEEDED BY TAIL OR FUSELAGE STG10470
      IF(LOGIC .NE. 4) GO TO 80                                         STG10480
      DO 60 K = 1,NROWS                                                 STG10490
      XLEAD(K) = 0.0                                                    STG10500
   60 XTRAIL(K) = 0.0                                                   STG10510
      DO 70 K = 1,10                                                    STG10520
   70 GHITE(K) = 0.0                                                    STG10530
      CALL NORM1                                                        STG10540
      CALL OUT1                                                         STG10550
      GO TO 220                                                         STG10560
C  READ THE FLOW FIELD REQUIREMENTS AND COMPUTE THE GEOMETRY PARAMETERS STG10580
   80 IF(NTPNP .NE. 0)CALL XEDGE                                        STG10590
C  READ THE CMU DATA                                                    STG10600
   90 IF(JETFLG .LT. 2) CALL BLOWIN(JETFLG,IR)                          STG10610
C  READ THE CIRCULAR JET DATA                                           STG10620
  100 IF(JETFLG .EQ. 2) CALL INJETS(NEWMAX,IR)                          STG10630
      GO TO ( 110 , 200 , 210 , 120 , 190 ), IR                         STG10640
  110 CALL BOXJ(NEWMAX,IR)                                              STG10650
  120 IF(LOGIC .EQ. 2) GO TO 170                                        STG10660
  130 IF(JETFLG .EQ. 2) GO TO 150                                       STG10670
      CALL HEADER(ILINES)                                               STG10680
      WRITE(6, 140 ) (K,CMU(K),K=1,NROWS)                               STG10690
  140 FORMAT(1H0//41X,10(4H****)/ 41X,                                  STG10700
     1      40H*  SECTIONAL JET BLOWING COEFFICIENTS  */41X,10(4H****)//STG10710
     2      53X,3HROW,5X,3HCMU,40(/53X,I2,F12.6))                       STG10720
      GO TO 170                                                         STG10730
  150 IF(NEWCMU .EQ. 1) GO TO 170                                       STG10740
      CALL HEADER(ILINES)                                               STG10750
      WRITE(6, 160 ) (CJ(N),UJ(N),DDO(N),THO(N),DIHE(N),                STG10760
     1                ISY(N),X0(N),Y0(N),Z0(N),DZ(N),N=1,NJETS)         STG10770
  160 FORMAT(1H0// 43X, 8(4H****),2H** /                                STG10780
     1      43X,34H*  VECTORED JET CHARACTERISTICS  * / 43X,8(4H****),  STG10790
     2      2H** / 13X, 2HCJ,10X, 3HUJD, 9X, 6HDO/B/2, 6X, 3HTHO, 9X,   STG10800
     3      5HDIHED, 6X, 3HISY, 4X, 6HXO/B/2, 6X, 6HYO/B/2,             STG10810
     4      6X, 6HZO/B/2, 6X, 4HDELZ ,10(/ 8X,5F12.6, I6,4F12.6))       STG10820
  170 IF(IR .EQ. 2) GO TO 190                                           STG10830
C  RETURN NORMALLY TO THE CONTROL PROGRAM                               STG10840
  180 IR = 1                                                            STG10850
      GO TO 220                                                         STG10860
C  A FATAL ERROR HAS OCCURED.   RETURN ABNORMALLY TO MAIN.              STG10870
  190 IR = 4                                                            STG10880
      GO TO 220                                                         STG10890
C  THIS RUN HAS BEEN COMPLETED.   THANK GOD FOR SMALL BLESSINGS.        STG10900
C  RETURN TO MAIN AND BEGIN A COMPLETELY NEW RUN                        STG10910
  200 IR = 2                                                            STG10920
      GO TO 220                                                         STG10930
C  RETURN TO MAIN AND STOP THE EXECUTION                                STG10940
  210 IR = 3                                                            STG10950
  220 CONTINUE                                                          STG10960
      END                                                               STG10970
```

```
*DECK SGMAIN                                                            SGMN0010
      SUBROUTINE SGMAIN(NOALFA,IR)                                      SGMN0020
C                                                                       SGMN0030
C  THIS SUBROUTINE CONTROLS ALL GEOMETRY CALCULATIONS FOR THE           SGMN0040
C  SECTIONAL GEOMETRY METHOD                                            SGMN0050
C                                                                       SGMN0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            SGMN0070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) SGMN0080
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTB                  SGMN0090
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          SGMN0100
      COMMON/CASES/INT(10),INH(10),IND(10),INL(10),INB(10),IHLE(10),    SGMN0110
     1             IHTE(10)                                             SGMN0120
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  SGMN0130
     1             D(40),KK(400),ITYPE(400)                             SGMN0140
C                                                                       SGMN0150
C  READ THE WING PLANFORM GEOMETRY DATA                                 SGMN0160
   20 CALL INPTS(IR)                                                    SGMN0170
      IF(IR .EQ. 2) GO TO 110                                           SGMN0180
      IF(IGTYPE .EQ. 1) CALL XLETR1(IR)                                 SGMN0190
      IF(IR .EQ. 2) GO TO 110                                           SGMN0200
      IF(IGTYPE .EQ. 2) CALL XLETR2                                     SGMN0210
C  READ THE GROUND HEIGHTS                                              SGMN0220
      DO 30 I = 1,10                                                    SGMN0230
      GHITE(I) = 0.00                                                   SGMN0240
   30 CONTINUE                                                          SGMN0250
      IF(IGRND .NE. 0) READ(5, 40 ) (GHITE(I),I=2,9)                    SGMN0260
   40 FORMAT(8F10.6)                                                    SGMN0270
C  NORMALIZE THE WING PLANFORM GEOMETRY DATA                            SGMN0280
   50 CALL NORM1                                                        SGMN0290
C  READ THE JET SHEET GEOMETRY DATA                                     SGMN0300
   60 CALL INPUTJ(IR)                                                   SGMN0310
      IF(IR .EQ. 2) GO TO 110                                           SGMN0320
C  CONSTRUCT THE EVD ELEMENTS                                           SGMN0330
   70 CALL BOXS(IR)                                                     SGMN0340
      IF(IR .EQ. 2) GO TO 110                                           SGMN0350
C  CONSTRUCT THE SET OF FUNDAMENTAL GEOMETRIC CASES                     SGMN0360
      DO 100 N=1,NCASES                                                 SGMN0370
      LCASE = N                                                         SGMN0380
C  READ THE GEOMETRY FOR THIS CASE                                      SGMN0390
   80 CALL INCASE(LCASE,NOALFA)                                         SGMN0400
C  CONSTRUCT THE CASE DATA                                              SGMN0410
      ILE=0                                                             SGMN0420
      ITE=0                                                             SGMN0430
   90 CALL BEECEE(LCASE,NOALFA,IR,ILE,ITE)                              SGMN0440
      IF(IR .EQ. 2) GO TO 110                                           SGMN0450
C  SET UP THE ARRAYS FOR OUTPUT DESCRIPTION OF EACH FUNDAMENTAL CASE    SGMN0460
      IF(N.EQ.1)GO TO 100                                               SGMN0470
      INT(N)=INPUTT                                                     SGMN0480
      INH(N)=INPUTH                                                     SGMN0490
      IND(N)=INPUTD                                                     SGMN0500
      INC(N)=INPUTC                                                     SGMN0510
      INB(N)=INPUTB                                                     SGMN0520
      IHLE(N)=ILE                                                       SGMN0530
      IHTE(N)=ITE                                                       SGMN0540
  100 CONTINUE                                                          SGMN0550
      IR = 2                                                            SGMN0560
      RETURN                                                            SGMN0570
C  AN ERROR HAS OCCURED.   RETURN ABNORMALLY TO STAGE1.                 SGMN0580
  110 IR = 3                                                            SGMN0590
      RETURN                                                            SGMN0600
      END                                                               SGMN0610
```

```
*DECK INPTS                                                             INPT0010
```

```
      SUBROUTINE INPTS(IR)                                              INPT0020
C                                                                       INPT0030
C  THIS SUBROUTINE READS THE WING GEOMETRY DATA                         INPT0040
C  FOR THE SECTIONAL GEOMETRY METHOD                                    INPT0050
C                                                                       INPT0060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) INPT0070
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  INPT0080
     1             D(40),KK(400),ITYPE(400)                             INPT0090
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             INPT0100
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),           INPT0110
     1           NWTYPE,NJTYPE                                          INPT0120
      DIMENSION NI(10)                                                  INPT0130
C                                                                       INPT0140
C  READ THE SECTIONAL PLANFORM DATA                                     INPT0150
   20 NWTYPE = 0                                                        INPT0160
      READ(5, 30 ) (Y(K),K=1,NROWS)                                     INPT0170
   30 FORMAT(8F10.6)                                                    INPT0180
      READ(5, 40 ) (ICTYPE(K),K=1,NROWS)                                INPT0190
   40 FORMAT(40I2)                                                      INPT0200
      DO 50 K = 1,NROWS                                                 INPT0210
      IF(ICTYPE(K) .GT. NWTYPE) NWTYPE = ICTYPE(K)                      INPT0220
   50 CONTINUE                                                          INPT0230
      IF(NWTYPE .GT. 10) GO TO 90                                       INPT0240
      READ(5, 40 )(NI(N),N=1,NWTYPE)                                    INPT0250
C                                                                       INPT0260
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE                   INPT0270
      DO 60 N = 1,NWTYPE                                                INPT0280
      NIN = NI(N)                                                       INPT0290
      IF((NIN .LT. 1) .OR. (NIN .GT. 20)) GO TO 110                     INPT0300
      READ(5, 30 ) (XBW(L,N),L=1,NIN)                                   INPT0310
   60 CONTINUE                                                          INPT0320
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW                INPT0330
      DO 80  K = 1,NROWS                                                INPT0340
      ICK = ICTYPE(K)                                                   INPT0350
   70 NW(K) = NI(ICK)                                                   INPT0360
   80 CONTINUE                                                          INPT0370
      IR = 1                                                            INPT0380
      RETURN                                                            INPT0390
C                                                                       INPT0400
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.                    INPT0410
   90 WRITE(6, 100 ) NWTYPE                                             INPT0420
  100 FORMAT(1H1/45X,26HNUMBER OF WING ROW TYPES =,I3)                  INPT0430
      IR = 2                                                            INPT0440
      RETURN                                                            INPT0450
  110 WRITE(6, 120 ) NIN,N                                              INPT0460
  120 FORMAT(1H1/38X,I3,30H WING ELEMENTS PRESCRIBED FOR ROW TYPE,I3)   INPT0470
      IR = 2                                                            INPT0480
      RETURN                                                            INPT0490
      END                                                               INPT0500
```

```
*DECK XLETR1                                                            XLT10010
      SUBROUTINE XLETR1(IR)                                             XLT10020
C                                                                       XLT10030
C  THIS SUBROUTINE READS THE LEADING AND TRAILING EDGE COORDINATES      XLT10040
C  AT SPANWISE STATIONS CONNECTED BY STRAIGHT LEADING AND TRAILING EDGESXLT10050
C  AND INTERPOLATES TO GET THE COORDINATES AT INTERMEDIATE SECTIONS     XLT10060
C                                                                       XLT10070
```

```
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) XLT10080
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  XLT10090
     1             D(40),KK(400),ITYPE(400)                             XLT10100
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             XLT10110
      DIMENSION YP(40),XLE(40),XTR(40)                                  XLT10120
C                                                                       XLT10130
C  READ XLEAD AND XTRAIL                                                XLT10140
      NX = 0                                                            XLT10150
      DO 40 N = 1,NROWS                                                 XLT10160
      READ(5, 20 )YP(N),XLE(N),XTR(N)                                   XLT10170
   20 FORMAT(3F10.6)                                                    XLT10180
   30 IF(YP(N) .GT. 1.1) GO TO 50                                       XLT10190
      NX = NX + 1                                                       XLT10200
   40 CONTINUE                                                          XLT10210
C  CHECK WHETHER THE Y VALUES ARE REALISTIC                             XLT10220
   50 IF(ABS(YP(1)-Y(1)) .GT. 0.0001)GO TO 120                          XLT10230
      IF(ABS(YP(NX)-Y(NROWS)) .GT. 0.0001)GO TO 120                     XLT10240
C  READ THE EXTRA 9 CARD IF NROWS CARDS HAVE BEEN INPUT                 XLT10250
      IF(NX .EQ. NROWS) READ(5, 20 ) EXTRA9                             XLT10260
C  INTERPOLATE FOR XLEAD AND XTRAIL AT THE INTERMEDIATE SECTIONS        XLT10270
      K = 0                                                             XLT10280
      DO 110 N = 1,NX                                                   XLT10290
   60 K = K + 1                                                         XLT10300
      IF(K .GT. NROWS) GO TO 120                                        XLT10310
      IF(ABS(YP(N)-Y(K)) .GT. 0.0001)GO TO 80                           XLT10320
C  XLE AND XTR WERE INPUT FOR ROW K                                     XLT10330
   70 XLEAD(K) = XLE(N)                                                 XLT10340
      XTRAIL(K) = XTR(N)                                                XLT10350
      GO TO 110                                                         XLT10360
C  XLE AND XTR MUST BE INTERPOLATED FOR ROW K                           XLT10370
   80 NMI = N - 1                                                       XLT10380
   90 YRATIO  = (Y(K)-YP(N)) / (YP(NMI)-YP(N))                          XLT10390
      XLEAD(K) = XLE(N) + YRATIO * (XLE(NMI)-XLE(N))                    XLT10400
  100 XTRAIL(K) = XTR(N) + YRATIO * (XTR(NMI)-XTR(N))                   XLT10410
      GO TO 60                                                          XLT10420
  110 CONTINUE                                                          XLT10430
      IR = 1                                                            XLT10440
      RETURN                                                            XLT10450
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND RETURN.                  XLT10460
  120 WRITE(6, 130 )                                                    XLT10470
  130 FORMAT(1H1///20X,38HAN INCONSISTENCY HAS BEEN FOUND IN THE,       XLT10480
     1       42H SECTIONAL LEADING AND TRAILING EDGE INPUT)             XLT10490
      IR = 2                                                            XLT10500
      RETURN                                                            XLT10510
      END                                                               XLT10520
```

```
*DECK XLETR2                                                            XLT20010
      SUBROUTINE XLETR2                                                 XLT20020
C                                                                       XLT20030
C  THIS SUBROUTINE READS THE FUNDAMENTAL PLANFORM PARAMETERS FOR A      XLT20040
C  TRAPEZOIDAL WING, AND CALCULATES THE LEADING AND TRAILING EDGE       XLT20050
C  COORDINATES AT EACH Y STATION.  NOTE THAT THE PLANFORM OUTLINE       XLT20060
C  MUST BE SYMETRIC.                                                    XLT20070
C                                                                       XLT20080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) XLT20090
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     XLT20100
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  XLT20110
```

```
     1             D(40),KK(400),ITYPE(400)                             XLT20120
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             XLT20130
C                                                                       XLT20140
C  READ THE FUNDAMENTAL PLANFORM PARAMETERS                              XLT20150
      READ(5, 20 ) ARATIO,SWEEP,TR                                      XLT20160
   20 FORMAT(4F10.6)                                                    XLT20170
C                                                                       XLT20180
C  COMPUTE THE GENERAL PLANFORM CHARACTERISTICS                         XLT20190
      B2 = SPAN / 2.00                                                  XLT20200
      SW = SWEEP / 57.295779                                            XLT20210
   30 CROOT = 2.0 * SPAN / ((1.0+TR)*ARATIO)                            XLT20220
      AREA = (1.0+TR) * CROOT * B2                                      XLT20230
      XLB2 = 0.250 * (1.0-TR) * CROOT + B2 * TAN(SW)                    XLT20240
   40 CMAC = 2.0 * CROOT * (1.0 + TR + TR*TR) / (3.0*(1.0+TR))          XLT20250
      IF(CREF .EQ. 0.0) CREF = CMAC                                     XLT20260
      CBAR=AREA/SPAN                                                    XLT20270
C                                                                       XLT20280
C  COMPUTE THE LEADING AND TRAILING EDGE COORDINATES                    XLT20290
   50 DO 70 K = 1,NROWS                                                 XLT20300
      YBAR = Y(K)                                                       XLT20310
      IF(YBAR .LT. 0.0) YBAR = -YBAR                                    XLT20320
      XLEAD(K) = XLB2 * YBAR                                            XLT20330
   60 C = CROOT * (1.0-(1.0-TR)*YBAR)                                   XLT20340
      XTRAIL(K) = XLEAD(K) + C                                          XLT20350
   70 CONTINUE                                                          XLT20360
      RETURN                                                            XLT20370
      END                                                               XLT20380
```

```
*DECK NORM1                                                             NORM0010
      SUBROUTINE NORM1                                                  NORM0020
C                                                                       NORM0030
C  THIS SUBROUTINE NORMALIZES ALL WING PLANFORM GEOMETRY BY SPAN/2      NORM0040
C                                                                       NORM0050
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) NORM0060
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     NORM0070
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  NORM0080
     1             D(40),KK(400),ITYPE(400)                             NORM0090
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             NORM0100
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          NORM0110
C                                                                       NORM0120
   20 B2 = SPAN / 2.00                                                  NORM0130
      AREA = AREA / B2**2                                               NORM0140
      CREF = CREF / B2                                                  NORM0150
   30 XMC = XMC / B2                                                    NORM0160
      XCG = XCG / B2                                                    NORM0170
      DO 50 K = 1,NROWS                                                 NORM0180
   40 XLEAD(K) = XLEAD(K) / B2                                          NORM0190
      XTRAIL(K) = XTRAIL(K) / B2                                        NORM0200
   50 CONTINUE                                                          NORM0210
      DO 60 I = 1,10                                                    NORM0220
      GHITE(I) = GHITE(I) / B2                                          NORM0230
   60 CONTINUE                                                          NORM0240
      SPAN = 2.00                                                       NORM0250
      ARATIO = SPAN * SPAN / AREA                                       NORM0260
      RETURN                                                            NORM0270
      END                                                               NORM0280
```

```
      SUBROUTINE BOXS(IR)                                               BOXS0020
C                                                                       BOXS0030
C  THIS SUBROUTINE CONSTRUCTS THE GEOMETRIC PARAMETERS FOR ALL THE      BOXS0040
C  EVD ELEMENTS ON THE WING AND JET                                     BOXS0050
C                                                                       BOXS0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            BOXS0070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BOXS0080
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     BOXS0090
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  BOXS0100
     1             D(40),KK(400),ITYPE(400)                             BOXS0110
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             BOXS0120
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),           BOXS0130
     1           NWTYPE,NJTYPE                                          BOXS0140
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         BOXS0150
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               BOXS0160
     2             XTR(40),XTL(40),CHR(40),CHL(40)                      BOXS0170
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HX,HH,VL,VH,     BOXS0180
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    BOXS0190
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES              BOXS0200
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XF(41),ZCL(41),    BOXS0210
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),               BOXS0220
     2              V(40,7),W(40,7)                                     BOXS0235
C                                                                       BOXS0240
C  CONSTRUCT THE ELEMENTS ON THE WING                                   BOXS0250
C                                                                       BOXS0260
C  COMPUTE SECTIONAL DATA                                               BOXS0270
   20 CHORD(1) = XTRAIL(1) - XLEAD(1)                                   BOXS0280
      DELTA(1) = 1.00 - Y(1)                                            BOXS0290
      CMAC = CHORD(1)**2 * DELTA(1)                                     BOXS0300
      DO 40 K = 2,NROWS                                                 BOXS0310
   30 CHORD(K) = XTRAIL(K) - XLEAD(K)                                   BOXS0320
      DELTA(K) = Y(K-1) - Y(K) - DELTA(K-1)                             BOXS0330
      IF(DELTA(K) .LT. 0.0) GO TO 200                                   BOXS0340
      CMAC = CMAC + CHORD(K)**2 * DELTA(K)                              BOXS0350
   40 CONTINUE                                                          BOXS0360
C  CHECK THE VALIDITY OF THE SECTIONAL ALLIGNMENT                       BOXS0361
      YD = Y(NROWS) - DELTA(NROWS)                                      BOXS0362
      IF((ISYMM .GE. 0) .AND. (ABS(YD) .GT. 0.0001)) GO TO 190          BOXS0363
      IF((ISYMM .EQ. 1) .AND. (ABS(YD+1.0) .GT. 0.0001)) GO TO 190      BOXS0364
      DSUM = DELTA(1)                                                   BOXS0365
      DO 35 K = 2,NROWS                                                 BOXS0366
      YL = Y(K) + DELTA(K)                                              BOXS0367
      YR = Y(K-1) - DELTA(K-1)                                          BOXS0368
      IF(ABS(YR-YL) .GT. 0.0001) GO TO 190                              BOXS0369
      DSUM = DSUM + DELTA(K)                                            BOXS0370
   35 CONTINUE                                                          BOXS0371
      IF(ABS(DSUM-0.50) .GT. 0.0001) GO TO 190                          BOXS0372
      CMAC = 2.0 * CMAC / AREA                                          BOXS0375
      IF(ISYMM .LT. 1) CMAC = 2.0 * CMAC                                BOXS0380
      IF(CREF .LT. 0.0001) CREF = CMAC                                  BOXS0390
      CALL TANS(TANLE,XLEAD,Y,NROWS)                                    BOXS0400
      IF((NPOINT .GT. 0) .OR. (NROWH .GT. 0) .OR. (NROWV .GT. 0) .OR.   BOXS0410
     1   (NSEG .GT. 0))CALL TANS(TANTE,XTRAIL,Y,NROWS)                  BOXS0420
C                                                                       BOXS0430
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION            BOXS0440
      I = 0                                                             BOXS0450
      DO 100 K = 1,NROWS                                                BOXS0460
C  COMPUTE X-COORDINATES                                                BOXS0470
      NWK = NW(K)                                                       BOXS0480
      DO 60 L = 1,NWK                                                   BOXS0490
```

```
      I = I + 1                                                         BOXS0500
      ICK = ICTYPE(K)                                                   BOXS0510
   50 XB(I) = XBW(L,ICK)                                                BOXS0520
   60 CONTINUE                                                          BOXS0530
C  COMPUTE ALL OTHER PARAMETERS                                         BOXS0540
      I = I - NWK                                                       BOXS0550
      IW(K) = I + 1                                                     BOXS0560
      DO 90 L = 1,NWK                                                   BOXS0570
      I = I + 1                                                         BOXS0580
   70 KK(I) = K                                                         BOXS0590
      DEL(I) = XB(I+1) - XB(I)                                          BOXS0600
   80 XI(I) = XLEAD(K) + XB(I) * CHORD(K)                               BOXS0610
      ITYPE(I) = 10                                                     BOXS0620
   90 CONTINUE                                                          BOXS0630
C  REDEFINE THE LAST DEL IN THIS SECTION, AND DEFINE THE L.E. EVO TYPE  BOXS0640
      DEL(I) = 1.00 - XB(I)                                             BOXS0650
      IWK = IW(K)                                                       BOXS0660
      ITYPE(IWK) = 20                                                   BOXS0670
  100 CONTINUE                                                          BOXS0680
      NWT = I                                                           BOXS0690
C                                                                       BOXS0700
C  CONSTRUCT THE ELEMENTS ON THE JET SHEET                              BOXS0710
C                                                                       BOXS0720
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION            BOXS0730
      IF(JETFLG .NE. 0) GO TO 190                                       BOXS0740
      DO 180 K = 1,NROWS                                                BOXS0750
C  COMPUTE X-COORDINATES                                                BOXS0760
      IJ(K) = 0                                                         BOXS0770
  110 NJK = NJ(K)                                                       BOXS0780
      IF(NJK .EQ. 0) GO TO 180                                          BOXS0790
      DO 130 L = 1,NJK                                                  BOXS0800
      I = I + 1                                                         BOXS0810
      IJK = IJTYPE(K)                                                   BOXS0820
  120 XB(I) = XBJ(L,IJK)                                                BOXS0830
  130 CONTINUE                                                          BOXS0840
C  COMPUTE ALL OTHER PARAMETERS                                         BOXS0850
      I = I - NJK                                                       BOXS0860
  140 IJ(K) = I + 1                                                     BOXS0870
      DO 170 L = 1,NJK                                                  BOXS0880
      I = I + 1                                                         BOXS0890
  150 KK(I) = K                                                         BOXS0900
      DEL(I) = XB(I+1) - XB(I)                                          BOXS0910
  160 XI(I) = XLEAD(K) + XB(I) * CHORD(K)                               BOXS0920
      ITYPE(I) = 10                                                     BOXS0930
  170 CONTINUE                                                          BOXS0940
C  REDEFINE THE LAST DEL AND EVD TYPE AND THE D VALUE FOR THIS SECTION  BOXS0950
      DEL(I) = 1.0E10                                                   BOXS0960
      ITYPE(I) = 30                                                     BOXS0970
      D(K) = XI(I) - XTRAIL(K)                                          BOXS0980
  180 CONTINUE                                                          BOXS0990
  190 NMAX = I                                                          BOXS1000
      IF(NMAX .GT. 400) GO TO 220                                       BOXS1010
      NJT = NMAX - NWT                                                  BOXS1020
      IR = 1                                                            BOXS1030
      RETURN                                                            BOXS1040
C                                                                       BOXS1050
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.                    BOXS1060
  200 WRITE(6, 210 )                                                    BOXS1070
  210 FORMAT(1H1/38X,44HPLEASE CHECK YOUR SECTION LOCATION (Y) INPUT)   BOXS1080
      IR = 2                                                            BOXS1090
      RETURN                                                            BOXS1100
  220 WRITE(6, 230 ) NMAX                                               BOXS1110
  230 FORMAT(1H1/48X,I4,21H IS TOO MANY ELEMENTS)                       BOXS1120
      IR = 2                                                            BOXS1130
      RETURN                                                            BOXS1140
      END                                                               BOXS1150
```

```
*DECK INPUTJ                                                            INPJ0010
      SUBROUTINE INPUTJ(IR)                                             INPJ0020
C                                                                       INPJ0030
C  THIS SUBROUTINE READS THE JET GEOMETRY INPUT                          INPJ0040
C  FOR THE SECTIONAL GEOMETRY METHOD                                    INPJ0050
C                                                                       INPJ0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            INPJ0070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) INPJ0080
      COMMON/SG1/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),           INPJ0090
     1                 NWTYPE,NJTYPE                                    INPJ0100
      DIMENSION NI(10)                                                  INPJ0110
C                                                                       INPJ0120
C  READ THE TYPE OF DIVISION FOR EACH ROW                               INPJ0130
   20 NJTYPE = 0                                                        INPJ0140
      NROWSJ = 0                                                        INPJ0150
      IF(JETFLG .NE. 0) GO TO 120                                       INPJ0160
      READ(5, 30 ) (IJTYPE(K),K=1,NROWS)                                INPJ0170
   30 FORMAT(40I2)                                                      INPJ0180
      DO 40 K = 1,NROWS                                                 INPJ0190
      IF(IJTYPE(K) .GT. NJTYPE) NJTYPE = IJTYPE(K)                      INPJ0200
      IF(IJTYPE(K) .NE. 0) NROWSJ = NROWSJ + 1                          INPJ0210
   40 CONTINUE                                                          INPJ0220
      IF(NJTYPE .GT. 10) GO TO 140                                      INPJ0230
C  READ THE NUMBER OF CHORDWISE DIVISIONS IN EACH ROW TYPE              INPJ0240
      READ(5, 50 ) (NI(N),N=1,NJTYPE)                                   INPJ0250
   50 FORMAT(10I2)                                                      INPJ0260
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE                   INPJ0270
      DO 70 N = 1,NJTYPE                                                INPJ0280
      NIN = NI(N)                                                       INPJ0290
      IF((NIN .LT. 1) .OR. (NIN .GT. 20)) GO TO 160                     INPJ0300
      READ(5, 60 ) (XBJ(L,N),L=1,NIN)                                   INPJ0310
   60 FORMAT(8F10.6)                                                    INPJ0320
   70 CONTINUE                                                          INPJ0330
C                                                                       INPJ0340
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW                INPJ0350
      DO 90  K = 1,NROWS                                                INPJ0360
      NJ(K) = 0                                                         INPJ0370
      IF(IJTYPE(K) .EQ. 0) GO TO 90                                     INPJ0380
      IJK = IJTYPE(K)                                                   INPJ0390
   80 NJ(K) = NI(IJK)                                                   INPJ0400
   90 CONTINUE                                                          INPJ0410
C  CHECK FOR ROW CONSISTENCY ON EITHER SIDE OF JET                      INPJ0420
      ICOUNT = 1                                                        INPJ0430
      IF(NJ(1) .EQ. 0)ITEST = 0                                         INPJ0440
      IF(NJ(1) .GT. 0)ITEST = 1                                         INPJ0450
      DO 110 K = 2,NROWS                                                INPJ0460
      IF(NJ(K) .EQ. 0)ICOMP = 0                                         INPJ0470
      IF(NJ(K) .GT. 0)ICOMP = 1                                         INPJ0480
      IF(ICOMP .EQ. ITEST)GO TO 100                                     INPJ0490
      IF(ICOUNT .LT. 3)GO TO 180                                        INPJ0500
      ICOUNT = 1                                                        INPJ0510
      IF(NJ(K) .EQ. 0)ITEST = 0                                         INPJ0520
      IF(NJ(K) .GT. 0)ITEST = 1                                         INPJ0530
      GO TO 110                                                         INPJ0540
  100 ICOUNT = ICOUNT + 1                                               INPJ0550
  110 CONTINUE                                                          INPJ0560
      IF(ICOUNT .LT. 3) GO TO 180                                       INPJ0570
      IR = 1                                                            INPJ0580
      RETURN                                                            INPJ0590
C                                                                       INPJ0600
C  THERE IS NO JET FOR THIS RUN                                         INPJ0610
  120 DO 130 K = 1,NROWS                                                INPJ0620
      IJTYPE(K) = 0                                                     INPJ0630
      NJ(K) = 0                                                         INPJ0640
  130 CONTINUE                                                          INPJ0650
      IR = 1                                                            INPJ0660
      RETURN                                                            INPJ0670
C                                                                       INPJ0680
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.                    INPJ0690
  140 WRITE(6, 150 ) NJTYPE                                             INPJ0700
  150 FORMAT(1H1/25HNUMBER OF JET ROW TYPES =,I3)                       INPJ0710
      IR = 2                                                            INPJ0720
      RETURN                                                            INPJ0730
  160 WRITE(6, 170 ) NIN,N                                              INPJ0740
  170 FORMAT(1H1/38X,I3,37H JET ELEMENTS PRESCRIBED FOR ROW TYPE,I3)    INPJ0750
      IR = 2                                                            INPJ0760
      RETURN                                                            INPJ0770
  180 WRITE(6, 190 )                                                    INPJ0780
  190 FORMAT(1H1,29H3 ROW CONTINUITY RULE FAILURE)                      INPJ0790
      IR = 2                                                            INPJ0800
      RETURN                                                            INPJ0810
      END                                                               INPJ0820
```

```
*DECK BOXJ                                                              BOXJ0010
      SUBROUTINE BOXJ(NEWMAX,IR)                                        BOXJ0020
C                                                                       BOXJ0030
C  THIS SUBROUTINE COMPUTES THE JET BLOWING FACTOR CMUP                 BOXJ0040
C                                                                       BOXJ0050
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BOXJ0060
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  BOXJ0070
     1                 D(40),KK(400),ITYPE(400)                         BOXJ0080
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           BOXJ0090
C                                                                       BOXJ0100
C  COMPUTE THE NEW CMUP AND SAVE THE OLD VALUES AS CMUPP                BOXJ0110
   20 NEWMAX = NMAX                                                     BOXJ0120
      ICOUNT = 0                                                        BOXJ0130
      DO 40 K = 1,NROWS                                                 BOXJ0140
      CMUPP(K) = CMUP(K)                                                BOXJ0150
      IF(NJ(K) .EQ. 0) GO TO 40                                         BOXJ0160
      IF(CMU(K) .LT. 0.0001) GO TO 30                                   BOXJ0170
      CMUP(K) = 2.00 / (CHORD(K)*CMU(K))                                BOXJ0180
      GO TO 40                                                          BOXJ0190
   30 ICOUNT = ICOUNT + 1                                               BOXJ0200
      CMUP(K) = 0.00                                                    BOXJ0210
   40 CONTINUE                                                          BOXJ0220
      IF(ICOUNT .EQ. 0) GO TO 50                                        BOXJ0230
      IF(ICOUNT .LT. NROWSJ) GO TO 60                                   BOXJ0240
      NEWMAX = NWT                                                      BOXJ0250
   50 IR = 1                                                            BOXJ0260
      RETURN                                                            BOXJ0270
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND STOP.                    BOXJ0280
   60 WRITE(6, 70 )                                                     BOXJ0290
   70 FORMAT(1H0   43X,35HA ZERO VALUE OF CMU HAS BEEN INPUT.)          BOXJ0300
      IR = 2                                                            BOXJ0310
      RETURN                                                            BOXJ0320
      END                                                               BOXJ0330
```

```
*DECK TANS                                                              TANS0010
      SUBROUTINE TANS(TAN,X,Y,NROWS)                                    TANS0020
C                                                                       TANS0030
C  THIS SUBROUTINE COMPUTES THE TANGENT OF THE LEADING OR TRAILING EDGE TANS0040
C  SWEEP ANGLE AT THE CENTERLINE OF EACH SECTION.  IT IS ACCURATE FOR   TANS0050
C  SECTIONS WITH STRAIGHT EDGES IN GROUPS OF THREE OR MORE.             TANS0060
C  IT IS ONLY APPROXIMATE FOR CURVED EDGES.                             TANS0070
C  IT MAY RESULT IN ERRORS FOR SECTIONS ADJACENT TO WING BREAKS,        TANS0080
C  IF STRAIGHT EDGES ARE IN ADJACENT GROUPS OF ONLY ONE OR TWO.         TANS0090
C                                                                       TANS0100
      DIMENSION TAN(40),X(40),Y(40),S(40)                               TANS0110
      SLOP(XR,XL,YR,YL) = (XR-XL) / (YR-YL)                             TANS0120
C                                                                       TANS0130
      DO 30 K = 1,NROWS                                                 TANS0140
      KR = K-1                                                          TANS0150
      KL = K                                                            TANS0160
      IF(K .GT. 1) GO TO 20                                             TANS0170
      KR = 1                                                            TANS0180
      KL = 2                                                            TANS0190
   20 S(K) = SLOP(X(KR),X(KL),Y(KR),Y(KL))                              TANS0200
   30 CONTINUE                                                          TANS0210
      DO 60  K = 1,NROWS                                                TANS0220
      IF(K .LT. 3) GO TO 40                                             TANS0230
      IF(K .EQ. NROWS) GO TO 40                                         TANS0240
      IF(K .EQ. (NROWS-1)) GO TO 50                                     TANS0250
C  CHECK WHETHER THE RIGHT OR LEFT SIDES ARE STRAIGHT                   TANS0260
      IF(ABS(S(K) - S(K-1)) .LT. 0.001) GO TO 40                        TANS0270
      IF(ABS(S(K+1) - S(K+2)) .LT. 0.001) GO TO 50                      TANS0280
C  NEITHER SIDE IS CONCLUSIVELY STRAIGHT - CHECK FURTHER LEFT AND RIGHT TANS0290
      IF(K .EQ. 3) GO TO 50                                             TANS0300
      IF(K .EQ. (NROWS-2)) GO TO 40                                     TANS0310
      IF(ABS(S(K-1) - S(K-2)) .LT. 0.001) GO TO 50                      TANS0320
      IF(ABS(S(K+2) - S(K+3)) .LT. 0.001) GO TO 40                      TANS0330
C  THE TRUE SHAPE CANNOT BE DETERMINED - GIVE UP AND TAKE THE AVERAGE   TANS0340
      TAN(K) = (S(K) + S(K+1)) / 2.00                                   TANS0350
      GO TO 60                                                          TANS0360
C  THE RIGHT EDGE IS STRAIGHT                                           TANS0370
   40 TAN(K) = S(K)                                                     TANS0380
      GO TO 60                                                          TANS0390
C  THE LEFT EDGE IS STRAIGHT                                            TANS0400
   50 TAN(K) = S(K+1)                                                   TANS0410
   60 CONTINUE                                                          TANS0420
      RETURN                                                            TANS0430
      END                                                               TANS0440
```

```
*DECK INCASE                                                            INCS0010
```

```
      SUBROUTINE INCASE(LCASE,NOALFA)                                   INCS0020
C                                                                       INCS0030
C  THIS SUBROUTINE READS THE FUNDAMENTAL GEOMETRIC CASE DATA            INCS0040
C                                                                       INCS0050
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) INCS0060
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTB                  INCS0070
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     INCS0080
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTINCS0090
      DIMENSION NI(10),DUMMY(40)                                        INCS0100
C                                                                       INCS0110
C  INITIALIZE SECTIONAL DATA                                            INCS0120
      DO 40 K = 1,NROWS                                                 INCS0130
   20 TST(K,LCASE) = 0.00                                               INCS0140
      HL(K,LCASE) = 0.00                                                INCS0150
      DJ(K) = 0 00                                                      INCS0160
      ACTE(K) = 0.00                                                    INCS0170
      ICT(K) = 0                                                        INCS0180
      IHT(K) = 0                                                        INCS0190
C  INITIALIZE THE CAMBER ANGLES                                         INCS0200
      NWK = NW(K)                                                       INCS0210
      DO 30 L = 1,NWK                                                   INCS0220
      AC(L,K) = 0.00                                                    INCS0230
   30 CONTINUE                                                          INCS0240
   40 CONTINUE                                                          INCS0250
C  INITIALIZE THE HINGE DATA                                            INCS0260
      DO 60 N = 1,NROWS                                                 INCS0270
      DO 50 L = 1,4                                                     INCS0280
      XHB(L,N) = 0.00                                                   INCS0290
      BET(L,N) = 0.00                                                   INCS0300
   50 CONTINUE                                                          INCS0310
   60 CONTINUE                                                          INCS0320
C                                                                       INCS0330
      IF((LCASE .EQ. 1) .AND. (NOALFA .GT. 0)) RETURN                   INCS0340
C                                                                       INCS0350
C  READ FUNDAMENTAL CASE CONTROL FLAGS                                  INCS0360
      READ(5, 70 ) INPUTT,INPUTH,INPUTD,INPUTC,INPUTB                   INCS0370
   70 FORMAT(5I2)                                                       INCS0380
C                                                                       INCS0390
C  READ SECTIONAL TWIST, HEIGHT AND JET DEFLECTION DATA                 INCS0400
      IF(INPUTT .NE. 0) READ(5, 80 ) (TST(K,LCASE),K=1,NROWS)           INCS0410
   80 FORMAT(8F10.6)                                                    INCS0420
      IF(INPUTH .NE. 0) READ(5, 80 ) (HL(K,LCASE),K=1,NROWS)            INCS0430
      IF(INPUTD .EQ. 0) GO TO 100                                       INCS0440
      IF(NROWSJ .GT. 0) READ(5, 80 ) (DUMMY(K),K=1,NROWSJ)              INCS0450
C  DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE DJ ARRAY                 INCS0460
      KP = 0                                                            INCS0470
      DO 90 K = 1,NROWS                                                 INCS0480
      IF(NJ(K) .EQ. 0) GO TO 90                                         INCS0490
      KP = KP + 1                                                       INCS0500
      DJ(K) = DUMMY(KP)                                                 INCS0510
   90 CONTINUE                                                          INCS0520
C                                                                       INCS0530
C  READ THE CAMBER ANGLES, IN DEGREES                                   INCS0540
  100 IF(INPUTC .EQ. 0) GO TO 170                                       INCS0550
      READ(5, 110 ) (ICT(K),K=1,NROWS)                                  INCS0560
  110 FORMAT(40I2)                                                      INCS0570
      NCT = 0                                                           INCS0580
      DO 120 K = 1,NROWS                                                INCS0590
      IF(ICT(K) .EQ. 0) GO TO 120                                       INCS0600
      IF(ICT(K) .GT. NCT) NCT = ICT(K)                                  INCS0610
```

```
      ICK = ICT(K)                                                      INCS0620
      NI(ICK) = NW(K)                                                   INCS0630
  120 CONTINUE                                                          INCS0640
      DO 140 N = 1,NCT                                                  INCS0650
      NIN = NI(N)                                                       INCS0660
  130 READ(5, 80 ) (AC(L,N),L=1,NIN)                                    INCS0670
  140 CONTINUE                                                          INCS0680
  150 IF(NROWSJ .GT. 0) READ(5, 80 ) (DUMMY(K),K=1,NROWSJ)              INCS0690
C  DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE ACTE ARRAY               INCS0700
      KP = 0                                                            INCS0710
      DO 160 K = 1,NROWS                                                INCS0720
      IF(NJ(K) .EQ. 0) GO TO 160                                        INCS0730
      KP = KP + 1                                                       INCS0740
      ACTE(K) = DUMMY(KP)                                               INCS0750
  160 CONTINUE                                                          INCS0760
C                                                                       INCS0770
C  READ THE HINGE LOCATION, TYPE AND TURNING ANGLE DATA                 INCS0780
  170 IF(INPUTB .EQ. 0) GO TO 220                                       INCS0790
  180 READ(5, 110 ) (IHT(K),K=1,NROWS)                                  INCS0800
      NHT = 0                                                           INCS0810
      DO 190 K = 1,NROWS                                                INCS0820
      IF(IHT(K) .GT. NHT) NHT = IHT(K)                                  INCS0830
  190 CONTINUE                                                          INCS0840
      DO 210 N = 1,NHT                                                  INCS0850
      READ(5, 200 ) (XHB(L,N),IFS(L,N),BET(L,N),L=1,4)                  INCS0860
  200 FORMAT(4(F10.6,I1,F9.6))                                          INCS0870
  210 CONTINUE                                                          INCS0880
  220 RETURN                                                            INCS0890
      END                                                               INCS0900
```

```
*DECK BEECEE                                                            BECE0010
      SUBROUTINE BEECEE(LCASE,NOALFA,IR,ILE,ITE)                        BECE0020
C                                                                       BECE0030
C  THIS SUBROUTINE CONSTRUCTS THE BOUNDARY CONDITION ARRAYS FOR THE     BECE0040
C  FUNDAMENTAL GEOMETRIC CASES                                          BECE0050
C                                                                       BECE0060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BECE0070
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  BECE0080
     1              D(40),KK(400),ITYPE(400)                            BECE0090
      COMMON/FCASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTB                  BECE0100
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     BECE0110
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTBECE0120
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    BECE0130
      DIMENSION NH(40)                                                  BECE0140
C                                                                       BECE0150
C  INITIALIZE THE BOUNDARY CONDITION ANGLES                             BECE0160
      DO 30 I = 1,NMAX                                                  BECE0170
   20 EPS(I,LCASE) = 0.00                                               BECE0180
      BETA(I,LCASE) = 0.00                                              BECE0190
   30 CONTINUE                                                          BECE0200
      DO 40 K = 1,NROWS                                                 BECE0210
      THETA(K,LCASE) = 0.00                                             BECE0220
      THS(K,LCASE) = 0.00                                               BECE0230
   40 CONTINUE                                                          BECE0240
C                                                                       BECE0250
C  DEFINE THE ANGLES FOR THE ANGLE-OF-ATTACK FUNDAMENTAL CASE           BECE0260
      IF((LCASE .GT. 1) .OR. (NOALFA .EQ. 0)) GO TO 70                  BECE0270
```

```
      DO 50 I = 1,NWT                                                   BECE0280
      EPS(I,1) = 1.00                                                   BECE0290
   50 CONTINUE                                                          BECE0300
      DO 60 K = 1,NROWS                                                 BECE0310
      IF(NJ(K) .GT. 0) THETA(K,1) = 1.000                               BECE0320
   60 CONTINUE                                                          BECE0330
      IR = 1                                                            BECE0340
      RETURN                                                            BECE0350
C                                                                       BECE0360
C  DEFINE THE ANGLES FOR ALL REMAINING FUNDAMENTAL CASES                BECE0370
C                                                                       BECE0380
C  CAMBER CONTRIBUTION                                                  BECE0390
   70 IF(INPUTC .EQ. 0) GO TO 120                                       BECE0400
      I = 0                                                             BECE0410
      DO 110 K = 1,NROWS                                                BECE0420
      IF(NJ(K) .EQ. 0) GO TO 80                                         BECE0430
      THETA(K,LCASE) = ACTE(K)                                          BECE0440
   80 NWK = NW(K)                                                       BECE0450
      IF(ICT(K) .EQ. 0) GO TO 100                                       BECE0460
      DO 90 L = 1,NWK                                                   BECE0470
      I = I + 1                                                         BECE0480
      ICK = ICT(K)                                                      BECE0490
      EPS(I,LCASE) = AC(L,ICK)                                          BECE0500
   90 CONTINUE                                                          BECE0510
      GO TO 110                                                         BECE0520
  100 I = I + NWK                                                       BECE0530
  110 CONTINUE                                                          BECE0540
C                                                                       BECE0550
C  TWIST CONTRIBUTION                                                   BECE0560
  120 IF(INPUTT .EQ. 0) GO TO 170                                       BECE0570
      I = 0                                                             BECE0580
      DO 160 K = 1,NROWS                                                BECE0590
      IF(NJ(K) .EQ. 0) GO TO 130                                        BECE0600
      THETA(K,LCASE) = THETA(K,LCASE) + TST(K,LCASE)                    BECE0610
  130 NWK = NW(K)                                                       BECE0620
      DO 150 L = 1,NWK                                                  BECE0630
      I = I + 1                                                         BECE0640
  140 EPS(I,LCASE) = EPS(I,LCASE) + TST(K,LCASE)                        BECE0650
  150 CONTINUE                                                          BECE0660
  160 CONTINUE                                                          BECE0670
C                                                                       BECE0680
C  FLAP AND SLAT DEFLECTION CONTRIBUTION                                BECE0690
C                                                                       BECE0700
  170 IF(INPUTB .EQ. 0) GO TO 330                                       BECE0710
C                                                                       BECE0720
C  SUM UP THE TOTAL SLAT ANGLE AND FIND THE NUMBER OF HINGES ON EACH ROWBECE0730
      DO 200 K = 1,NROWS                                                BECE0740
      NH(K) = 0                                                         BECE0750
      IF(IHT(K) .EQ. 0) GO TO 200                                       BECE0760
      DO 190 L = 1,4                                                    BECE0770
  180 N = IHT(K)                                                        BECE0780
      IF(XHB(L,N) .LT. 0.001) GO TO 190                                 BECE0790
      NH(K) = NH(K) + 1                                                 BECE0800
      IF(IFS(L,N) .GT. 0) THS(K,LCASE) = THS(K,LCASE) + BET(L,N)        BECE0810
  190 CONTINUE                                                          BECE0820
  200 CONTINUE                                                          BECE0830
C                                                                       BECE0840
C  COMPUTE INCIDENCE OF EACH ELEMENT AND FIND TURNING ANGLE AND EVU TYPEBECE0850
C  FOR EACH HINGE ELEMENT                                               BECE0860
      I = 0                                                             BECE0870
```

```
      DO 320 K = 1,NROWS                                                BECE0880
      NWK = NW(K)                                                       BECE0890
      N = IHT(K)                                                        BECE0900
  210 IF(N .EQ. 0) GO TO 310                                            BECE0910
      LSTART = 1                                                        BECE0920
      B = 0.0                                                           BECE0930
      NHK = NH(K)                                                       BECE0940
  220 IF(NHK .EQ. 0) GO TO 370                                          BECE0950
C  CYCLE THE HINGE POINTS IN CHORDWISE ORDER                            BECE0960
      DO 280 LH = 1,NHK                                                 BECE0970
C  CYCLE THE VORTEX POINTS IN CHORDWISE ORDER, LOOKING FOR NEXT HINGE   BECE0980
      DO 260 L = LSTART,NWK                                             BECE0990
      I = I + 1                                                         BECE1000
C  CHECK ON RELATIVE LOCATION OF VORTEX POINT AND HINGE POINT           BECE1010
  230 XDIFF = XHB(LH,N) - XB(I)                                         BECE1020
      IF(ABS(XDIFF) .LT. 0.001) GO TO 240                               BECE1030
      IF(XDIFF .GT. 0.001) GO TO 250                                    BECE1040
      IF(XDIFF .LT. -0.001) GO TO 370                                   BECE1050
C  THE ITH VORTEX POINT IS A HINGE POINT                                BECE1060
  240 B = B + BET(LH,N)                                                 BECE1070
      BETA(I,LCASE) = BET(LH,N)                                         BECE1080
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B                    BECE1090
      ITYPE(I) = 42                                                     BECE1100
      IF(IFS(LH,N) .GT. 0) ITYPE(I) = 41                                BECE1110
      ILE = 0                                                           BECE1111
      ITE = 0                                                           BECE1112
      IF(ITYPE(I).EQ.41)ILE=1                                           BECE1120
      IF(ITYPE(I).EQ.42)ITE=1                                           BECE1130
      GO TO 270                                                         BECE1140
C  THE ITH VORTEX POINT IS NOT A HINGE POINT                            BECE1150
  250 EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B                    BECE1160
  260 CONTINUE                                                          BECE1170
  270 LSTART = I - IW(K) + 1                                            BECE1180
  280 CONTINUE                                                          BECE1190
C  DEFINE THE INCIDENCE ANGLE FOR REMAINING POINTS BEHIND THE LAST HINGEBECE1200
      IF(LSTART .EQ. NWK) GO TO 300                                     BECE1210
      LSTART = LSTART + 1                                               BECE1220
      DO 290 L = LSTART,NWK                                             BECE1230
      I = I + 1                                                         BECE1240
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B                    BECE1250
  290 CONTINUE                                                          BECE1260
C  COMPUTE THE EFFECT OF THE HINGES ON THE JET ANGLE                    BECE1270
  300 IF(NJ(K) .GT.  0) THETA(K,LCASE) = THETA(K,LCASE)-THS(K,LCASE)+B  BECE1280
      GO TO 320                                                         BECE1290
  310 I = I + NWK                                                       BECE1300
  320 CONTINUE                                                          BECE1310
C                                                                       BECE1320
C  JET DEFLECTION CONTRIBUTION                                          BECE1330
C                                                                       BECE1340
  330 IF(INPUTD .EQ. 0) GO TO 360                                       BECE1350
      DO 350 K = 1,NROWS                                                BECE1360
      IF(NJ(K) .EQ. 0) GO TO 350                                        BECE1370
      I = IJ(K)                                                         BECE1380
  340 BETA(I,LCASE) = DJ(K)                                             BECE1390
      IF(ABS(DJ(K)) .GT. 0.0001) ITYPE(I) = 43                          BECE1400
      THETA(K,LCASE) = THETA(K,LCASE) + DJ(K)                           BECE1410
  350 CONTINUE                                                          BECE1420
  360 IR = 1                                                            BECE1430
      RETURN                                                            BECE1440
C                                                                       BECE1450
```

```
C   AN ERROR HAS OCCURED.    PRINT A MESSAGE AND QUIT.                     BECE1460
C                                                                          BECE1470
  370 WRITE(6, 380 ) LCASE,K,N                                             BECE1480
  380 FORMAT(1H1///45X,26HFUNDAMENTAL GEOMETRIC CASE,I3/                   BECE1490
     1        18X,50HAN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT,   BECE1500
     2        18H DATA FOR WING ROW,I3,10H, ROW TYPE,I3)                   BECE1510
      IR = 2                                                               BECE1520
      RETURN                                                               BECE1530
      END                                                                  BECE1540
```

```
*DECK OUT1                                                                 OUT10010
      SUBROUTINE OUT1                                                      OUT10020
C                                                                          OUT10030
C  THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE          OUT10040
C  SECTIONAL METHOD INPUT                                                  OUT10050
C                                                                          OUT10060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE               OUT10070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)    OUT10080
      COMMON/LUKE/TITLE(8)                                                 OUT10090
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG        OUT10100
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                   OUT10110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),     OUT10120
     1             D(40),KK(400),ITYPE(400)                                OUT10130
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)                OUT10140
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),        OUT10150
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTOUT10160
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)       OUT10170
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI,IGROUT10180
      COMMON/GEFECT/GHITE(10),IGRND,IGND,HHITE                             OUT10190
      COMMON/FFLO3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),            OUT10200
     1             IFFLD,TCH,AINF(40,10),XLR(40),XLL(40),                  OUT10210
     2             XTR(40),XTL(40),CHR(40),CHL(40)                         OUT10220
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,H ,HH,VL,VH,        OUT10230
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,       OUT10240
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES                 OUT10250
      COMMON/CASES/INT(10),INH(10),IND(10),INC(10),INB(10),IHLE(10),       OUT10260
     1             IHTE(10)                                                OUT10270
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XF(41),ZCL(41),       OUT10280
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                  OUT10290
     2              V(40,7),W(40,7)                                        OUT10300
      DIMENSION FUND(2,6),HEAD(2,6)                                        OUT10310
      DATA BLANK/10H          /                                            OUT10320
C  STORE ALL OF THE FUNDAMENTAL CASE TITLES                                OUT10330
      DATA HEAD/10H     WING ,10HTWIST     ,10H     WING ,10HCAMBER     ,OUT10340
     1          10H     L.E. ,10HHEIGHT    ,10H     JET T,10HURNING     ,OUT10350
     2          10H     L.E. ,10HHINGE     ,10H     T.E. ,10HHINGE      /OUT10360
C                                                                          OUT10370
C  PRINT TITLE AND GENERAL GEOMETRIC PARAMETERS                            OUT10380
   20 WRITE(6, 30 )TITLE                                                   OUT10390
   30 FORMAT(1H1,38X,41(1H*)/                                              OUT10400
     1  39X,41H*  STOL  AERODYNAMIC  METHODS  PROGRAM  */                  OUT10410
     2   39X,41(1H*)///34X,8A10)                                           OUT10420
      CMA = CMAC * SPA / 2.0                                               OUT10430
   40 WRITE(6, 50 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,     OUT10440
     1              ARATIO,XCG,XC                                          OUT10450
   50 FORMAT(1H0//54X,4HUSED,11X,5HINPUT /                                 OUT10460
     1        41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /                  OUT10470
     2        41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /                   OUT10480
     3        41X,6HCMAC =,2F15.6 / 39X,8HARATIO =,2F15.6 /                OUT10490
     4        42X,5HXCG =,2F15.6)                                          OUT10500
      IDERIV=0                                                             OUT10510
      IF(LOGIC.EQ.3)IDERIV=1                                               OUT10520
      WRITE(6, 60 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,OUT10530
     1              IGTYPE,IGT,IHINGE,IHI,IDERIV,IDERIV,IGRND,IGR,IFFLD,OUT10540
     2              IFFLD,NPOINT,NPOINT,NROWH,NROWH,ICAMH,ICAMH,NROWV,   OUT10550
     3              NROWV,ICAMV,ICAMV,NSEG,NSEG,ISECT,ISECT,ICAMF,ICAMF,OUT10560
     4              NWT,NJT,NMAX                                           OUT10570
   60 FORMAT(1H0/ 47X,8HNROWS  =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /    OUT10580
     1              47X,8HISYMM  =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /    OUT10590
     2              47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /    OUT10600
     3              47X,8HIHINGE =,I3,7X,I3 / 47X,8HIDERIV =,I3,7X,I3 /    OUT10610
     4              47X,8HIGRND  =,I3,7X,I3 / 47X,8HIFFLD  =,I3,7X,I3 /    OUT10620
     5              47X,8HNPOINT =,I3,7X,I3 / 47X,8HNROWH  =,I3,7X,I3 /    OUT10630
     6              47X,8HICAMH  =,I3,7X,I3 / 47X,8HNROWV  =,I3,7X,I3 /    OUT10640
     7              47X,8HICAMV  =,I3,7X,I3 / 47X,8HNSEG   =,I3,7X,I3 /    OUT10650
     8              47X,8HISECT  =,I3,7X,I3 / 47X,8HICAMF  =,I3,7X,I3 /// OUT10660
     9              47X,26HNUMBER OF WING ELEMENTS  =,I4 /                 OUT10670
     *              47X,26HNUMBER OF JET ELEMENTS   =,I4 /                 OUT10680
     1              47X,26HTOTAL NUMBER OF ELEMENTS =,I4)                  OUT10690
      IF(LOGIC .EQ. 4) RETURN                                              OUT10700
C  PRINT FUNDAMENTAL CASE HEADER                                           OUT10710
      WRITE(6, 70 )                                                        OUT10720
   70 FORMAT(1H1,28X,74(1H*)/                                              OUT10730
     1    29X,53H*  ELEMENT GEOMETRY AND FUNDAMENTAL CASE DATA FOR ALL,    OUT10740
     2    21H FUNDAMENTAL CASES  *,/,29X,74(1H*))                          OUT10750
C  PRINT OUT DESCRIPTIONS OF EACH FUNDAMENTAL CASE                         OUT10760
C  PRINT OUT DESCRIPTION OF FUNDAMENTAL CASE 1                             OUT10770
      WRITE(6, 80 )                                                        OUT10780
   80 FORMAT(1H0,/,29X,34HFUNDAMENTAL CASE  1 =    ALPHA CASE)             OUT10790
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT10800
   90 FORMAT(1H ,28X,66H(MODIFIED BY INDUCED CAMBER EFFECT DUE TO YAWINGOUT10810
     1 ABOUT THE ORIGIN),/)                                                OUT10820
      IF(LOGIC.NE.3)WRITE(6, 100 )                                         OUT10830
  100 FORMAT(1H0)                                                          OUT10840
      NCM1=NCASES-1                                                        OUT10850
      IF(NCASES.EQ.1)GO TO 410                                             OUT10860
      IF(NCASES.EQ.2.AND.LOGIC.EQ.3)GO TO 410                              OUT10870
C  PRINT OUT DESCRIPTION OF EACH ADDITIONAL FUNDAMENTAL CASE               OUT10880
C  FILL UP ARRAY FOR EACH FUNDAMENTAL CASE AND PRINT OUT                   OUT10890
      IF(LOGIC.EQ.1)NFC=NCASES                                             OUT10900
      IF(LOGIC.EQ.3)NFC=NCM1                                               OUT10910
      DO 400 N=2,NFC                                                       OUT10920
C  FILL UP A BLANK ARRAY FOR FUNDAMENTAL CASES                             OUT10930
      DO 110 I = 1,2                                                       OUT10940
      DO 110 J=1,6                                                         OUT10950
  110 FUND(I,J)=BLANK                                                      OUT10960
      M=1                                                                  OUT10970
      K=0                                                                  OUT10980
  120 GO TO( 130 , 150 , 170 , 190 , 210 , 230 , 270 ),M                   OUT10990
  130 IF(INT(N).GT.0)GO TO 140                                             OUT11000
      M=M+1                                                                OUT11010
      GO TO 150                                                            OUT11020
  140 K=K+1                                                                OUT11030
      IK=1                                                                 OUT11040
      GO TO 250                                                            OUT11050
  150 IF(INC(N).GT.0)GO TO 160                                             OUT11060
      M=M+1                                                                OUT11070
      GO TO 170                                                            OUT11080
  160 K=K+1                                                                OUT11090
      IK=2                                                                 OUT11100
      GO TO 250                                                            OUT11110
  170 IF(INH(N).GT.0)GO TO 180                                             OUT11120
      M=M+1                                                                OUT11130
      GO TO 190                                                            OUT11140
  180 K=K+1                                                                OUT11150
      IK=3                                                                 OUT11160
      GO TO 250                                                            OUT11170
  190 IF(IND(N).GT.0)GO TO 200                                             OUT11180
      M=M+1                                                                OUT11190
      GO TO 210                                                            OUT11200
  200 K=K+1                                                                OUT11210
      IK=4                                                                 OUT11220
      GO TO 250                                                            OUT11230
  210 IF(INB(N).EQ.0)GO TO 270                                             OUT11240
      IF(IHLE(N).GT.0)GO TO 220                                            OUT11250
      M=M+1                                                                OUT11260
      GO TO 230                                                            OUT11270
  220 K=K+1                                                                OUT11280
      IK=5                                                                 OUT11290
      GO TO 250                                                            OUT11300
  230 IF(IHTE(N).GT.0)GO TO 240                                            OUT11310
      GO TO 270                                                            OUT11320
  240 K=K+1                                                                OUT11330
      IK=6                                                                 OUT11340
  250 DO 260 J = 1,2                                                       OUT11350
  260 FUND(J,K)=HEAD(J,IK)                                                 OUT11360
      M=M+1                                                                OUT11370
      GO TO 120                                                            OUT11380
  270 GO TO ( 280 , 300 , 320 , 340 , 360 , 380 ),K                        OUT11390
  280 WRITE(6, 290 )N,((FUND(J,L),J=1,2),L=1,K)                            OUT11400
  290 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,2H = ,2A10)                    OUT11410
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT11420
      IF(LOGIC.NE.3)WRITE(6, 100 )                                         OUT11430
      GO TO 400                                                            OUT11440
  300 WRITE(6, 310 )N,((FUND(J,L),J=1,2),L=1,K)                            OUT11450
  310 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,2H = ,2A10,3H + ,2A10)         OUT11460
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT11470
      IF(LOGIC.NE.3)WRITE(6, 100 )                                         OUT11480
      GO TO 400                                                            OUT11490
  320 WRITE(6, 330 )N,((FUND(J,L),J=1,2),L=1,K)                            OUT11500
  330 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,2H = ,2A10,3H + ,2A10,3H + ,OUT11510
     12A10)                                                                OUT11520
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT11530
      IF(LOGIC.NE.3)WRITE(6, 100 )                                         OUT11540
      GO TO 400                                                            OUT11550
  340 WRITE(6, 350 )N,((FUND(J,L),J=1,2),L=1,K)                            OUT11560
  350 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,2H = ,2A10,3H + ,2A10,3H + ,OUT11570
     12A10/48X,2H + ,2A10)                                                 OUT11580
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT11590
      GO TO 400                                                            OUT11600
  360 WRITE(6, 370 )N,((FUND(J,L),J=1,2),L=1,K)                            OUT11610
  370 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,2H = ,2A10,3H + ,2A10,3H + ,OUT11620
     12A10/48X,2H + ,2A10,3H + ,2A10)                                      OUT11630
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT11640
      GO TO 400                                                            OUT11650
  380 WRITE(6, 390 )N,((FUND(J,L),J=1,2),L=1,K)                            OUT11660
  390 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,2H = ,2A10,3H + ,2A10,3H + ,OUT11670
     12A10/48X,2H + ,2A10,3H + ,2A10,3H + ,2A10)                           OUT11680
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 90 )                           OUT11690
  400 CONTINUE                                                             OUT11700
  410 IF(LOGIC.EQ.3.AND.ISTAB.EQ.0)WRITE(6, 420 )NCASES                    OUT11710
  420 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,50H =    INDUCED CAMBER DUE TOUT11720
     1O PITCHING RATE ABOUT XCG,//)                                        OUT11730
      IF(LOGIC.EQ.3.AND.ISTAB.EQ.1)WRITE(6, 430 )NCASES                    OUT11740
  430 FORMAT(1H ,28X,16HFUNDAMENTAL CASE,I3,52H =    INDUCED CAMBER DUE TOUT11750
     1O ROLLING RATE ABOUT X-AXIS,//)                                      OUT11760
      IF(LOGIC.EQ.1)ILINES=15+3*NCM1                                       OUT11770
      IF(LOGIC.EQ.3)ILINES=15+4*NCM1                                       OUT11780
      WRITE(6, 440 )                                                       OUT11790
  440 FORMAT(1H0,/)                                                        OUT11800
      DO 640 K=1,NROWS                                                     OUT11810
      IWK=IW(K)                                                            OUT11820
      NWK=NW(K)                                                            OUT11830
      NJK=NJ(K)                                                            OUT11840
      IJK=IJ(K)                                                            OUT11850
      ILINES=ILINES+NW(K)+NJ(K)+15                                         OUT11860
      IF(ILINES.LT.60)GO TO 460                                            OUT11870
      ILINES=NW(K)+NJ(K)+22                                                OUT11880
      WRITE(6, 450 )                                                       OUT11890
  450 FORMAT(1H1)                                                          OUT11900
  460 WRITE(6, 470 )K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)OUT11910
  470 FORMAT(1H ,1X,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,            OUT11920
     1        7HDELTA =,F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,   OUT11930
     2        7HCHORD =,F10.6,2X,7HTANLE =,F10.6)                          OUT11940
      WRITE(6, 480 )                                                       OUT11950
  480 FORMAT(1H0,32X,44(1H.),12HINPUT VALUES,43(1H.))                      OUT11960
      WRITE(6, 490 )                                                       OUT11970
  490 FORMAT(1H0,35X,3HFC1,7X,3HFC2,7X,3HFC3,7X,3HFC4,7X,3HFC5,7X,3HFC6,OUT11980
     1        7X,3HFC7,7X,3HFC8,7X,3HFC9,7X,4HFC10)                        OUT11990
      IF(LOGIC.EQ.3)GO TO 530                                              OUT12000
      WRITE(6, 500 )(TST(K,L),L=1,NCASES)                                  OUT12010
  500 FORMAT(1H ,20X,10HTWIST(DEG),1X,10(1X,F9.5))                         OUT12020
      IF(NJK .GT. 0) WRITE(6, 510 )(BETA(IJK,L),L=1,NCASES)                OUT12030
  510 FORMAT(1H ,20X,10HDJ    (DEG),1X,10(1X,F9.5))                        OUT12040
      WRITE(6, 520 )(HL(K,L),L=1,NCASES)                                   OUT12050
  520 FORMAT(1H ,20X,10HHL    (Z/C),1X,10(1X,F9.5))                        OUT12060
      GO TO 540                                                            OUT12070
  530 WRITE(6, 500 )(TST(K,L),L=1,NCM1)                                    OUT12080
      IF(NJK .GT. 0) WRITE(6, 510 )(BETA(IJK,L),L=1,NCM1)                  OUT12090
      WRITE(6, 520 )(HL(K,L),L=1,NCM1)                                     OUT12100
  540 WRITE(6, 550 )                                                       OUT12110
  550 FORMAT(1H0,1X,13HWING ELEMENTS,18X,48(1H.),3HEPS,48(1H.))            OUT12120
      WRITE(6, 560 )                                                       OUT12130
  560 FORMAT(1H0,1X,2HNO,3X,3HX/C,7X,1HX,8X,3HDEL,6X,3HFC1,7X,3HFC2,       OUT12140
     17X,3HFC3,7X,3HFC4,7X,3HFC5,7X,3HFC6,7X,3HFC7,7X,3HFC8,7X,3HFC9,      OUT12150
     27X,4HFC10)                                                           OUT12160
      INWK=IWK+NWK-1                                                       OUT12170
      DO 570 J=IWK,INWK                                                    OUT12180
  570 WRITE(6, 580 )J,XB(J),XI(J),DEL(J),(EPS(J,L),L=1,NCASES)             OUT12190
  580 FORMAT(1H ,I3,1X,F8.5,1X,F10.5,1X,F7.5,10(1X,F9.5))                  OUT12200
C  PRINT CHORDWISE DATA ON JET                                             OUT12210
      IF(NJK.GT.0)GO TO 600                                                OUT12220
      WRITE(6, 590 )                                                       OUT12230
  590 FORMAT(1H0,/,1X,19HTHIS ROW HAS NO JET,//)                           OUT12240
      GO TO 640                                                            OUT12250
  600 WRITE(6, 610 )                                                       OUT12260
  610 FORMAT(1H0,1X,12HJET ELEMENTS,19X,47(1H.),5HTHETA,47(1H.))           OUT12270
```

```
      WRITE(6, 560 )
      IJKP1=IJK+1
      INJK=IJK+NJK-1
      WRITE(6, 580 )IJK,XB(IJK),XI(IJK),DEL(IJK),(THETA(K,L),L=1,NCASES)
      DO 620 J=IJKP1,INJK
  620 WRITE(6, 630 )J,XB(J),XI(J),DEL(J)
  630 FORMAT(1H ,I3,1X,F8.5,1X,F10.5,1X,F7.5)
      WRITE(6, 440 )
  640 CONTINUE
      RETURN
      END
```

```
*DECK INCOMP
      SUBROUTINE INCOMP(NCASES,IR)
C
C  THIS SUBROUTINE READS IN THE COMPOSITE CASE REQUIREMENTS
C  WHICH DEFINE THE FUNDAMENTAL CASES AND THEIR DEFLECTION MAGNITUDE
C  FOR SUPERPOSITION IN UP TO 24 COMBINATIONS
C
      COMMON/COMPOS/FACTOR(10,24),NCC
      DIMENSION FUNNY(10),ND(10)
C
C  INITIALIZE THE ARRAY OF FUNDAMENTAL CASE DEFLECTIONS
      DO 30 M = 1,24
      DO 20 N = 1,10
      FACTOR(N,M) = 0.00
   20 CONTINUE
   30 CONTINUE
C
C  READ THE COMPOSITE CASE DATA, CONSISTING OF FUNDAMENTAL CASE
C  DEFLECTIONS, IN DEGREES
      NCC = 0
   40 NCC = NCC + 1
      READ(5, 50 )(ND(N),FUNNY(N),N=1,10)
      IF(EOF(5)) 140 , 60
   50 FORMAT(10(I2,F6.4))
C  CHECK THE VALIDITY OF THE DATA
   60 IF(ND(1) .GT. 10) GO TO 110
      IF(NCC .GT. 24) GO TO 120
      DO 100 N = 1,10
      IF(ND(N) .GT. NCASES) GO TO 80
      IF(ND(N) .LT. 1) GO TO 100
C  THE DATA IS OK.  DEFINE FACTOR.
      NON = ND(N)
   70 FACTOR(NON,NCC) = FUNNY(N)
      GO TO 100
   80 WRITE(6, 90 )
   90 FORMAT(1H0,22X,76HAN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN
     1 FOUND.  IT WILL BE IGNORED.)
  100 CONTINUE
      GO TO 40
C  THE END OF THE INPUT DATA HAS BEEN REACHED
  110 NCC = NCC - 1
      IF(NCC .GT. 24) NCC = 24
      IR = 1
      RETURN
C  TOO MANY COMPOSITE CASES HAVE BEEN REQUESTED.  READ ON UNTIL AN END
C  CARD IS FOUND.
  120 WRITE(6, 130 )
  130 FORMAT(1H0,20X,47HMORE THAN 24 COMPOSITE CASES HAVE BEEN INPUT
     1       34HSUBSEQUENT INPUTS WILL BE IGNORED.)
      GO TO 40
C
C  AN END OF FILE HAS BEEN READ.  PRINT A MESSAGE AND QUIT.
  140 WRITE(6, 150 )
  150 FORMAT(1H1/// 31X, 35HAN END OF FILE HAS BEEN READ DURING,
     1      21H COMPOSITE CASE INPUT)
      IR = 2
      RETURN
      END
```

```
*DECK INTAIL
      SUBROUTINE INTAIL
C  THIS SUBROUTINE READS IN THE GEOMETRIC DATA TO DESCRIBE THE
C  HORIZONTAL AND VERTICAL TAILS.
      COMMON/SPIRIT/ NEWMAX,NEWUMU,NDALFA,LOGIC,IR,ISTAB
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,MC,ISY,IPR,JET,IGT,IHI,IGR
      COMMON/TAIL1/AAH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,
     1               NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2               ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),
     1      IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),
     2               CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),
     3               CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),
     4      SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)
      LOGICAL L1,L2
      L1 = NROWH .EQ. 0
      L2 = NROWV .EQ. 0
      HL = 0.0
      HH = 0.0
      VL = 0.0
      VH = 0.0
      B2 = SPA / 2.0
      IF(L1)GO TO 60
      READ(5, 20 )ARH,TRH,SWH,SPANH,HL,HH
      READ(5, 20 )(YH(K),K=1,NROWH)
   20 FORMAT(8F10.6)
      DELTAH(1) = 1.00 - YH(1)
      DO 30  K = 2,NROWH
      DELTAH(K) =YH(K-1) - DELTAH(K-1) -YH(K)
      IF(DELTAH(K) .LT. 0.0)GO TO 290
   30 CONTINUE
      SPANH = SPANH / B2
      HL = HL / B2
      HH = HH / B2
      DO 40 K = 1,NROWH
      DELTAH(K) = DELTAH(K) * SPANH / 2.0
   40 YH(K) = YH(K) * SPANH / 2.0
      READ(5, 50 )(IHX(K),K=1,NROWH)
      IF(ICAMH .NE. 0)READ(5, 50 )(IHC(K),K=1,NROWH)
   50 FORMAT(20I2)
   60 IF(L2)GO TO 90
      READ(5, 20 )ARV,TRV,SWV,SPANV,VL,VH
      READ(5, 20 )(ZV(K),K=1,NROWV)
      DELTAV(1) = 1.0 - ZV(1)
      DO 70   K = 2,NROWV
      DELTAV(K) = ZV(K-1) - DELTAV(K-1) - ZV(K)
      IF(DELTAV(K) .LT. 0.0)GO TO 310
   70 CONTINUE
      SPANV = SPANV / B2
      VL = VL / B2
      VH = VH / B2
      DO 80 K = 1,NROWV
      DELTAV(K) = DELTAV(K) * SPANV
   80 ZV(K) = ZV(K) * SPANV
      READ(5, 50 )(IVX(K),K=1,NROWV)
      IF(ICAMV .NE. 0)READ(5, 50 )(IVC(K),K=1,NROWH)
   90 NXTYPE = 0
      NCTYPE = 0
      IF(L1)GO TO 120
      DO 100 K = 1,NROWH
      IF(IHX(K) .GT. NXTYPE)NXTYPE = IHX(K)
  100 CONTINUE
      IF(ICAMH .EQ. 0)GO TO 120
      DO 110 K = 1,NROWH
      IF(IHC(K) .GT. NCTYPE)NCTYPE = IHC(K)
  110 CONTINUE
  120 IF(L2)GO TO 150
      DO 130 K = 1,NROWV
      IF(IVX(K) .GT. NXTYPE)NXTYPE = IVX(K)
  130 CONTINUE
      IF(ICAMV .EQ. 0)GO TO 150
      DO 140 K = 1,NROWV
      IF(IVC(K) .GT. NCTYPE)NCTYPE = IVC(K)
  140 CONTINUE
  150 IF(NXTYPE .GT. 5)GO TO 250
      IF(NCTYPE .GT. 5)GO TO 270
      READ(5, 50 )(NX(N),N=1,NXTYPE)
      IF((ICAMH.NE.0).OR.(ICAMV.NE.0))READ(5, 50 )(NC(N),N=1,NCTYPE)
      NMAXH = 0
      IF(L1)GO TO 170
      DO 160 K = 1,NROWH
      ICK = IHX(K)
  160 NMAXH = NMAXH + NX(ICK)
  170 NMAXV = 0
      IF(L2)GO TO 190
      DO 180 K = 1,NROWV
      ICK = IVX(K)
  180 NMAXV = NMAXV + NX(ICK)
  190 NTOTAL = NMAXH + NMAXV
      IF((NMAXH .GT. 80) .OR. (NMAXV .GT. 20)) GO TO 330
      DO 200 N = 1,NXTYPE
      NXN = NX(N)
      READ(5, 20 )(XBT(L,N),L=1,NXN)
  200 CONTINUE
  220 DO 230 I = 1,NMAXH
  230 CAMH(I) = 0.0
      DO 240 I = 1,NMAXV
  240 CAMV(I) = 0.0
      DO 210 N = 1,NCTYPE
      NXN = NC(N)
      READ(5, 20 )(CAMBER(L,N),L=1,NXN)
  210 CONTINUE
      RETURN
  250 WRITE(6, 260 )NXTYPE
  260 FORMAT(1H1/45X,26HNUMBER OF TAIL ROW TYPES =,I3)
      IR = 2
      RETURN
  270 WRITE(6, 280 )NCTYPE
  280 FORMAT(1H1/45X,29HNUMBER OF TAIL CAMBER TYPES =,I3)
      IR = 2
      RETURN
  290 WRITE(6, 300 )
  300 FORMAT(1H1,35X,46HTHERE IS AN ERROR IN HORIZONTAL TAIL  Y INPUTS)
      IR = 2
      RETURN
  310 WRITE(6, 320 )
  320 FORMAT(1H1,35X,43HTHERE IS AN ERROR IN VERTICAL TAIL Z INPUTS)
      IR = 2
      RETURN
  330 WRITE(6,340 )NMAXH,NMAXV
  340 FORMAT(1H1,38X,7HNMAXH =,I10,10X,7HNMAXV =,I10)
      IR = 2
      RETURN
      END
```

```
*DECK INFUS
      SUBROUTINE INFUS
C  THIS SUBROUTINE READS IN THE GEOMETRY DEFINITION FOR THE FUSELAGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI,IGR
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1               IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),
     2               XTR(40),XTL(40),CHR(40),CHL(40)
      COMMON/TAIL1/AAH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HX,HH,VL,VH,
     1               NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2               ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON/FUSLGI/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),
     1                RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2                V(40,7),W(40,7)
      COMMON/SLIPIT/SSLIP
      DIMENSION A(84)
      DIMENSION YF(21),ZF(21),YA(21),ZA(21)
      EQUIVALENCE (A(1),YF(1)),(A(22),ZF(1)),(A(43),YA(1)),(A(64),ZA(1))
      PI = 3.1415927
      SSS = SIN(SSLIP/57.295779)
      CSS = COS(SSLIP/57.295779)
C  READ THE SECTION BOUNDARIES
      NSEG1 = NSEG + 1
      READ(5, 20 )(XB(I),I = 1,NSEG1)
   20 FORMAT(8F10.0)
C  READ BODY CAMBER
      IF(ICAMF .GT. 0)GO TO 40
      DO 30  I = 1,NSEG1
   30 ZCL(I) = 0.0
      GO TO 50
   40 READ(5, 20 )(ZCL(I),I = 1,NSEG1)
   50 II = NTOTAL
      B2 = SPA / 2.0
      DO 60  I = 1,NSEG
      II = II + 1
```

```
      DX = 0.5 * (XB(I+1) + XB(I)) / B2
      SQUIG(II) = DX * CSS
      ETA(II) = DX * SSS
   60 ZETA(II) = 0.5 * (ZCL(I+1) + ZCL(I)) / B2
      READ(5, 70 )(NF(I),I = 1,NSEG)
   70 FORMAT(40I2)
C  READ IN FUSELAGE CONTOURS
C           ISECT = 1 - ARBITRARY SECTIONS
C           ISECT = 2 - CIRCULAR SECTIONS
C           ISECT = 3 - ELLIPTICAL SECTIONS
      GO TO ( 80 , 140 , 160 ),ISECT
C  ARBITRARY FUSELAGE CROSS - SECTIONS
   80 IWRITE = 2.0 * NMAX
      DO 130 I = 1,NSEG
      IWRITE = IWRITE + 1
      N = NF(I)
      NI = N + 1
      IF(I .GT. 1)GO TO 90
      READ(5, 20 )(YF(J),J = 1,NI)
      READ(5, 20 )(ZF(J),J = 1,NI)
      GO TO 120
   90 IF(NF(I) .EQ. NF(I-1))GO TO 100
      READ(5, 20 )(YF(J),J = 1,NI)
      READ(5, 20 )(ZF(J),J = 1,NI)
      GO TO 120
  100 DO 110 J = 1,NI
      YF(J) = YA(J)
  110 ZF(J) = ZA(J)
  120 READ(5, 20 )(YA(J),J = 1,NI)
      READ(5, 20 )(ZA(J),J = 1,NI)
      CALL WRITMS(9,A,84,IWRITE)
  130 CONTINUE
      RETURN
C  CIRCULAR FUSELAGE CROSS - SECTIONS
  140 READ(5, 20 )(RADIUS(I),I = 1,NSEGI)
      DO 150 I = 1,NSEGI
  150 CSA(I) = PI * RADIUS(I)**2
      RETURN
C  ELLIPTICAL FUSELAGE CROSS - SECTIONS
  160 READ(5, 20 )(AMIN(I),I = 1,NSEGI)
      READ(5, 20 )(AMAJ(I),I = 1,NSEGI)
      RETURN
      END
```

```
*DECK XEDGE
      SUBROUTINE XEDGE
C
C  READ ADDITIONAL INPUTS FOR FLOW FIELD CALCULATIONS
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),
     1             D(40),XK(400),ITYPE(400)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI,IGR
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1             IFFLD,ICR,AINF(40,10),XLR(40),XLL(40),
     2             XTR(40),XTL(40),CHR(40),CHL(40)
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON/FUSLG1/PI,NSEG,NSEGI,NF(40),ISECT,ICAMF,XF(41),ZCL(41),
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2              V(40,7),W(40,7)
C
      ALPHA = -5.0
      DO 20 I = 1,5
      ALPHA = ALPHA + 5.0
   20 AL(I) = ALPHA
      NP1 = NTOTAL + NSEG + 1
      IF(NPOINT .NE. 0)READ(5, 30 )(SQUIG(I),ETA(I),ZETA(I),I = NP1,
     1   NTPNP)
   30 FORMAT(3F10.0)
C  SET UP THE ADDITIONAL GEOMETRY REQUIRED FOR FLOW FIELDS
      DO 40 I = 1,NROWS
      DTT = DELTA(I) * TANLE(I)
      XLR(I) = XLEAD(I) + DTT
      XLL(I) = XLEAD(I) - DTT
      DTT = DELTA(I) * TANTE(I)
      XTR(I) = XTRAIL(I) + DTT
      XTL(I) = XTRAIL(I) - DTT
      CHR(I) = XTR(I) - XLR(I)
      CHL(I) = XTL(I) - XLL(I)
   40 CONTINUE
C  NORMALIZE THE FLOW FIELD POINT COORDINATES
      B2 = SPA / 2.00
      IF(NPOINT .EQ. 0)GO TO 60
      DO 50 I = NP1,NTPNP
      SQUIG(I) = SQUIG(I) / B2
      ETA(I) = ETA(I) / B2
      ZETA(I) = ZETA(I) / B2
   50 CONTINUE
   60 ICASES = NCC
      RETURN
      END
```

```
*DECK BLOWIN
      SUBROUTINE BLOWIN(JETFLG,IR)
C
C  THIS SUBROUTINE READS THE SECTIONAL JET BLOWING STRENGTH
C  CMU(K) = J / (Q * CHORD(K))
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/JET12/UJ(10)?DDO(10),THTE(10),HH(10),ZMX(10),XO(10),YO(15),
     1             ZO(10),DI E(15),ISY(15),JETRE(10)/NJETS
      DIMENSION DCMU(40)
C
      NJETS = 0
      IF(JETFLG .NE. 0) GO TO 40
C  READ THE CMU DATA ONLY FOR THOSE SECTIONS WHICH HAVE A JET
      READ(5, 20 )(DCMU(K),K = 1,NROWSJ)
      IF(EOF(5)) 70 , 30
   20 FORMAT(8F10.6)
   30 IF(DCMU(1) .LT. 800.0) GO TO 40
      IR = 2
      RETURN
C
C  REARRANGE THE DATA INTO THE PROPER SEQUENCE
   40 KP = 0
      DO 60 K = 1,NROWS
   50 CMU(K) = 0.00
      IF(NJ(K) .EQ. 0) GO TO 60
      KP = KP + 1
      CMU(K) = DCMU(KP)
   60 CONTINUE
      IR = 1
      RETURN
C
C  AN END OF FILE HAS BEEN READ.  THIS RUN IS COMPLETELY FINISHED.
   70 WRITE(6, 80 )
   80 FORMAT(1H1///41X,37HNO MORE CMU CASES HAVE BEEN REQUESTED)
      IR = 3
      RETURN
      END
```

```
*DECK INJETS
      SUBROUTINE INJETS(NEWMAX,IR)
C
C  THIS SUBROUTINE READS THE CIRCULAR JET INPUT DATA AND STORES IT
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/JET12/ UJ(10),DDO(10),THTA(10),HH(10),ZMX(10),XO(10),
     1             YO(10),ZO(10),DIHE(10),ISY(10),JETRE(10),NJETS
      COMMON/JET13/CJ(10),THO(10)
C
      NEWMAX = NWT
      NROWSJ = 0
      DO 20 K = 1,NROWS
      CMU(K) = 0.0
   20 CONTINUE
      NJETS = 0
      N = 0
   30 N = N + 1
C  READ THE INPUT
   40 READ(5, 50 ) CJ(N),DDO(N),THO(N),HH(N),
     1                XO(N),YO(N),ZO(N),DIHE(N),ISY(N),JETRE(N),LAST
      IF(EOF(5)) 150 , 60
   50 FORMAT(7F10.0,F7.0,3I1)
   60 IF(CJ(N) .GT. 8999.) GO TO 100
      ATHO = ABS(THO(N))
      IF((ATHO .LT. 1.0) .OR. (ATHO .GT. 179.0)) GO TO 130
      UJ(N) = SQRT(CJ(N) / (1.570795*DDO(N)**2/AREA))
      THTA(N) = 90.0 - THO(N)
      ZMX(N) = 10.0 * UJ(N) + 10.0
      IF(HH(N) .NE. 0.0) GO TO 70
      CJP = CJ(N) * AREA / DDO(N)**2
      IF(THTA(N) .GE. 0.0) HH(N) = (-0.000120*THTA(N)+0.0210)
     1        * SQRT(CJP) * DDO(N)
      IF(THTA(N) .LT. 0.0) HH(N) = (0.000120*THTA(N)+0.0210)
     1        * SQRT(CJP) * DDO(N)
      IF((CJP .GT. 50.0) .AND. (THTA(N) .GT. 80.0)
     1     .AND. (THTA(N) .LT. 100.0)) HH(N) = 0.070 * DDO(N)
   70 IF(JETRE(N) .EQ. 0) GO TO 90
C  DEFINE THE PARAMETERS FOR A JET SIMILAR TO THE PREVIOUS ONE
      NI = N-1
      CJ(N) = CJ(NI)
      UJ(N) = UJ(NI)
      DDO(N) = DDO(NI)
   80 THTA(N) = THTA(NI)
      HH(N) = HH(NI)
      ZMX(N) = ZMX(NI)
   90 IF((LAST .EQ. 0) .AND. (N .EQ. 10)) GO TO 110
      IF(LAST .EQ. 0) GO TO 30
      IR = 4
      NJETS = N
      RETURN
  100 IR = 2
      NJETS = N - 1
      RETURN
C  AN ERROR HAS OCCURED
  110 IR = 5
      NJETS = N
      WRITE(6, 120 )
  120 FORMAT(1H0/ 35X, 41HNO LAST FLAG FOUND WHILE READING CIRCULAR,
     1        10H JET INPUT)
      RETURN
  130 IR = 5
      NJETS = N - 1
      WRITE(6, 140 )
  140 FORMAT(1H0 / 32X,43HTHTA ANGLE MUST BE IN 0 TO 180 OR 0 TO -180,
     1        13H DEGREE RANGE)
      RETURN
  150 NJETS = N - 1
      IR = 3
      RETURN
      END
```

```
*DECK STAGE2
      OVERLAY(OVLY,2,0)
      PROGRAM STAGE2
C
C  THIS PROGRAM CONTROLS THE CALCULATION OF ALL EVD DOWNWASH
C  INFLUENCE COEFFICIENTS AND THE FORMATION OF THE LEFT AND RIGHT SIDE
C  MATRICES.
C
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE
      DIMENSION W(400),WI(400),WS(400),WSI(400)
      DIMENSION ONSET(400),ONSETS(400)
      LOGICAL HING
      HING = IHINGE .EQ. 0
C  DECIDE WHICH STORAGE UNIT TO DEAL WITH
      IO = 1
```

```
      IF(IGND .NE. 0) IO = 8                                            STG20190
C  CHECK WHETHER THIS IS A NEW GROUND HEIGHT CASE OR A NEW CMU CASE     STG20200
      IST = 0                                                           STG20210
      IF(LOGIC .GE. 0) GO TO 20                                         STG20220
      IST = 1                                                           STG20230
      LOGIC = 1                                                         STG20240
   20 IF(NHITE .GT. 1) GO TO 30                                         STG20250
      IF(NEWCMU .EQ. 1) GO TO 30                                        STG20260
C  AUGMENT THE JET ROWS OF THE OLD MATRIX FOR A NEW CMU CASE            STG20270
      ISIZE = NEWMAX                                                    STG20280
      IF(NEWMAX .GT. NWT) CALL SHUFL2(W,ISIZE,IGND,IO,ISTAB)            STG20290
      GO TO 50                                                          STG20300
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENTS AT ALL CONTROL POINTS  STG20310
C  DUE TO ALL TRIANGULAR, LEADING EDGE AND JET-INFINITY EVD ELEMENTS    STG20320
C  AND GROUND EFFECT                                                    STG20330
   30 ISIZE = NMAX                                                      STG20340
      IF(LOGIC .EQ. 2) GO TO 40                                         STG20350
      CALL DWNWSH(W,WI,WS,WSI,ONSET,ONSETS,ISIZE,IST)                   STG20360
C  AUGMENT THE MATRIX ROWS FOR CONTROL POINTS ON THE JET.               STG20370
C  NOTE THAT THIS MUST BE DONE EVEN THOUGH CMU MAY BE ZERO,             STG20380
C  IN ORDER TO PREPARE FOR FUTURE NONZERO CMU CASES.                    STG20390
   40 IF(NMAX .GT. NWT) CALL SHUFL1(W,ISIZE,IGND,IO,ISTAB)              STG20400
C  DEVELOP THE RIGHT SIDE COLUMN MATRIX                                 STG20410
   50 ISIZE = NEWMAX                                                    STG20420
      DO 100 N = 1,NCASES                                               STG20430
      LCASE = N                                                         STG20440
C  DEFINE THE LCASE COLUMN, NOT INCLUDING THE INFLUENCE OF ANY HINGES   STG20450
   60 CALL COLUM1(LCASE)                                                STG20460
C  COMPUTE THE HINGE DOWNWASH INFLUENCE FACTORS                          STG20470
      IF((LCASE .EQ. 1) .OR. IHINGI) GO TO 100                          STG20480
      DO 70 I = 1,NEWMAX                                                STG20490
      W(I) = 0.00                                                       STG20500
   70 CONTINUE                                                          STG20510
   80 CALL HINGE(W,ISIZE,NEWMAX,LCASE)                                  STG20520
C  MODIFY THE LCASE COLUMN TO INCLUDE THE INFLUENCE OF ALL HINGES       STG20530
   90 CALL COLUM2(W,ISIZE,NEWMAX,LCASE)                                 STG20540
  100 CONTINUE                                                          STG20550
C  THE MATRIX DEVELOPMENT IS NOW COMPLETE.                              STG20560
      CONTINUE                                                          STG20570
      END                                                               STG20580


*DECK DWNWSH                                                            DWNW0010
      SUBROUTINE DWNWSH(W,WI,WS,WSI,ONSET,ONSETS,ISIZE,ISTAB)           DWNW0020
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            DWNW0030
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) DWNW0040
      COMMON/GEOMI/Y(40),CHORD(40),DELTA(40),KB(400),XI(400),DEL(400),  DWNW0050
     1             D(40),KK(400),ITYPE(400)                             DWNW0060
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          DWNW0070
      DIMENSION W(ISIZE),WI(ISIZE),WS(ISIZE),WSI(ISIZE)                 DWNW0080
      DIMENSION ONSET(ISIZE),ONSETS(ISIZE)                              DWNW0090
      LOGICAL STAB,GE,SYMM,NGE                                          DWNW0100
      STAB = ISTAB.NE.0                                                 DWNW0110
      GE = IGND.NE.0                                                    DWNW0120
      SYMM = ISYMM .LE. 0                                               DWNW0130
      NGE = IGND .EQ. 0                                                 DWNW0140
      IF(GE)ZZ = GHITE(NHITE) * 2.0                                     DWNW0150
      IWRITE = -1                                                       DWNW0160


      IF(IPRINT .LT. 0)WRITE(6, 20 )                                    DWNW0170
   20 FORMAT(1H1,38X,43HWING - DUE - TO - WING - AND - JET DOWNWASH/)   DWNW0180
C  I = CONTROL POINT                                                    DWNW0190
C  J = INFLUENCING POINT                                                DWNW0200
      DO 290 I = 1,NMAX                                                 DWNW0210
      IWRITE = IWRITE + 2                                               DWNW0220
      IWR1=IWRITE + 1                                                   DWNW0230
      KI = KK(I)                                                        DWNW0240
      IF(NGE)GO TO 30                                                   DWNW0250
      CALL READMS(1,W,ISIZE,IWRITE)                                     DWNW0260
      IF(STAB) CALL READMS(1,WS,ISIZE,IWR1)                             DWNW0270
   30 DO 190 J = 1,NMAX                                                 DWNW0280
      KJ = KK(J)                                                        DWNW0290
      X  = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)                        DWNW0300
      YY = Y(KI) - Y(KJ)                                                DWNW0310
      YR = Y(KI) + Y(KJ)                                                DWNW0320
      IT = ITYPE(J)/10                                                  DWNW0330
      IF(GE)GO TO 70                                                    DWNW0340
C  DECIDE WHICH TYPE OF EVD TO USE.  ONLY THE TRIANGULAR PART OF THE    DWNW0350
C  HINGE WILL BE CONSIDERED AT THIS TIME.                               DWNW0360
      GO TO ( 40 , 50 , 60 , 40 ),IT                                    DWNW0370
C  REGULAR TRIANGULAR EVD (INCLUDING TRIANGULAR PART OF HINGE EVD)      DWNW0380
   40 D1 = DEL(J-1) * CHORD(KJ)                                         DWNW0390
      IN1 = IW(KJ) + NW(KJ) - 1                                         DWNW0400
      IF(J .EQ. IJ(KJ))D1 = DEL(IN1) * CHORD(KJ)                        DWNW0410
      D2 = DEL(J) * CHORD(KJ)                                           DWNW0420
      DW = EVD1(X,YY,D1,D2,DELTA(KJ))                                   DWNW0430
      IF(SYMM)DWR = EVD1(X,YR,D1,D2,DELTA(KJ))                          DWNW0440
      GO TO 110                                                         DWNW0450
C  LEADING EDGE EVD                                                     DWNW0460
   50 D2 = DEL(J) * CHORD(KJ)                                           DWNW0470
      DW = EVD2(X,YY,D2,DELTA(KJ))                                      DWNW0480
      IF(SYMM)DWR = EVD2(X,YR,D2,DELTA(KJ))                             DWNW0490
      GO TO 110                                                         DWNW0500
C  FAR JET EVD                                                          DWNW0510
   60 D1 = DEL(J-1) * CHORD(KJ)                                         DWNW0520
      DW = EVD3(X,YY,D1,D(KJ),DELTA(KJ))                                DWNW0530
      IF(SYMM)DWR = EVD3(X,YR,D1,D(KJ),DELTA(KJ))                       DWNW0540
      GO TO 110                                                         DWNW0550
C  DECIDE WHICH TYPE OF GROUND EFFECTS INFLUENCE COEFFIEIENT TO USE     DWNW0560
   70 GO TO ( 80 , 90 , 100 , 80 ),IT                                   DWNW0570
C  REGULAR TRIANGULAR GROUND EFFECTS ELEMENT                            DWNW0580
   80 D1 = DEL(J-1)*CHORD(KJ)                                           DWNW0590
      IN1 = IW(KJ) + NW(KJ) - 1                                         DWNW0600
      IF(J .EQ. IJ(KJ))D1 = DEL(IN1)*CHORD(KJ)                          DWNW0610
      D2 = DEL(J) * CHORD(KJ)                                           DWNW0620
      CALL GROUND(X,YY,ZZ,D1,D2,DELTA(KJ),IT,DWI,DUI)                   DWNW0630
      IF(SYMM)CALL GROUND(X,YR,ZZ,D1,D2,DELTA(KJ),IT,DWIR,DUIR)         DWNW0640
      GO TO 150                                                         DWNW0650
C  LEADING EDGE GROUND EFFECTS ELEMENT                                  DWNW0660
   90 D2 = DEL(J) * CHORD(KJ)                                           DWNW0670
      CALL GROUND(X,YY,ZZ,D1,D2,DELTA(KJ),IT,DWI,DUI)                   DWNW0680
      IF(SYMM)CALL GROUND(X,YR,ZZ,D1,D2,DELTA(KJ),IT,DWIR,DUIR)         DWNW0690
      GO TO 150                                                         DWNW0700
C  FAR JET GROUND EFFECTS ELEMENT                                       DWNW0710
  100 D1 = DEL(J-1) * CHORD(KJ)                                         DWNW0720
      CALL GROUND(X,YY,ZZ,D1,D(KJ),DELTA(KJ),IT,DWI,DUI)                DWNW0730
      IF(SYMM)CALL GROUND(X,YR,ZZ,D1,D(KJ),DELTA(KJ),IT,DWIR,DUIR)      DWNW0740
      GO TO 150                                                         DWNW0750
C  PUT THE RESULTS INTO THE PROPER ARRAYS                               DWNW0760
```

```
  110 IF(ISYMM) 140 , 120 , 130                                         DWNW0770
  120 W(J) =  DW + DWR                                                  DWNW0780
      IF(STAB)WS(J) = DW - DWR                                          DWNW0790
      GO TO 190                                                         DWNW0800
  130 W(J) = DW                                                         DWNW0810
      GO TO 190                                                         DWNW0820
  140 W(J) = DW - DWR                                                   DWNW0830
      GO TO 190                                                         DWNW0840
  150 IF(ISYMM) 180 , 160 , 170                                         DWNW0850
  160 WI(J) = DWI + DWIR + W(J)                                         DWNW0860
      ONSET(J) = DUI + DUIR                                             DWNW0870
      IF(STAB)WSI(J) = DWI - DWIR +WS(J)                                DWNW0880
      IF(STAB)ONSETS(J) = DUI - DUIR                                    DWNW0890
      GO TO 190                                                         DWNW0900
  170 WI(J) = DWI +  W(J)                                               DWNW0910
      ONSET(J) = DUI                                                    DWNW0920
      GO TO 190                                                         DWNW0930
  180 WI(J) = DWI - DWIR + W(J)                                         DWNW0940
      ONSET(J) = DUI - DUIR                                             DWNW0950
  190 CONTINUE                                                          DWNW0960
C  WRITE THE I-TH ROWS ON THE APPROPRIATE DISKS                         DWNW0970
      IF(GE)GO TO 200                                                   DWNW0980
      CALL WRITMS(1,W,ISIZE,IWRITE)                                     DWNW0990
      IF(STAB) CALL WRITMS(1,WS,ISIZE,IWR1)                             DWNW1000
      GO TO 210                                                         DWNW1010
  200 CALL WRITMS(8,WI,ISIZE,IWRITE)                                    DWNW1020
      IF(STAB) CALL WRITMS(8,WSI,ISIZE,IWR1)                            DWNW1030
      CALL WRITMS(9,ONSET,ISIZE,IWRITE)                                 DWNW1040
      IF(STAB) CALL WRITMS(9,ONSETS,ISIZE,IWR1)                         DWNW1050
  210 IF(IPRINT .GE. 0)GO TO 290                                        DWNW1060
      IF(I .EQ. NWT+1)WRITE(6, 220 )                                    DWNW1070
  220 FORMAT(1H1,38X,42HJET - DUE - TO - WING - AND - JET DOWNWASH/)    DWNW1080
      IF(NGE)WRITE(6, 230 )I,W                                          DWNW1090
      IF(GE)WRITE(6, 240 )I,WI                                          DWNW1100
      IF(GE)WRITE(6, 270 )I,ONSET                                       DWNW1110
      IF(.NOT.STAB)GO TO 290                                            DWNW1120
      IF(NGE)WRITE(6, 250 )I,WS                                         DWNW1130
      IF(GE)WRITE(6, 260 )I,WSI                                         DWNW1140
      IF(GE)WRITE(6, 280 )I,ONSETS                                      DWNW1150
  230 FORMAT(1H0,55X,19HREGULAR MATRIX ROW ,I4,60(/,1X,10E13.5))        DWNW1160
  240 FORMAT(1H0,53X,25HGROUND EFFECT MATRIX ROW ,I4,60(/,1X,10E13.5))  DWNW1170
  250 FORMAT(1H0,54X,21HSTABILITY MATRIX ROW ,I4,60(/,1X,10E13.5))      DWNW1180
  260 FORMAT(1H0,48X,35HGROUND EFFECT STABILITY MATRIX ROW ,I4,         DWNW1190
     1       60(/,1X,10E13.5))                                          DWNW1200
  270 FORMAT(1H0,55X,22HONSET FLOW MATRIX ROW ,I4,60(/,1X,10E13.5))     DWNW1210
  280 FORMAT(1H0,50X,32HSTABILITY ONSET FLOW MATRIX ROW ,I4,60(/,1X,    DWNW1220
     1       10E13.5))                                                  DWNW1230
  290 CONTINUE                                                          DWNW1240
      RETURN                                                            DWNW1250
C  ** WOOPEE **                                                         DWNW1260
      END                                                               DWNW1270


*DECK EVD1                                                              EVD10010
      FUNCTION EVD1(X,Y,D1,D2,DELTA)                                    EVD10020
C                                                                       EVD10030
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y               EVD10040
C  DUE TO A REGULAR TRIANGULAR EVD ELEMENT WITH UNIT PEAK VORTICITY,    EVD10050


C  LOCATED AT THE ORIGIN 0,0                                            EVD10060
C                                                                       EVD10070
      R(A,B) = SQRT(A*A + B*B)                                          EVD10080
C                                                                       EVD10090
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS                           EVD10100
      IF(Y .LT. 0.0) Y = -Y                                             EVD10110
      YMD = Y - DELTA                                                   EVD10120
      YPD = Y + DELTA                                                   EVD10130
      PART1 = (D1 + D2) * (1.0/YMD - 1.0/YPD)                           EVD10140
      IF(X/(0.50*(D1+D2)) .GT. 100.0) GO TO 100                         EVD10150
   20 XPD = X + D1                                                      EVD10160
      XMD = X - D2                                                      EVD10170
   30 R0P = R(X,YMD)                                                    EVD10180
      R1P = R(XPD,YMD)                                                  EVD10190
      R2P = R(XMD,YMD)                                                  EVD10200
   40 R0PP = R(X,YPD)                                                   EVD10210
      R1PP = R(XPD,YPD)                                                 EVD10220
      R2PP = R(XMD,YPD)                                                 EVD10230
C                                                                       EVD10240
C  CALCULATE THE DOWNWASH                                               EVD10250
   50 PART2 = (XPD/D1) * ((R1P-R0P)/YMD - (R1PP-R0PP)/YPD)              EVD10260
      PART3 = (XMD/D2) * ((R2P-R0P)/YMD - (R2PP-R0PP)/YPD)              EVD10270
      YRATIO = (YPD+R0PP) / (YMD+R0P)                                   EVD10280
   60 PART4 = (XPD/D1) * ALOG(((YMD+R1P)/(YPD+R1PP)) * YRATIO)          EVD10290
      PART5 = (XMD/D2) * ALOG(((YMD+R2P)/(YPD+R2PP)) * YRATIO)          EVD10300
   70 PART6 = YMD * ALOG((XPD+R1P)/(X+R0P))                             EVD10310
      PART7 = YPD * ALOG((XPD+R1PP)/(X+R0PP))                           EVD10320
   80 PART8 = YMD * ALOG((XMD+R2P)/(X+R0P))                             EVD10330
      PART9 = YPD * ALOG((XMD+R2PP)/(X+R0PP))                           EVD10340
   90 EVD1 = -(PART1 + (PART2 + PART3) - 2.0*(PART4 + PART5)            EVD10350
     1     -((PART6 - PART7)/D1 + (PART8 - PART9)/D2)) / 25.13274       EVD10360
      RETURN                                                            EVD10370
  100 EVD1 = -PART1 / 12.56673                                          EVD10380
  110 RETURN                                                            EVD10390
      END                                                               EVD10400


*DECK EVD2                                                              EVD20010
      FUNCTION EVD2(X,Y,DEL,DELTE)                                      EVD20020
C                                                                       EVD20030
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y               EVD20040
C  DUE TO A LEADING EDGE EVD EPEMENT WITH UNIT AVERAGE VORTICITY,       EVD20050
C  LOCATED AT THE OR GIN 0,0                                            EVD20060
C                                                                       EVD20070
      DIMENSION SI(9),FACTOR(9)                                         EVD20080
C                                                                       EVD20090
      R(A,B) = SQRT(A*A + B*B)                                          EVD20100
      G(A) = 1.00/SQRT(A) - A                                           EVD20110
      S(A) = ABS(A) / A                                                 EVD20120
C                                                                       EVD20130
      DATA SI/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,          EVD20140
     1         0.3242534, 0.6133714, 0.8360311, 0.9681602/              EVD20150
      DATA FACTOR/0.0812744,0.1806482,0.2606107,0.3123471,0.3302394,    EVD20160
     1            0.3123471,0.2606107,0.1806482,0.0812744/              EVD20170
C                                                                       EVD20180
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS                           EVD20190
   20 XB = X / DEL                                                      EVD20200
      YB = Y / DEL                                                      EVD20210
```

```
      DB = DELTA / DEL                                                  EVD20220
      YPD = YB + DB                                                     EVD20230
      YMD = YB - DB                                                     EVD20240
   30 IF(XB .GT. 100.0) GO TO 290                                       EVD20250
      XM1 = XB - 1.00                                                   EVD20260
   40 ROP = R(XB,YMD)                                                   EVD20270
   50 ROPP = R(XB,YPD)                                                  EVD20280
C                                                                       EVD20290
C  CALCULATE RK(XB)                                                     EVD20300
      IF(ABS(XB) .LT. 1.0E-04) GO TO 110                                EVD20310
      IF(ABS(XM1) .LT. 1.0E-06) GO TO 90                                EVD20320
   60 PART = ALOG(ABS(XM1/XB))                                          EVD20330
      PART1 = XB * PART + 1.00                                          EVD20340
      IF(XB .GT. 0.00) GO TO 80                                         EVD20350
   70 SQX = SQRT(-XB)                                                   EVD20360
      RK = -2.00 / SQX * ATAN(1.00/SQX) + PART1                         EVD20370
      GO TO 100                                                         EVD20380
   80 SQX = SQRT(XB)                                                    EVD20390
      RK = -ALOG(ABS((1.00-SQX) /(1.00+SQX))) / SQX + PART1             EVD20400
      GO TO 100                                                         EVD20410
   90 RK = 2.386294                                                     EVD20420
C                                                                       EVD20430
C  CALCULATE P(XB)                                                      EVD20440
  100 PART2 = ROP/YMD - ROPP/YPD                                        EVD20450
      P = PART2 * RK                                                    EVD20460
      GO TO 120                                                         EVD20470
  110 P = 0.00                                                          EVD20480
C                                                                       EVD20490
C  CALCULATE F(XB) BY GAUSSIAN INTEGRATION.                             EVD20500
  120 IF((XB .GT. 0.0) .AND. (XB .LT. 1.00)) GO TO 160                  EVD20510
C  XB IS NOT WITHIN THE X DIMENSIONS OF THE ELEMENT.                    EVD20520
      F = 0.00                                                          EVD20530
      DO 150 N = 1,9                                                    EVD20540
  130 SB = (SI(N)+1.00) / 2.00                                          EVD20550
      XMS = XB - SB                                                     EVD20560
      GS = G(SB)                                                        EVD20570
  140 PART3 = (GS*(R(XMS,YMD)-ROP))/YMD - (GS*(R(XMS,YPD)-ROPP))/YPD    EVD20580
      F = F + FACTOR(N) * PART3 / XMS                                   EVD20590
  150 CONTINUE                                                          EVD20600
      F = 0.50 * F                                                      EVD20610
      GO TO 280                                                         EVD20620
C                                                                       EVD20630
C  XB IS WITHIN THE X DIMENSIONS OF THE ELEMENT.   CALCULATE F0.        EVD20640
  160 F0 = 0.00                                                         EVD20650
      GPX = 0.00                                                        EVD20660
      GPPX = 0.00                                                       EVD20670
      IF(XB .LT. 1.0E-04) GO TO 180                                     EVD20680
      GX = G(XB)                                                        EVD20690
      GPX = GX * (ABS(YMD) - ROP)                                       EVD20700
      GPPX = GX * (ABS(YPD) - ROPP)                                     EVD20710
      IF(XM1 .GT. -1.0E-06) GO TO 180                                   EVD20720
  170 F0 = -(GPX/YMD - GPPX/YPD) * PART                                 EVD20730
C  CALCULATE F1 BY GAUSSIAN INTEGRATION.                                EVD20740
  180 F1 = 0.00                                                         EVD20750
      DO 260 N = 1,9                                                    EVD20760
  190 SB = (SI(N)+1.00) / 2.00                                          EVD20770
      XMS = XB - SB                                                     EVD20780
  200 IF(ABS(XMS) .LT. 1.0E-04) GO TO 230                               EVD20790
      GS = G(SB)                                                        EVD20800
  210 PART4 = (GS*(R(XMS,YMD)-ROP) - GPX) / XMS                         EVD20810
```

```
  220 PART5 = (GS*(R(XMS,YPD)-ROPP) - GPPX) / XMS                       EVD20820
      PART6 = PART4/YMD - PART5/YPD                                     EVD20830
      GO TO 250                                                         EVD20840
  230 PART4 = S(YMD) - S(YPD) - PART2                                   EVD20850
      PART5 = 1.00 + 0.50 / (SQRT(XB))**3                               EVD20860
  240 PART6 = PART4 * PART5                                             EVD20870
  250 F1 = F1 + FACTOR(N) * PART6                                       EVD20880
  260 CONTINUE                                                          EVD20890
  270 F = F0 + 0.50 * F1                                                EVD20900
C                                                                       EVD20910
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT.                        EVD20920
  280 EVD2 = -(1.50 * (1.00/YMD - 1.00/YPD) + P + F) / 18.84956         EVD20930
      RETURN                                                            EVD20940
  290 EVD2 = -(1.00/YMD - 1.00/YPD) / 6.283185                          EVD20950
      RETURN                                                            EVD20960
      END                                                               EVD20970


*DECK EVD3                                                              EVD30010
      FUNCTION EVD3(X,Y,DEL,D,DELTA)                                    EVD30020
C                                                                       EVD30030
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y DUE TO A      EVD30040
C  FAR-JET EVD ELEMENT WITH UNIT PEAK VORTICITY, LOCATED AT THE ORIGIN  EVD30050
C                                                                       EVD30060
      R(A,B) = SQRT(A*A + B*B)                                          EVD30070
C                                                                       EVD30080
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS                           EVD30090
      IF(Y .LT. 0.0) Y = -Y                                             EVD30100
      YMD = Y - DELTA                                                   EVD30110
      YPD = Y + DELTA                                                   EVD30120
   20 XPD = X + D                                                       EVD30130
   30 PART1 = (DEL/2.00 + D) * (1.00/YMD - 1.00/YPD)                    EVD30140
C  CHECK ON INFINITY                                                    EVD30150
      IF((XPD/YMD)**2 .GT. 1.0E06) GO TO 170                            EVD30160
      XPD1 = X + DEL                                                    EVD30170
      XD = X/DEL                                                        EVD30180
      ROP = R(X,YMD)                                                    EVD30190
      ROPP = R(X,YPD)                                                   EVD30200
   40 R1P = R(XPD1,YMD)                                                 EVD30210
      R1PP = R(XPD1,YPD)                                                EVD30220
      RDP = R(XPD,YMD)                                                  EVD30230
      RDPP = R(XPD,YPD)                                                 EVD30240
C                                                                       EVD30250
C  CALCULATE F0                                                         EVD30260
   50 PART2 = ROP/YMD - ROPP/YPD                                        EVD30270
      PART3 = 0.50 * (XD+1.00) * ((R1P-ROP)/YMD - (R1PP-ROPP)/YPD)      EVD30280
   60 PART4 = ALOG(ABS(((YMD+R1P)/(YPD+R1PP)) * ((YPD+ROPP)/(YMD+ROP))))EVD30290
      PART5 = YMD/DEL * ALOG((XPD1+R1P)/(X+ROP))                        EVD30300
   70 PART6 = YPD/DEL * ALOG((XPD1+R1PP)/(X+ROPP))                      EVD30310
   80 F0 = PART1 - 0.50*PART2 + PART3 - (XD+1.00)*PART4                 EVD30320
     1        - 0.50*(PART5-PART6)                                      EVD30330
C                                                                       EVD30340
C  CALCULATE F1                                                         EVD30350
      IF(ABS(XPD/D) .LT. 1.0E-02) GO TO 140                             EVD30360
C  X IS NOT NEAR -D                                                     EVD30370
      Q = D/XPD                                                         EVD30380
   90 PART1 = -D * Q * (1.00/YMD - 1.00/YPD)                            EVD30390
      PART2 = Q * PART2                                                 EVD30400
```

```
  100 PART3 = Q * Q * ALOG(ABS((YMD+ROP)/(YPD+ROPP)))                   EVD30410
  110 PART4 = YMD/RDP*ALOG(ABS((-XPD+RDP*((ROP+RDP)/D))/(RDP-XPD)))     EVD30420
  120 PART5 = YPD/RDPP*ALOG(ABS((-XPD+RDPP*((ROPP+RDPP)/D))/(RDPP-XPD)))EVD30430
  130 F1 = PART1 + PART2 - PART3 + Q * Q * (PART4 - PART5)              EVD30440
      GO TO 160                                                         EVD30450
C  X IS NEAR -D                                                         EVD30460
  140 X = -D                                                            EVD30470
      ROP = R(X,YMD)                                                    EVD30480
      ROPP = R(X,YPD)                                                   EVD30490
      PART2 = ROP/YMD - ROPP/YPD                                        EVD30500
      PART3 = (X/YMD)**2 * ALOG(ABS((YMD+ROP)/X))                       EVD30510
      PART4 = (X/YPD)**2 * ALOG(ABS((YPD+ROPP)/X))                      EVD30520
  150 F1 = -0.50*PART2 - 0.50*(PART3 - PART4)                           EVD30530
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT                         EVD30540
C                                                                       EVD30550
  160 EVD3 = -(F0 + F1) / 12.56637                                      EVD30560
      RETURN                                                            EVD30570
  170 EVD3 = -PART1 / 6.283185                                          EVD30580
      RETURN                                                            EVD30590
      END                                                               EVD30600


*DECK EVD4                                                              EVD40010
      FUNCTION EVD4(X,Y,D1,D2,DELTA)                                    EVD40020
C                                                                       EVD40030
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y               EVD40040
C  DUE TO A HINGE EVD ELEMENT WITH ONE RADIAN TURNING ANGLE             EVD40050
C  LOCATED AT THE ORIGIN 0,0                                            EVD40060
C                                                                       EVD40070
      DIMENSION SI(9),FACTOR(9)                                         EVD40080
      R(A,B) = SQRT(A*A + B*B)                                          EVD40090
      CHANGE(A,B,C) = 0.50 * (C * (B-A) + (A+B))                        EVD40100
      S(A) = ABS(A) / A                                                 EVD40110
      SO(T) = 0.50 * (-D1L/D1*(1.00-S(T)) + D2L/D2*(1.00+S(T)))         EVD40120
      G(A) = ALOG(ABS(A)) - SO(A) * A                                   EVD40130
      DATA SI/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,          EVD40140
     1          0.3242534, 0.6133714, 0.8360311, 0.9681602/             EVD40150
      DATA FACTOR/0.0812744,0.1806482,0.2606107,0.3123471,0.3302394,    EVD40160
     1              0.3123471,0.2606107,0.1806482,0.0812744/            EVD40170
C                                                                       EVD40180
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS.                          EVD40190
      DB = 0.50 * (D1 + D2)                                             EVD40200
   20 XB = X / DB                                                       EVD40210
      XMD = X - D2                                                      EVD40220
      XPD = X + D1                                                      EVD40230
   30 YMD = Y - DELTA                                                   EVD40240
      YPD = Y + DELTA                                                   EVD40250
   40 ROP = R(X,YMD)                                                    EVD40260
      ROPP = R(X,YPD)                                                   EVD40270
      AX = ABS(X)                                                       EVD40280
      AXB = ABS(XB)                                                     EVD40290
      D1L = ALOG(D1)                                                    EVD40300
      D2L = ALOG(D2)                                                    EVD40310
      XD1 = X/D1                                                        EVD40320
      XD2 = X/D2                                                        EVD40330
   50 PART5 = 1.00/YMD - 1.00/YPD                                       EVD40340
      PART6 = -(D1+D2) + 0.50 * (D1*D1L + D2*D2L)                       EVD40350
C                                                                       EVD40360
```

```
C  CALCULATE RK(X)                                                      EVD40370
      IF(AXB .LT. 7.5) GO TO 80                                         EVD40380
      N = 0                                                             EVD40390
      RK = 0.00                                                         EVD40400
      D1X = D1/X                                                        EVD40410
      D2X = D2/X                                                        EVD40420
      DO 70 N1 = 1,100,2                                                EVD40430
      RKN = 0.000                                                       EVD40440
      DO 60 N2 = 1,2                                                    EVD40450
      N = N + 1                                                         EVD40460
      RN = N                                                            EVD40470
      RNN = N * (N+1)                                                   EVD40480
      RKN = RKN + (-1.0)**(N+1) * (D1L/RNN - 1.0/(RN*RN)) * D1X**N      EVD40490
     1       +(D2L/RNN - 1.0/(RN*RN)) * D2X**N                          EVD40500
   60 CONTINUE                                                          EVD40510
      RK = RK + RKN                                                     EVD40520
      IF(ABS(RKN/RK) .LT. 1.0E-07) GO TO 200                            EVD40530
   70 CONTINUE                                                          EVD40540
   80 IF(AXB .LT. 1.0E-04) GO TO 210                                    EVD40550
      SX = S(X)                                                         EVD40560
      RK1 = 0.00                                                        EVD40570
      RK2 = 0.00                                                        EVD40580
      RK3 = 0.00                                                        EVD40590
      RK4 = 0.00                                                        EVD40600
      RK1P = ABS(XD1+1.00)                                              EVD40610
   90 IF(RK1P .LT. 1.0E-06) GO TO 100                                   EVD40620
      RK1 = (ALOG(ABS(XD1)) + (XD1+1.00) * D1L) * ALOG(ABS(XPD/X))      EVD40630
  100 RK2P = ABS(XD2-1.00)                                              EVD40640
      IF(RK2P .LT. 1.0E-06) GO TO 110                                   EVD40650
      RK2 = -(ALOG(ABS(XD2)) - (XD2-1.00) * D2L) * ALOG(ABS(XMD/X))     EVD40660
  110 IF(ABS(D1/AX-1.00) .LT. 1.0E-04) GO TO 150                        EVD40670
C  CALCULATE RK3 BY GAUSSIAN INTEGRATION.                               EVD40680
      AL = 1.00                                                         EVD40690
      BL = D1/AX                                                        EVD40700
      DO 140 N = 1,9                                                    EVD40710
  120 T = CHANGE(AL,BL,SI(N))                                           EVD40720
  130 RK3 = RK3 + FACTOR(N) * (ALOG(T) / (SX+T))                        EVD40730
  140 CONTINUE                                                          EVD40740
      RK3 = 0.50 * (BL-AL) * RK3                                        EVD40750
C  CALCULATE RK4 BY GAUSSIAN INTEGRATION.                               EVD40760
  150 IF(ABS(D2/AX-1.00) .LT. 1.0E-04) GO TO 190                        EVD40770
      AL = 1.00                                                         EVD40780
      BL = D2/AX                                                        EVD40790
      DO 180 N = 1,9                                                    EVD40800
  160 T = CHANGE(AL,BL,SI(N))                                           EVD40810
  170 RK4 = RK4 + FACTOR(N) * (ALOG(T) / (SX-T))                        EVD40820
  180 CONTINUE                                                          EVD40830
      RK4 = 0.50 * (BL-AL) * RK4                                        EVD40840
  190 RK = -2.467401 * SX - (D1L-D2L) + (RK1 + RK2) + (RK3 + RK4)       EVD40850
  200 P = (ROP/YMD - ROPP/YPD) * RK                                     EVD40860
      GO TO 220                                                         EVD40870
  210 P = 0.00                                                          EVD40880
C                                                                       EVD40890
C  CALCULATE F(X) BY GAUSSIAN INTEGRATION.                              EVD40900
  220 IF((X .GT. -D1) .AND. (X .LT. D2)) GO TO 300                      EVD40910
C  X IS NOT WITHIN THE DIMENSIONS OF THE ELEMENT.                       EVD40920
C  LEFT SIDE INTEGRAL.                                                  EVD40930
      FL = 0.00                                                         EVD40940
      AL = -D1                                                          EVD40950
      BL = 0.00                                                         EVD40960
```

```
      DO 250 N = 1,9                                                    EVD40970
  230 SY = CHANGE(AL,BL,SI(N))                                          EVD40980
      XMS = X - SY                                                      EVD40990
      GS = G(SY)                                                        EVD41000
  240 PART1 = GS * (R(XMS,YMD)-ROP) / YMD - GS * (R(XMS,YPD)-ROPP) / YPDEVD41010
      FL = FL + FACTOR(N) * PART1 / XMS                                 EVD41020
  250 CONTINUE                                                          EVD41030
      FL = 0.50 * (BL-AL) * FL                                          EVD41040
C  RIGHT SIDE INTEGRAL                                                  EVD41050
      FR = 0.00                                                         EVD41060
      AL = 0.00                                                         EVD41070
      BL = D2                                                           EVD41080
      DO 280 N = 1,9                                                    EVD41090
  260 SY = CHENGE(AL,BL,S (N))                                          EVD41100
      XMS = X - SY                                                      EVD51115
      GS = G(SY)                                                        EVD41120
  270 PART1 = GS * (R(XMS,YMD)-ROP) / YMD - GS * (R(XMS,YPD)-ROPP) /  PDEVD41130
      FR = FR + FACTOR(N) * PART1 / XMS                                 EVD41140
  280 CONTINUE                                                          EVD41150
      FR = 0.50 * (BL-AL) * FR                                          EVD41160
  290 F = FL + FR                                                       EVD41170
      GO TO 470                                                         EVD41180
C                                                                       EVD41190
C  X IS WITHIN THE DIMENSIONS OF THE ELEMENT                            EVD41200
C  CALCULATE F0                                                         EVD41210
  300 F0 = 0.00                                                         EVD41220
      GPX = 0.00                                                        EVD41230
      IF(AXB .LT. 1.0E-04) GO TO 320                                    EVD41250
      GPPX = 0.00                                                       EVD41240
      SDX = SO(X) - 1.00/X                                              EVD41260
      GX = G(X)                                                         EVD41270
      GPX = GX * (ABS(YMD)-ROP)                                         EVD41280
      GPPX = GX * (ABS(YPD)-ROPP)                                       EVD41290
      IF(((1.00-XD2) .LT. 1.0E-06) .OR. ((XD1+1.00) .LT. 1.0E-06))      EVD41300
     1          GO TO 320                                               EVD41310
  310 F0 = -(GPX/YMD - GPPX/YPD) * ALOG(ABS(XMD/XPD))                   EVD41320
C                                                                       EVD41330
C  CALCULATE F1                                                         EVD41340
C  LEFT SIDE INTEGRAL                                                   EVD41350
  320 FL = 0.00                                                         EVD41360
      AL = -D1                                                          EVD41370
      BL = 0.00                                                         EVD41380
      DO 380 N = 1,9                                                    EVD41390
  330 SY = CHANGE(AL,BL,SI(N))                                          EVD41400
      XMS = X - SY                                                      EVD41410
      IF(ABS(XMS/DB) .LT. 1.0E-04) GO TO 360                            EVD41420
      GS = G(SY)                                                        EVD41430
  340 PART2 = (GS * (R(XMS,YMD)-ROP) - GPX) / XMS                       EVD41440
  350 PART3 = (GS * (R(XMS,YPD)-ROPP) - GPPX) / XMS                     EVD41450
      PART4 = PART2/YMD - PART3/YPD                                     EVD41460
      GO TO 370                                                         EVD41470
  360 PART2 = ROP/YMD - ROPP/YPD                                        EVD41480
      PART4 = (S(YMD) - S(YPD) - PART2) * SDX                           EVD41490
  370 FL = FL + FACTOR(N) * PART4                                       EVD41500
  380 CONTINUE                                                          EVD41510
      FL = 0.50 * (BL-AL) * FL                                          EVD41520
C                                                                       EVD41530
C  RIGHT SIDE INTEGRAL                                                  EVD41540
  390 FR = 0.00                                                         EVD41550
      AL = 0.00                                                         EVD41560
      BL = D2                                                           EVD41570
      DO 450 N = 1,9                                                    EVD41580
  400 SY = CHANGE(AL,BL,SI(N))                                          EVD41590
      XMS = X - SY                                                      EVD41600
      IF(ABS(XMS/DB) .LT. 1.0E-04) GO TO 430                            EVD41610
      GS = G(SY)                                                        EVD41620
  410 PART2 = (GS * (R(XMS,YMD)-ROP) - GPX) / XMS                       EVD41630
  420 PART3 = (GS * (R(XMS,YPD)-ROPP) - GPPX) / XMS                     EVD41640
      PART4 = PART2/YMD - PART3/YPD                                     EVD41650
      GO TO 440                                                         EVD41660
  430 PART2 = ROP/YMD - ROPP/YPD                                        EVD41670
      PART4 = (S(YMD) - S(YPD) - PART2) * SDX                           EVD41680
  440 FR = FR + FACTOR(N) * PART4                                       EVD41690
  450 CONTINUE                                                          EVD41700
      FR = 0.50 * (BL-AL) * FR                                          EVD41710
  460 F = FL + FR + F0                                                  EVD41720
C                                                                       EVD41730
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT                         EVD41740
  470 EVD4 = (PART5 * PART6 + P + F) / 19.739202                        EVD41750
      RETURN                                                            EVD41760
      END                                                               EVD41770
```

```
*DECK GROUND                                                            GRND0010
      SUBROUTINE GROUND(X,Y,ZZ,D1,D2,DELTA,IT,DW,DU)                    GRND0020
C                                                                       GRND0030
C     THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT (X,Y) ON THE   GRND0040
C     WING DUE TO A SEMI-INFINITE HORSESHOE VORTEX AT (0,0,-2*Z) OF     GRND0050
C     UNIT EVD PEAK VORTICITY.                                          GRND0060
      REAL NB,NTL,NTR                                                   GRND0070
      R2(A,B) = SQRT(A**2+B**2)                                         GRND0080
      R(A,B,C) = SQRT(A**2+B**2+C**2)                                   GRND0090
C     DETERMINE THE RELATIONSHIP BETWEEN EVD STRENGTH AND VORTEX LATTICEGRND0100
C     STRENGTH                                                          GRND0110
      GO TO ( 20 , 30 , 40 , 20 ),IT                                    GRND0120
   20 GAMMA = 0.5 * (D1+D2)                                             GRND0130
      GO TO 50                                                          GRND0140
   30 GAMMA = D2                                                        GRND0150
      GO TO 50                                                          GRND0160
   40 GAMMA = 0.5 * D1 + D2                                             GRND0170
   50 YMD = Y + DELTA                                                   GRND0180
      YPD = Y - DELTA                                                   GRND0190
      RL = R(X,YMD,ZZ)                                                  GRND0200
      RR = R(X,YPD,ZZ)                                                  GRND0210
      RB = R2(X,ZZ)                                                     GRND0220
      RTL = R2(YMD,ZZ)                                                  GRND0230
      RTR = R2(YPD,ZZ)                                                  GRND0240
      CT1B = YMD/RL                                                     GRND0250
      CT2B = YPD/RR                                                     GRND0260
      CTL = X/RL                                                        GRND0270
      CTR = X/RR                                                        GRND0280
      QB =  (CT1B - CT2B)/RB                                            GRND0290
      QTL = (CTL + 1.0)/RTL                                             GRND0300
      QTR= (CTR+1.0)/RTR                                                GRND0310
      NB=X/RB                                                           GRND0320
      NTL=YMD/RTL                                                       GRND0330
      NTR=-YPD/RTR                                                      GRND0340
      DW = -(QB*NB + QTL*NTL + QTR*NTR) * GAMMA / 12.5663706            GRND0350
      DU = (-ZZ/RB) * QB * GAMMA / 12.5663706                           GRND0360
      RETURN                                                            GRND0370
      END                                                               GRND0380
```

```
*DECK SHUFL1                                                            SHF10010
      SUBROUTINE SHUFL1(W,ISIZE,IGND,IO,ISTAB)                          SHF10020
C                                                                       SHF10030
C  THIS SUBROUTINE READS THE PORTION OF THE DOWNWASH MATRIX WHICH       SHF10040
C  CONTAINS THE DOWNWASH DUE TO THE JET, AUGMENTS IT ACCORDING TO       SHF10050
C  THE CURRENT CMU VALUES, AND WRITES IT BACK ON UNIT 1                 SHF10060
C  BEHIND THE DOWNWASH MATRIX                                           SHF10070
C                                                                       SHF10080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            SHF10090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) SHF10100
      COMMON/GEOMI/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  SHF10110
     1              D(40),KK(400),ITYPE(400)                            SHF10120
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           SHF10130
      DIMENSION A(400),AIM1(400),W(ISIZE)                               SHF10140
      LOGICAL PRINT                                                     SHF10150
C                                                                       SHF10160
      PRINT = IPRINT .LT. 0                                             SHF10170
      IREAD = 2*NWT + 1                                                 SHF10180
      IWRITE = 2 * NMAX + 1                                             SHF10190
      IF(ISTAB .EQ. 0) GO TO 20                                         SHF10200
      IREAD = IREAD + 1                                                 SHF10210
      IWRITE = IWRITE + 1                                               SHF10220
   20 CONTINUE                                                          SHF10230
      NWT1 = NWT + 1                                                    SHF10240
      IF(IPRINT .GE. 0) GO TO 50                                        SHF10250
      WRITE(6, 30 )                                                     SHF10260
   30 FORMAT(1H1,42X,36HAUGMENTED PORTION OF SOLUTION MATRIX/)          SHF10270
      IF(IGND .NE. 0) WRITE(6, 40 )                                     SHF10280
   40 FORMAT(1H ,51X,19HWITH GROUND EFFECTS/)                           SHF10290
C  PREPARE THE SOLUTION MATRIX FOR ROWS ON THE JET                      SHF10300
C       I IS THE COUNTER FOR IDENTIFYING ELEMENTS ON THE JET            SHF10310
C   IREAD IS THE COUNTER FOR DOWNWASH ROWS ON THE JET STORED ON UNIT 10 SHF10320
C  IWRITE IS THE COUNTER FOR AUGMENTED ROWS TO BE WRITTEN ON UNIT 10    SHF10330
   50 IREAD = IREAD - 2                                                 SHF10340
      IWRITE = IWRITE - 2                                               SHF10350
      C1 = 0.125000                                                     SHF10360
      C3 = 0.375000                                                     SHF10370
      DO 180 I = NWT1,NMAX                                              SHF10380
   60 IM1 = I - 1                                                       SHF10390
      IP1 = I + 1                                                       SHF10400
      K = KK(I)                                                         SHF10410
      IREAD = IREAD + 2                                                 SHF10420
      IWRITE = IWRITE + 2                                               SHF10430
C  READ THE ITH ROW OF THE DOWNWASH MATRIX  (IREADTH RECORD)            SHF10440
   70 CALL READMS(IO,W,ISIZE,IREAD)                                     SHF10450
C  SAVE THE EXISTING ROW OF SIMPLE DOWNWASH COEFFICIENTS                SHF10460
      DO 80 J = 1,NMAX                                                  SHF10470
      A(J) = W(J)                                                       SHF10480
   80 CONTINUE                                                          SHF10490
C  SUBTRACT THE PREVIOUS ROW FROM THE PRESENT ROW IF THE DOWNWASH POINT SHF10500
C  IS NOT ON A LEADING JET ELEMENT                                      SHF10510
      PROD1 = CMUP(K) * DEL(I) * CHORD(K)                               SHF10520
      PROD2 = CMUP(K) * DEL(IM1) * CHORD(K)                             SHF10530
   90 IF(I .EQ. IJ(K)) GO TO 120                                        SHF10540
      DO 100 J = 1,NMAX                                                 SHF10550
      W(J) = W(J) - AIM1(J)                                             SHF10560
  100 CONTINUE                                                          SHF10570
C  MODIFY THE TWO OR THREE SPECIAL ELEMENTS FURTHER                     SHF10580
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 130                              SHF10590
C  DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT                   SHF10600
  110 W(IM1) = W(IM1) + C1 * PROD2                                      SHF10610
      W(I) = W(I) + C3 * (PROD1 + PROD2)                                SHF10620
      W(IP1) = W(IP1) + C1 * PROD1                                      SHF10630
      GO TO 140                                                         SHF10640
C  DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT                   SHF10650
  120 W(I) = W(I) + C3 * PROD1                                          SHF10660
      W(IP1) = W(IP1) + C1 * PROD1                                      SHF10670
      GO TO 140                                                         SHF10680
C  DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT                  SHF10690
  130 W(IM1) = W(IM1) + C1 * PROD2                                      SHF10700
      W(I) = W(I) + C3 * PROD2 + CMUP(K) * D(K)                         SHF10710
C  STORE THE AUGMENTED ITH ROW ON UNIT 1  (IWRITETH RECORD)             SHF10720
  140 CALL WRITMS(IO,W,ISIZE,IWRITE)                                    SHF10730
C  PRINT THE AUGMENTED PORTION OF THE MATRIX                            SHF10740
      IF(PRINT) WRITE(6, 150 ) I,W                                      SHF10750
  150 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))                  SHF10760
C  SAVE THE ITH ROW FOR USE AS THE I-1 ROW ON THE NEXT PASS             SHF10770
  160 DO 170 J = 1,NMAX                                                 SHF10780
      AIM1(J) = A(J)                                                    SHF10790
  170 CONTINUE                                                          SHF10800
  180 CONTINUE                                                          SHF10810
      RETURN                                                            SHF10820
C  DIRECT ACCESS UNITS NOW CONTAIN THE FOLLOWING ON ALTERNATING RECORDS SHF10830
C    WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NWT RECORDS)       SHF10840
C     JET-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NJT RECORDS)       SHF10850
C     JET-DUE-TO-WING-AND-JET AUGMENTED DOWNWASH COEFFICIENTS (NJT RECS)SHF10860
      END                                                               SHF10870
```

```
*DECK SHUFL2                                                            SHF20010
      SUBROUTINE SHUFL2(W,ISIZE,IGND,IO,ISTAB)                          SHF20020
C                                                                       SHF20030
C  THIS SUBROUTINE READS EACH MATRIX ROW CORRESPONDING TO A DOWNWASH    SHF20040
C  CONTROL POINT ON THE JET, MODIFIES IT ACCORDING TO THE NEW VALUES    SHF20050
C  OF CMU, AND RESTORES IT IN ITS ORIGINAL PLACE                        SHF20060
C                                                                       SHF20070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            SHF20080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) SHF20090
      COMMON/GEOMI/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  SHF20100
     1              D(40),KK(400),ITYPE(400)                            SHF20110
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           SHF20120
      DIMENSION W(ISIZE)                                                SHF20130
      LOGICAL PRINT                                                     SHF20140
C                                                                       SHF20150
      PRINT = IPRINT .LT. 0                                             SHF20160
      IREAD = 2 * NMAX + 1                                              SHF20170
      IF(ISTAB .NE. 0) IREAD = IREAD + 1                                SHF20180
      IF(IPRINT .GE. 0) GO TO 40                                        SHF20190
      WRITE(6, 20 )                                                     SHF20200
   20 FORMAT(1H1,42X,36HAUGMENTED PORTION OF SOLUTION MATRIX)           SHF20210
      IF(IGND .NE. 0) WRITE(6, 30 )                                     SHF20220
```

```
   30 FORMAT(1H ,51X,19HWITH GROUND EFFECTS/)                           SHF20230
C  CYCLE THE AUGMENTED MATRIX ROWS                                      SHF20240
   40 IREAD = IREAD - 2                                                 SHF20250
      NWT1 = NWT + 1                                                    SHF20260
      DO 120 I = NWT1,NMAX                                              SHF20270
      IREAD = IREAD + 2                                                 SHF20280
      IM1 = I - 1                                                       SHF20290
      IP1 = I + 1                                                       SHF20300
      K = KK(I)                                                         SHF20310
   50 CMUDIF = (CMUP(K) - CMUPP(K)) * CHORD(K)                          SHF20320
C  READ THE ITH AUGMENTED MATRIX ROW                                    SHF20330
   60 CALL READMS(10,W,ISIZE,IREAD)                                     SHF20340
C  MODIFY THE TWO OR THREE SPECIAL ELEMENTS ACCORDING TO THE NEW CMU    SHF20350
      IF(I .EQ. IJ(K)) GO TO 80                                         SHF20360
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 90                               SHF20370
C  DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT                   SHF20380
   70 W(IM1) = W(IM1) + 0.1250 * DEL(IM1) * CMUDIF                      SHF20390
      W(I) = W(I) + 0.3750 * (DEL(IM1)+DEL(I)) * CMUDIF                 SHF20400
      W(IP1) = W(IP1) + 0.1250 * DEL(I) * CMUDIF                        SHF20410
      GO TO 100                                                         SHF20420
C  DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT                   SHF20430
   80 W(I) = W(I) + 0.3750 * DEL(I) * CMUDIF                            SHF20440
      W(IP1) = W(IP1) + 0.1250 * DEL(I) * CMUDIF                        SHF20450
      GO TO 100                                                         SHF20460
C  DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT                  SHF20470
   90 W(IM1) = W(IM1) + 0.1250 * DEL(IM1) * CMUDIF                      SHF20480
      W(I) = W(I) + (0.3750*DEL(IM1) + D(K)/CHORD(K)) * CMUDIF          SHF20490
C  WRITE THE REVISED ITH ROW ON UNIT 10                                 SHF20500
  100 CALL WRITMS(10,W,ISIZE,IREAD)                                     SHF20510
      IF(PRINT) WRITE(6, 110 ) I,W                                      SHF20520
  110 FORMAT(1H0,55X,10HMATRIX ROW,I4,60(/1X,10E13.5))                  SHF20530
  120 CONTINUE                                                          SHF20540
      RETURN                                                            SHF20550
      END                                                               SHF20560
```

```
*DECK COLUM1                                                            COL10010
      SUBROUTINE COLUM1(LCASE)                                          COL10020
C                                                                       COL10030
C  THIS SUBROUTINE SETS UP THE RIGHT SIDE COLUMN MATR X WITHOUT         COL10040
C  GONS DERATION OF ANY HINGE DOWNWASH  NFLVENCE                        COL10050
C                                                                       COL10060
      COMMON/MARK/N OWS?NROWSJ,NWT,NJT,NMAX?NW(40),NJ(45),IW(40),IJ(40) COL10070
      COMMON/GEOM1/Y(40),CHORE(40),DELTA(40),XB(400),XI(400),EEL(400),  CPL10080
     1                 E(40),KK)400),IT PE(400)                         COL10090
      COMMON/GCASE3/EPS(400,15),BETA(400,10),THETE(40,10),THS(40,10)    CPL10100
      COMMON/SOPV1/G(401,10)                                            COL10110
C                                                                       CPL10120
C  EEFINE T E EPEMENTS PN 7 E W NG                                      COL10135
      DO 20 I = 1,NWT                                                   COL10140
      B(I,LCAXE) = EPX(I,LCASN) / 57.395779                             CPL10550
   20 CPNTINUE                                                          COL10160
C                                                                       CPL10570
C  EEFINE T E EPEMENTS PN T E JET                                       COL10180
      I = NWT                                                           CPL10190
      DO 50 K = 1,N OWS                                                 COL10200
      NJK = NN(K)                                                       CPL10210
      IG(NJK .EQ. 0) GO TO 50                                           COL10220
```

```
C  FIRST JET ELENENT                                                    COL10230
      I = I + 1                                                         C?L10240
      KKI = KK(I)                                                       COL10250
   30 B(I,LCASE) = THETA(KKI,LCASE) / 57.295779                         COL10260
C  REMAINING JET ELEMENTS                                               COL10270
      DO 40 L = 2,NJK                                                   COL10280
      I = I + 1                                                         COL10290
      B(I,LCASE) = 0.00                                                 COL10300
   40 CONTINUE                                                          COL10310
   50 CONTINUE                                                          COL10320
      RETURN                                                            COL10330
      END                                                               COL10340
```

```
*DECK COLUM2                                                            COL20010
      SUBROUTINE COLUM2(H,ISIZE,NEWMAX,LCASE)                           COL20020
C                                                                       COL20030
C  THIS SUBROUTINE ADDS THE APPROPRIATE HINGE DOWNWASH INFLUENCE        COL20040
C  TO THE RIGHT SIDE COLUMN MATRIX                                      COL20050
C                                                                       COL20060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) COL20070
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  COL20080
     1                 D(40),KK(400),ITYPE(400)                         COL20090
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    COL20100
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           COL20110
      COMMON/SOLV1/B(400,10)                                            COL20120
      DIMENSION H(ISIZE)                                                COL20130
C                                                                       COL20140
C  DEFINE THE ELEMENTS ON THE WING                                      COL20150
      DO 20 I = 1,NEWMAX                                                COL20160
      B(I,LCASE) = B(I,LCASE) - H(I)                                    COL20170
   20 CONTINUE                                                          COL20180
C                                                                       COL20190
C  DEFINE THE ELEMENTS ON THE JET                                       COL20200
      IF(NEWMAX .EQ. NWT) RETURN                                        COL20210
      I = NWT                                                           COL20220
      DO 100 K = 1,NROWS                                                COL20230
   30 NJK = NJ(K)                                                       COL20240
      IF(NJK .EQ. 0) GO TO 100                                          COL20250
C  COMPUTE THE CMU INFLUENCE FACTORS H1 AND H2                          COL20260
      I = I + 1                                                         COL20270
      H1 = 0.00                                                         COL20280
      H2 = 0.00                                                         COL20290
      BTA = BETA(I,LCASE)                                               COL20300
      IF(ABS(BTA) .LT. 0.0001) GO TO 60                                 COL20310
      BTA = BTA / 57.295779                                             COL20320
   40 D2 = DEL(I) * CHORD(K)                                            COL20330
      DL = ALOG(D2)                                                     COL20340
      PROD = -CMUP(K) * D2 * BTA / 3.1415927                            COL20350
   50 H1 = PROD * (1.6931472 - 0.750 * DL)                              COL20360
      H2 = PROD * (0.3068528 - 0.250 * DL)                              COL20370
C  FIRST POINT ON THE JET                                               COL20380
   60 B(I,LCASE) = B(I,LCASE) + H1                                      COL20390
C  SECOND POINT ON THE JET                                              COL20400
      I = I + 1                                                         COL20410
   70 B(I,LCASE) = B(I,LCASE) + H(I-1) + H2                             COL20420
C  REMAINING POINTS ON THE JET                                          COL20430
      IF(NJK .LT. 3) GO TO 100                                          COL20440
      DO 90 L = 3,NJK                                                   COL20450
      I = I + 1                                                         COL20460
   80 B(I,LCASE) = B(I,LCASE) + H(I-1)                                  COL20470
   90 CONTINUE                                                          COL20480
  100 CONTINUE                                                          COL20490
      RETURN                                                            COL20500
      END                                                               COL20510
```

```
*DECK HINGE                                                             HING0010
      SUBROUTINE HINGE(H,ISIZE,NEWMAX,LCASE)                            HING0020
C                                                                       HING0030
C  THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENTS       HING0040
C  AT EACH DOWNWASH CONTROL POINT DUE TO ALL DEFLECTED HINGE ELEMENTS.  HING0050
C  FOR EACH CONTROL POINT THE INFLUENCE COEFFICIENTS ARE MULTIPLIED BY  HING0060
C  THEIR RESPECTIVE DEFLECTION ANGLE AND SUMMED UP TO OBTAIN THE        HING0070
C  COMPLETE HINGE-INDUCED DOWNWASH.                                     HING0080
C                                                                       HING0090
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            HING0100
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) HING0110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  HING0120
     1                 D(40),KK(400),ITYPE(400)                         HING0130
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    HING0140
      DIMENSION H(ISIZE)                                                HING0150
C                                                                       HING0160
      IF(LCASE .GT. 2) GO TO 30                                         HING0170
      IF(IPRINT .LT. 0) WRITE(6, 20 )                                   HING0180
   20 FORMAT(1H1)                                                       HING0190
      ILINES = 1                                                        HING0200
C                                                                       HING0210
C  CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET                HING0220
   30 DO 120 I = 1,NEWMAX                                               HING0230
      KI = KK(I)                                                        HING0240
C                                                                       HING0250
C  CYCLE THE VORTEX POINTS ON THE WING AND JET                          HING0260
      DO 110 J = 1,NEWMAX                                               HING0270
C  CHECK WHETHER THERE IS A DEFLECTED HINGE AT ELEMENT J                HING0280
   40 B = BETA(J,LCASE)                                                 HING0290
      IF(ABS(B) .LT. 0.0001) GO TO 110                                  HING0300
      B = B / 57.295779                                                 HING0310
C  COMPUTE THE GEOMETRIC PARAMETERS                                     HING0320
      KJ = KK(J)                                                        HING0330
   50 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)                         HING0340
      YY = Y(KI) - Y(KJ)                                                HING0350
   60 D2 = DEL(J) * CHORD(KJ)                                           HING0360
      D1 = DEL(J-1) * CHORD(KJ)                                         HING0370
      IN1 = IW(KJ) + NW(KJ) - 1                                         HING0380
   70 IF(ITYPE(J) .EQ. 43) D1 = DEL(IN1) * CHORD(KJ)                    HING0390
C  COMPUTE AND SUM UP THE INFLUENCE OF ELEMENT J                        HING0400
   80 H(I) = H(I) + EVDN(X,YY,D1,D2,DELTA(KJ)) * B                      HING0410
C  SUPERIMPOSE DOWNWASH FOR SYMETRIC OR ANTI-SYMETRIC GEOMETRY          HING0420
      IF(ISYMM .GT. 0) GO TO 110                                        HING0430
      YY = Y(KI) + Y(KJ)                                                HING0440
   90 HDUMMY = EVDN(X,YY,D1,D2,DELTA(KJ))                               HING0450
      IF(ISYMM .LT. 0) HDUMMY = - HDUMMY                                HING0460
  100 H(I) = H(I) + HDUMMY * B                                          HING0470
  110 CONTINUE                                                          HING0480
  120 CONTINUE                                                          HING0490
```

```
C                                                                       HING0500
C  PRINT OUT THE DOWNWASH IF REQUIRED                                   HING0510
      IF(IPRINT .GE. 0) RETURN                                          HING0520
      NEXT = NEWMAX/10 + 3                                              HING0530
      IF(((ILINES+NEXT) .LT. 56) .OR. (ILINES .EQ. 1)) GO TO 130        HING0540
      WRITE(6, 20 )                                                     HING0550
      ILINES = 1                                                        HING0560
  130 IF(IPRINT .LT. 0) WRITE(6, 140 ) LCASE,H                          HING0570
  140 FORMAT(1H0,35X,44HHINGE INFLUENCE COEFFICIENTS FOR FUNDAMENTAL,   HING0580
     1          5H CASE, I3,60(/1X,10E13.5))                            HING0590
      ILINES = ILINES + NEXT                                            HING0600
      RETURN                                                            HING0610
      END                                                               HING0620
```

```
*DECK STAGE3                                                            STG30010
      OVERLAY(OVLY,3,0)                                                 STG30020
      PROGRAM STAGE3                                                    STG30030
C                                                                       STG30040
C  THIS SUBROUTINE CONTROLS CALCULATION OF WING LOADINGS FOR THE        STG30050
C  FUNDAMENTAL AND COMPOSITE CASES                                      STG30060
C                                                                       STG30070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                STG30080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            STG30090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) STG30100
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     STG30110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  STG30120
     1                 D(40),KK(400),ITYPE(400)                         STG30130
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             STG30140
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    STG30150
      COMMON/COMPOS/FACTOR(10,24),NCC                                   STG30160
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           STG30170
      COMMON /DERIV/ UO(40),CLQ,CMQ,CMQMC                               STG30180
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          STG30190
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     STG30200
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,   STG30210
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             STG30220
      COMMON/KEEPEM/CLW(5),CMW(5),CLLW(5),CDW(5),CNW(5),CYW(5),CLQW,    STG30230
     1        CMQW,CLLPW,CLLRW,CNPW,CNRW,CYPW,CYRW,CLH(5),CMH(5),CLLH(5),STG30240
     2        CDH(5),CLQH,CMQH,CLLPH,CYV(5),CYBV,CNV(5),CNBV,CLLV(5),   STG30250
     3        CLLBV,CDV(5),CLLPV,CLLRV,CNPV,CNRV,CYPV,CYRV,CMF(5),CNF(5),STG30260
     4        CMQF,CNRF                                                 STG30270
      LOGICAL GE,NGE,VT                                                 STG30280
      GE = IGND .NE. 0                                                  STG30290
      NGE = IGND .EQ. 0                                                 STG30300
      VT = JETFLG .GT. 1                                                STG30310
C  CHECK FOR GROUND EFFECT WITH NO VECTORED JETS                        STG30320
      IF(GE) GO TO 60                                                   STG30330
C  CHECK FOR VECTORED JETS WITH POWER ON                                STG30340
      IF((VT) .AND. (LOGIC .EQ. 1)) GO TO 80                            STG30350
C  COMPUTE THE REGULAR FUNDAMENTAL CASE CHORDWISE LOADING               STG30360
   20 CALL STG3FC(NEWMAX)                                               STG30370
C  COMPUTE THE REGULAR FUNDAMENTAL CASE SPANWISE AND TOTAL LOADING      STG30380
      DO 40  N = 1,NCASES                                               STG30390
      LCASE = N                                                         STG30400
   30 CALL STG3FS(CLQ,CMQ,CMQMC,DUMM,NEWMAX,NOALFA,LCASE)               STG30410
   40 CONTINUE                                                          STG30420
C  IF FIRST STABILITY RUN, PRINT WING PITCHING RATE DERIVATIVES         STG30430
```

```
      IF(LOGIC .EQ. 2) WRITE(6, 50 ) CLQ,CMQ,CMQML                      STG30440
   50 FORMAT(1H0/ 26X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,   STG30450
     1        17H ABOUT XCG, CLQ =,F10.6 /                              STG30460
     2        14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,STG30470
     3        33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /              STG30480
     4        16X,42HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,        STG30490
     5        35H DUE TO PITCHING ABOUT XCG, CMQML =, F10.6)            STG30500
      CLQW = CLQ                                                        STG30510
      CMQW = CMQMC                                                      STG30520
C  PRINT SUMMARY TABLE FOR FUNDAMENTAL CASE TOTAL LOADINGS              STG30530
      CALL STG3FT                                                       STG30540
C  CALCULATE AND PRINT THE LOADING FOR ALL COMPOSITE CASES              STG30550
   60 IF(NCC .LT. 1)GO TO 90                                            STG30560
      DO 70  M = 1,NCC                                                  STG30570
      KCASE = M                                                         STG30580
      IF(NGE) CALL OVERLAY(4HOVLY,3,1)                                  STG30590
      IF(GE) CALL OVERLAY(4HOVLY,3,2)                                   STG30600
   70 CONTINUE                                                          STG30610
      GO TO 90                                                          STG30620
C  COMPUTE KCASE COMPOSITE LOADING WITH VECTORED JET POWER ON           STG30630
   80 CALL OVERLAY(4HOVLY,3,3)                                          STG30640
   90 CONTINUE                                                          STG30650
      END                                                               STG30660
```

```
*DECK STG3FC                                                            S3FC0010
      SUBROUTINE STG3FC(NEWMAX)                                         S3FC0020
C                                                                       S3FC0030
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE LOADING            S3FC0040
C  FOR FUNDAMENTAL CASES                                                S3FC0050
C                                                                       S3FC0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            S3FC0070
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  S3FC0080
     1             D(40),KK(400),ITYPE(400)                             S3FC0090
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    S3FC0100
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) S3FC0110
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           S3FC0120
      COMMON/SOLV1/CP(400,10)                                           S3FC0130
      DIMENSION XBB(5),CPEXP(5,10)                                      S3FC0140
C                                                                       S3FC0150
C  CALCULATE AND PRINT THE CHORDWISE LOADING OF THE FUNDAMENTAL CASES   S3FC0160
C  INITIALIZE THE UNUSED VALUES OF CP FOR PRINTING                      S3FC0170
      NC1 = NCASES + 1                                                  S3FC0180
      IF(NC1 .GT. 10) GO TO 40                                          S3FC0190
      DO 30 N = NC1,10                                                  S3FC0200
      DO 20 I = 1,NEWMAX                                                S3FC0210
      CP(I,N) = 0.00                                                    S3FC0220
   20 CONTINUE                                                          S3FC0230
   30 CONTINUE                                                          S3FC0240
   40 I = 0                                                             S3FC0250
      II = NWT                                                          S3FC0260
      DO 330 K = 1,NROWS                                                S3FC0270
      IF(IPRINT .GT. 0) GO TO 80                                        S3FC0280
      CALL HEADER(ILINES)                                               S3FC0290
      WRITE(6, 50 )                                                     S3FC0300
   50 FORMAT(1H0,37X,4H****,10(5H*****)/                                S3FC0310
     1        38X,46H*  WING CHORDWISE LOADING FOR ALL FUNDAMENTAL ,    S3FC0320
     2        8HCASES  */38X,4H****,10(5H*****))                        S3FC0330
```

```
      WRITE(6, 60 ) K,Y(K),CHORD(K),(N,N=1,10)                          S3FC0340
   60 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/    S3FC0350
     1        8X,1HI,5X,2HXB,3X,9(5X,4HCASE,I2),5X,4HCASE,I3)           S3FC0360
C  ON THE WING                                                          S3FC0370
      WRITE(6, 70 )                                                     S3FC0380
   70 FORMAT(1H+,4HWING)                                                S3FC0390
   80 NWK = NW(K)                                                       S3FC0400
      DO 110 L = 1,NWK                                                  S3FC0410
      I = I + 1                                                         S3FC0420
      DO 90 N = 1,NCASES                                                S3FC0430
      CP(I,N) = 2.00 * CP(I,N)                                          S3FC0440
   90 CONTINUE                                                          S3FC0450
      IF(IPRINT .LT. 1) WRITE(6, 100 ) I,XB(I),(CP(I,N),N=1,10)         S3FC0460
  100 FORMAT(1H ,I8,11F11.6)                                            S3FC0470
  110 CONTINUE                                                          S3FC0480
      ILINES = ILINES + NWK + 5                                         S3FC0490
C  ON THE JET                                                           S3FC0500
      NJK = NJ(K)                                                       S3FC0510
      IF(CMU(K) .LT. 0.0001) GO TO 150                                  S3FC0520
      IF(IPRINT .LT. 1) WRITE(6, 120 )                                  S3FC0530
  120 FORMAT(1H ,1X,3HJET)                                              S3FC0540
      DO 140 L = 1,NJK                                                  S3FC0550
      II = II + 1                                                       S3FC0560
      DO 130 N = 1,NCASES                                               S3FC0570
      CP(II,N) = 2.00 * CP(II,N)                                        S3FC0580
  130 CONTINUE                                                          S3FC0590
      IF(IPRINT .LT. 1) WRITE(6, 100 ) II,XB(II),(CP(II,N),N=1,10)      S3FC0600
  140 CONTINUE                                                          S3FC0610
      ILINES = ILINES + NJK + 1                                         S3FC0620
C  PRINT THE DETAILED LOADING ON THE SINGULAR ELEMENTS                  S3FC0630
C  LEADING EDGE                                                         S3FC0640
  150 IF(IPRINT .GT. 0) GO TO 330                                       S3FC0650
      WRITE(6, 160 )                                                    S3FC0660
  160 FORMAT(1H0,45X,29HDETAILED LEADING EDGE LOADING)                  S3FC0670
      IP = IW(K)                                                        S3FC0680
      DO 170 N = 1,NCASES                                               S3FC0690
      LCASE = N                                                         S3FC0700
      CALL EXPLE(LCASE,CP(IP,LCASE),CP(IP+1,LCASE),DEL(IP),XBB,CPEXP)   S3FC0710
  170 CONTINUE                                                          S3FC0720
      DO 180 M = 1,5                                                    S3FC0730
      WRITE(6, 100 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)                   S3FC0740
  180 CONTINUE                                                          S3FC0750
      ILINES = ILINES + 7                                               S3FC0760
C                                                                       S3FC0770
C  HINGES                                                               S3FC0780
      IF(IHINGE .EQ. 0) GO TO 330                                       S3FC0790
      J = IW(K) - 1                                                     S3FC0800
      DO 260 L = 1,NWK                                                  S3FC0810
      J = J + 1                                                         S3FC0820
      IF(ITYPE(J) .LT. 40) GO TO 260                                    S3FC0830
      DO 190 N = 1,NCASES                                               S3FC0840
      LCASE = N                                                         S3FC0850
      CALL EXPH1(LCASE,CP(J,N),CP(J-1,N),DEL(J-1),BETA(J,N),            S3FC0860
     1           CHORD(K),XB(J),XBB,CPEXP)                              S3FC0870
  190 CONTINUE                                                          S3FC0880
      IF(ILINES .LT. 46) GO TO 210                                      S3FC0890
      WRITE(6, 200 )                                                    S3FC0900
  200 FORMAT(1H1)                                                       S3FC0910
      ILINES = 1                                                        S3FC0920
  210 WRITE(6, 220 ) J                                                  S3FC0930
```

```
  220 FORMAT(1H0,42X,33HDETAILED HINGE LOADING ON ELEMENT,I4)           S3FC0940
      DO 230 M = 1,5                                                    S3FC0950
      WRITE(6, 100 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)                   S3FC0960
  230 CONTINUE                                                          S3FC0970
      DO 240 N = 1,NCASES                                               S3FC0980
      LCASE = N                                                         S3FC0990
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),              S3FC1000
     1           CHORD(K),XB(J),XBB,CPEXP)                              S3FC1010
  240 CONTINUE                                                          S3FC1020
      DO 250 M = 6,10                                                   S3FC1030
      WRITE(6, 100 ) M,XBB(M-5),(CPEXP(M-5,N),N=1,NCASES)               S3FC1040
  250 CONTINUE                                                          S3FC1050
      ILINES = ILINES + 12                                              S3FC1060
  260 CONTINUE                                                          S3FC1070
      IF((NJ(K) .EQ. 0) .OR.(CMU(K) .LT. 0.0001)) GO TO 330             S3FC1080
      J = IJ(K)                                                         S3FC1090
      IF(ITYPE(J) .NE. 43) GO TO 330                                    S3FC1100
      DO 270 N = 1,NCASES                                               S3FC1110
      LCASE = N                                                         S3FC1120
      II = IW(K) + NW(K) - 1                                            S3FC1130
      CALL EXPH1(LCASE,CP(J,N),CP(II,N),DEL(II),BETA(J,N),              S3FC1140
     1           CHORD(K),XB(J),XBB,CPEXP)                              S3FC1150
  270 CONTINUE                                                          S3FC1160
      IF(ILINES .LT. 46) GO TO 280                                      S3FC1170
      WRITE(6, 200 )                                                    S3FC1180
      ILINES = 1                                                        S3FC1190
  280 WRITE(6, 290 ) J                                                  S3FC1200
  290 FORMAT(1H0,40X,37HDETAILED JET HINGE LOADING ON ELEMENT,I4)       S3FC1210
      DO 300 M = 1,5                                                    S3FC1220
      WRITE(6, 100 ) M,XBB(M),(CPEXP(M,N),N=1,NCASES)                   S3FC1230
  300 CONTINUE                                                          S3FC1240
      DO 310 N = 1,NCASES                                               S3FC1250
      LCASE = N                                                         S3FC1260
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),              S3FC1270
     1           CHORD(K),XB(J),XBB,CPEXP)                              S3FC1280
  310 CONTINUE                                                          S3FC1290
      DO 320 M = 6,10                                                   S3FC1300
      WRITE(6, 100 ) M,XBB(M-5),(CPEXP(M-5,N),N=1,NCASES)               S3FC1310
  320 CONTINUE                                                          S3FC1320
      ILINES = ILINES + 12                                              S3FC1330
  330 CONTINUE                                                          S3FC1340
      RETURN                                                            S3FC1350
      END                                                               S3FC1360
```

```
*DECK STG3FS                                                            S3FS0010
      SUBROUTINE STG3FS(CLQ,CMQ,CMQMC,CLLP,NEWMAX,NOALFA,LC)            S3FS0020
C                                                                       S3FS0030
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL SPANWISE AND TOTAL       S3FS0040
C  LOADING FOR FUNDAMENTAL CASES                                        S3FS0050
C                                                                       S3FS0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            S3FS0070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) S3FS0080
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     S3FS0090
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  S3FS0100
     1             D(40),KK(400),ITYPE(400)                             S3FS0110
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             S3FS0120
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AL(20,40),     S3FS0130
```

```
     1             XHB(4,40),BET(4,40),IFS(4,40),ICI(40),IHT(40),NLT,NHIS3FS0140
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    S3FS0150
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           S3FS0160
      COMMON/SOLV1/CP(400,10)                                           S3FS0170
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          S3FS0180
      COMMON/LOAD1/ TWIST(40),HO(40),TH(40),THETS(40),                  S3FS0190
     1        BTA(400),EP(400),CPD(400)                                 S3FS0200
      COMMON/LOAD2/  CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),  S3FS0210
     1        CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)S3FS0220
      COMMON/LOAD3/  CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),       S3FS0230
     1      CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),    S3FS0240
     2      CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),S3FS0250
     3      CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNJJ(10),CNG(10),S3FS0260
     4      CCY(10),CXCPB(10),CXCLB(10)                                 S3FS0270
      COMMON/LOAD4/  CLGO(40),CLMUO(40),CLO(40),CDMUO(40),CDGO(40),     S3FS0280
     1      CDIO(40),CSO(40),CTO(40),CMGO(40),CMMUO(40),CMTO(40),CMO(40),S3FS0290
     2        XBCPO(40),XBCLO(40),FACT(10)                              S3FS0300
      COMMON/LOAD5/  CGAM(40),CGAMO(40),ALFINF(40),ALFINO(40),DUMB(40)  S3FS0310
      COMMON/FFLO3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         S3FS0320
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               S3FS0330
     2             XTR(40),XTL(40),CHR(40),CHL(40)                      S3FS0340
C                                                                       S3FS0350
   20 N = LC                                                            S3FS0360
      LCASE = LC                                                        S3FS0370
C  LOAD THOSE GEOMETRIC PARAMETERS WHICH ARE DIFFERENT FOR EACH CASE    S3FS0380
C  INTO THEIR RESPECTIVE DUMMY ARRAYS                                   S3FS0390
      DO 40  K = 1,NROWS                                                S3FS0400
      TWIST(K) = TST(K,LCASE)                                           S3FS0410
   30 TH(K) = THETA(K,LCASE)                                            S3FS0420
      THETS(K) = THS(K,LCASE)                                           S3FS0430
      HO(K) = HL(K,LCASE)                                               S3FS0440
      DUMB(K) = 0.00                                                    S3FS0450
   40 CONTINUE                                                          S3FS0460
      DO 60  I = 1,NEWMAX                                               S3FS0470
   50 BTA(I) = BETA(I,LCASE)                                            S3FS0480
      EP(I) = EPS(I,LCASE)                                              S3FS0490
      CPD(I) = CP(I,LCASE)                                              S3FS0500
   60 CONTINUE                                                          S3FS0510
C  COMPUTE SECTIONAL COEFFICIENTS                                       S3FS0520
      ALPHA = 0.00                                                      S3FS0530
      IF((LCASE .EQ. 1) .AND. (NOALFA .NE. 0)) ALPHA = 1.00             S3FS0540
   70 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,               S3FS0550
     1        XB,DEL,BTA,EP,CPD,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,  S3FS0560
     2        CDI,CDMU,CDG,CS,CTO,NROWS,IHINGE)                         S3FS0570
C  COMPUTE SECTIONAL VORTICITY OF WING-JET SYSTEM                       S3FS0580
      CALL SLOADG(CPD,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)  S3FS0590
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY                               S3FS0600
   80 CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)                  S3FS0610
      DO 90 I=1,NROWS                                                   S3FS0620
   90 AINF(I,LCASE) = ALFINF(I)                                         S3FS0630
      IF(IFFLD .GT. 0) WRITE(7, 100 )(ALFINF(I),I = 1,NROWS)            S3FS0640
  100 FORMAT(8F10.6)                                                    S3FS0650
C  COMPUTE TOTAL LINEAR COEFFICIENTS                                    S3FS0660
  110 CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,HO,XB,XLEAD,BTA,     S3FS0670
     1        CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG(N),CCLJ(N),CCL(N),CCMG(N), S3FS0680
     2        CCMJ(N),CCMT(N),CCM(N),CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),S3FS0690
     3        CXCP(N),CXCL(N),CXCPB(N),CXCLB(N),CCJ(N),CLLG(N),CLLJ(N), S3FS0700
     4        CLL(N),ITYPE,IW,NW,NROWS,ISYMM)                           S3FS0710
C  COMPUTE TOTAL NONLINEAR COEFFICIENTS                                 S3FS0720
      LOGIC = 0                                                         S3FS0730
```

```
  120 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,DUMB,          S3FS0740
     1 CCDG(N),CCDJ(N),CCS(N),CCDI(N),CDITZ(N),DUMB,ALFINF,DUMMY,CCJ(1),S3FS0750
     2      CNJ(N),CNJJ(N),CNG(N),CCY(N),XLEAD,TANLE,XMC,NROWS,ISYMM,     S3FS0760
     3      LOGIC)                                                        S3FS0770
C  DEFINE THE CONSTANT STABILITY DERIVATIVES                               S3FS0780
      CLQ = CCLG(NCASES)                                                  S3FS0790
      CMQ = CCMG(NCASES)                                                  S3FS0800
      CMQMC = CMGMC(NCASES)                                               S3FS0810
      CLLP = CLLG(NCASES)                                                 S3FS0820
C  PRINT LIFT AND DRAG COEFFICIENTS                                        S3FS0830
      IF(IPRINT .GT. 1) RETURN                                            S3FS0840
      CALL HEADER(ILINES)                                                 S3FS0850
      WRITE(6, 130 ) LCASE                                                S3FS0860
  130 FORMAT(1H0,39X,1H*,10(5H*****)/                                     S3FS0870
     1        40X,45H*  WING SPANWISE LOADING FOR FUNDAMENTAL CASE,I3,    S3FS0880
     2        3H  */40X,1H*,10(5H*****))                                  S3FS0890
      WRITE(6, 140 )                                                      S3FS0900
  140 FORMAT(1H ,19X,29H............ LIFT .............,5X,               S3FS0910
     1        2H..,7(4H....),14H INDUCED DRAG , 8(4H....)/                S3FS0920
     2        1X, 7HSECTION, 5X,1HY,8X,3HCLG,7X,4HCLMU,6X,2HCL,5X,        S3FS0930
     3        4H  * ,3X,3HCDG,8X,4HCDMU,7X,2HCS,9X,2HCD,9X,3HCMU,         S3FS0940
     4        8X,5HGAMMA,6X,5HALF(N))                                     S3FS0950
      WRITE(6, 150 ) (K,Y(K),CLG(K),CLMU(K),CL(K),CDG(K),CDMU(K),        S3FS0960
     1                 CS(K),CDI(K),CMU(K),CGAM(K),ALFINF(K),K=1,NROWS)   S3FS0970
  150 FORMAT(1H ,I4, 4X, 4F10.6, 4H  * ,7F11.7)                          S3FS0980
      WRITE(6, 160 ) CCLG(N),CCLJ(N),CCL(N),CCDG(N),CCDJ(N),CCS(N),       S3FS0990
     1           CCDI(N),CCJ(N),CDITZ(N)                                  S3FS1000
  160 FORMAT(1H , 8X, 4(10H----------),4H  * ,7(11H-----------)/          S3FS1010
     1        12X,5HTOTAL,2X,3F10.6,4H  * ,5F11.7,11X,F11.7)             S3FS1020
C  PRINT PITCHING MOMENT AND CENTER OF LIFT DATA                           S3FS1030
      IF(NROWS .GT. 21) WRITE(6, 170 )                                    S3FS1040
  170 FORMAT(1H1)                                                         S3FS1050
      WRITE(6, 180 )                                                      S3FS1060
  180 FORMAT(1H0,24X,39H........... PITCHING MOMENT ...........,10X,     S3FS1070
     1        20H ... LIFT CENTER .../                                    S3FS1080
     2        6X, 7HSECTION,5X,1HY,8X,3HCMG,7X,4HCMMU,6X,3HCMT,7X,2HCM,   S3FS1090
     3        5X,10H   *  *   ,3X,5HXCP/C,5X,5HXCL/C)                     S3FS1100
      WRITE(6, 190 ) (K,Y(K),CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)S3FS1110
     1,        K=1,NROWS)                                                 S3FS1120
  190 FORMAT(1H ,I9,4X,5F10.6,10H   *  *   ,2F10.6)                      S3FS1130
      WRITE(6, 200 ) CCMG(N),CCMJ(N),CCMT(N),CCM(N),CXCP(N),CXCL(N),      S3FS1140
     1         CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),CXCPB(N),CXCLB(N)       S3FS1150
  200 FORMAT(1H ,13X, 5(10H----------),10H   *  *   , 2(10H----------)/ S3FS1160
     1       17X,5HTOTAL,2X, 4F10.6, 10H  (APEX)  ,2F10.6,9H (X/CREF)/    S3FS1170
     2       24X,4F10.6,10H  (XMC)   ,2F10.6,8H (Y/B/2))                  S3FS1180
      RETURN                                                              S3FS1190
      END                                                                 S3FS1200


*DECK STG3FT                                                              S3FT0010
      SUBROUTINE STG3FT                                                   S3FT0020
C                                                                         S3FT0030
C  THIS SUBROUTINE PRINTS A TABLE OF ALL TOTAL COEFFICIENTS               S3FT0040
C  FOR ALL FUNDAMENTAL CASES                                              S3FT0050
C                                                                         S3FT0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE              S3FT0070
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),          S3FT0080
     1       CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),     S3FT0090
     2       CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),S3FT0100
     3       CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNJJ(10),CNG(10),S3FT0110
     4       CCY(10),CXCPB(10),CXCLB(10)                                  S3FT0120
      COMMON/LOAD6/ COEFF(27)                                             S3FT0130
C                                                                         S3FT0140
C  INITIALIZE ALL UNUSED TOTAL COEFFICIENTS TO ZERO FOR PRINTING          S3FT0150
      IF(NCASES .EQ. 10) GO TO 30                                         S3FT0160
      NC1 = NCASES + 1                                                    S3FT0170
      DO 20  N = NC1,10                                                   S3FT0180
       CCLG(N) = 0.00                                                     S3FT0190
       CCLJ(N) = 0.00                                                     S3FT0200
        CCL(N) = 0.00                                                     S3FT0210
       CCDG(N) = 0.00                                                     S3FT0220
       CCDJ(N) = 0.00                                                     S3FT0230
        CCS(N) = 0.00                                                     S3FT0240
       CCDI(N) = 0.00                                                     S3FT0250
      CDITZ(N) = 0.00                                                     S3FT0260
        CCJ(N) = 0.00                                                     S3FT0270
       CCMG(N) = 0.00                                                     S3FT0280
       CCMJ(N) = 0.00                                                     S3FT0290
       CCMT(N) = 0.00                                                     S3FT0300
        CCM(N) = 0.00                                                     S3FT0310
       CXCP(N) = 0.00                                                     S3FT0320
       CXCL(N) = 0.00                                                     S3FT0330
      CXCPB(N) = 0.00                                                     S3FT0340
      CXCLB(N) = 0.00                                                     S3FT0350
      CMGMC(N) = 0.00                                                     S3FT0360
      CMJMC(N) = 0.00                                                     S3FT0370
      CMTMC(N) = 0.00                                                     S3FT0380
       CMMC(N) = 0.00                                                     S3FT0390
       CLLG(N) = 0.00                                                     S3FT0400
       CLLJ(N) = 0.00                                                     S3FT0410
        CLL(N) = 0.00                                                     S3FT0420
        CNJ(N) = 0.00                                                     S3FT0430
        CNG(N) = 0.00                                                     S3FT0440
        CCY(N) = 0.00                                                     S3FT0450
   20 CONTINUE                                                            S3FT0460
   30 CALL HEADER(ILINES)                                                 S3FT0470
      WRITE(6, 40 ) (N,N=1,10)                                            S3FT0480
   40 FORMAT(1H0,44X,1H*,8(5H*****)/                                      S3FT0490
     1        45X,41H*  WING TOTAL AERODYNAMIC COEFFICIENTS  */           S3FT0500
     2        45X,1H*,8(5H*****)//14X,9(4HCASE,I2,5X),4HCASE,I3)          S3FT0510
   50 FORMAT(1H0,2X,A8,10F11.7// 26(3X,A8,10F11.7//))                     S3FT0520
      WRITE(6, 50 )COEFF( 1),  CCLG,  COEFF( 2),  CCLJ,  COEFF( 3),   CCL, S3FT0530
     1               COEFF( 4),  CCDG,  COEFF( 5),  CCDJ,  COEFF( 6),   CCS, S3FT0540
     2               COEFF( 7),  CCDI,  COEFF( 8), CDITZ,  COEFF( 9),   CCJ, S3FT0550
     3               COEFF(10),  CCMG,  COEFF(11),  CCMJ,  COEFF(12),  CCMT, S3FT0560
     4               COEFF(13),   CCM,  COEFF(14),  CXCP,  COEFF(15),  CXCL, S3FT0570
     5               COEFF(16), CXCPB,  COEFF(17), CXCLB,  COEFF(18), CMGMC, S3FT0580
     6               COEFF(19), CMJMC,  COEFF(20), CMTMC,  COEFF(21),  CMMC, S3FT0590
     7               COEFF(22),  CLLG,  COEFF(23),  CLLJ,  COEFF(24),   CLL, S3FT0600
     8               COEFF(25),   CNJ,  COEFF(26),   CNG,   COEFF(27),  CCY  S3FT0610
   60 RETURN                                                              S3FT0620
      END                                                                 S3FT0630


*DECK DATA                                                                DATA0010
      BLOCK DATA BLKDAT                                                   DATA0020
      COMMON/LOAD6/ COEFF(27)                                             DATA0030
      DATA COEFF/8H    CCLG,8H    CCLJ,8H**   CCL,8H    CCDG,8H    CCDJ,DATA0040
     1           8H     CCS,8H     CCD,8H** CDITZ,8H**   CCJ,8H    CCMG,DATA0050
     2           8H    CCMJ,8H    CCMT,8H     CCM,8H    CXCP,8H    CXCL,DATA0060
     3           8H   CXCPB,8H   CXCLB,8H  CCMGMC,8H  CCMJMC,8H  CCMTMC,DATA0070
     4           8H** CCMMC,8H    CLLG,8H    CLLJ,8H*    CLL,8H     CNJ,DATA0080
     5           8H*  CNGMC,8H*    CCY/                                   DATA0090
      END                                                                 DATA0100


*DECK EXPLE                                                               EXLE0010
      SUBROUTINE EXPLE(LCASE,CPI,CPI1,DEL,XBB,CPEXP)                      EXLE0020
C                                                                         EXLE0030
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF A LEADING EDGE EVD            EXLE0040
C  AT 5 INTERMEDIATE POINTS ON THE ELEMENT                                EXLE0050
C                                                                         EXLE0060
      DIMENSION XBB(5),CPEXP(5,10)                                        EXLE0070
C                                                                         EXLE0080
   20 DN = 0.20                                                           EXLE0090
      DO 50 N = 1,5                                                       EXLE0100
   30 X = DN * N                                                          EXLE0110
      XBB(N) = X * DEL                                                    EXLE0120
   40 CPEXP(N,LCASE) = 0.666666*CPI*(1.0/SQRT(X)-X) + CPI1*X              EXLE0130
   50 CONTINUE                                                            EXLE0140
      RETURN                                                              EXLE0150
      END                                                                 EXLE0160


*DECK EXPH1                                                               EXH10010
      SUBROUTINE EXPH1(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)             EXH10020
C                                                                         EXH10030
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE FORWARD HALF OF A HINGE  EXH10040
C  EVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT                            EXH10050
C                                                                         EXH10060
      DIMENSION XBB(5),CPEXP(5,10)                                        EXH10070
C                                                                         EXH10080
      DN = 0.20                                                           EXH10090
      DO 40  N = 1,5                                                      EXH10100
   20 X = XB - DEL + (N-1)*DN*DEL                                         EXH10110
      DX = X - XB                                                         EXH10120
      XBB(N) = X                                                          EXH10130
   30 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(-C*DX)               EXH10140
     1        + ALOG(C*DEL)*DX/DEL) +(CPI + (CPI-CPI1)*DX/DEL)            EXH10150
   40 CONTINUE                                                            EXH10160
      RETURN                                                              EXH10170
      END                                                                 EXH10180


*DECK EXPH2                                                               EXH20010
      SUBROUTINE EXPH2(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)             EXH20020
C                                                                         EXH20030
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE REAR HALF OF A HINGE     EXH20040
C  EVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT                            EXH20050
C                                                                         EXH20060
      DIMENSION XBB(5),CPEXP(5,10)                                        EXH20070
C                                                                         EXH20080
      DN = 0.20                                                           EXH20090
      DO 40  N = 1,5                                                      EXH20100
   20 DX = N*DN*DEL                                                       EXH20110
      X = XB + DX                                                         EXH20120
      XBB(N) = X                                                          EXH20130
   30 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(C*DX)                EXH20140
     1        - ALOG(C*DEL)*DX/DEL) + (CPI - (CPI-CPI1)*DX/DEL)           EXH20150
   40 CONTINUE                                                            EXH20160
      RETURN                                                              EXH20170
      END                                                                 EXH20180


*DECK SLOAD                                                               SLOD0010
      SUBROUTINE SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,THETA,THETAS,TST,         SLOD0020
     1    XB,DEL,BETA,EPS,CP,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,       SLOD0030
     2    CDI,CDMU,CDG,CS,CT,NROWS,IHINGE)                                SLOD0040
C                                                                         SLOD0050
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF LIFT, PITCHING     SLOD0060
C  MOMENT, AND INDUCED DRAG FOR EITHER A FUNDAMENTAL OR A COMPOSITE CASESLOD0070
C                                                                         SLOD0080
      DIMENSION IJ(40),NW(40),NJ(40)                                      SLOD0090
      DIMENSION CHORD(40),CMU(40),THETA(40),THETAS(40),TST(40)            SLOD0100
      DIMENSION CP(400),XB(400),DEL(400),BETA(400),EPS(400)               SLOD0110
      DIMENSION CLG(40),CLMU(40),CL(40),CMG(40),CMMU(40),CMT(40),CM(40),SLOD0120
     1    XBCP(40),XBCL(40),CDG(40),CDMU(40),CS(40),CDI(40),CT(40)        SLOD0130
C                                                                         SLOD0140
C  INTEGRATE THE CHORDWISE PRESSURES FOR EACH SPANWISE SECTION            SLOD0150
   20 I = 0                                                               SLOD0160
      LOGIC = 0                                                           SLOD0170
      IF(IHINGE  GE. 0)GO TO 30                                           SLOD0180
      LOGIC = 1                                                           SLOD0190
      IHINGE = 0                                                          SLOD0200
   30 IF(IHINGE .GT. 1)IHINGE = 1                                         SLOD0210
      DO 170 K = 1,NROWS                                                  SLOD0220
C                                                                         SLOD0230
C  LEADING EDGE CONTRIBUTIONS                                             SLOD0240
      I = I + 1                                                           SLOD0250
   40 CLI = DEL(I) * (CP(I)+0.50*CP(I+1))                                 SLOD0260
      CLG(K) = CLI                                                        SLOD0270
      CMG(K) = -DEL(I)**2 * (0.666667*CP(I)+CP(I+1)) / 3.0                SLOD0280
      CDG(K) = CLI * EPS(I)/57.295779                                     SLOD0290
   50 IF(LOGIC .EQ. 0)CS(K) = 0.1745329 * DEL(I) * CP(I)**2               SLOD0300
C                                                                         SLOD0310
      BCF = 0.00                                                          SLOD0320
      NWK = NW(K)                                                         SLOD0330
      DO 120 L = 2,NWK                                                    SLOD0340
      I = I + 1                                                           SLOD0350
      CPI = CP(I)                                                         SLOD0360
      IF(L .EQ. NWK) GO TO 60                                             SLOD0370
      CPI1 = CP(I+1)                                                      SLOD0380
      GO TO 70                                                            SLOD0390
C  DEFINE TRAILING EDGE CP VALUE                                          SLOD0400
   60 CPI1 = 0.0                                                          SLOD0410
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 70               SLOD0420
      IJK = IJ(K)                                                         SLOD0430
      CPI1 = CP(IJK)                                                      SLOD0440
```

```
C
C  REGULAR EVD CONTRIBUTIONS
   70 CLI = 0.50 * DEL(I) * (CPI+CPI1)
      CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CLI*XB(I) - (CPI+2.0*CPI1)*DEL(I)**2/6.00
   80 CDG(K) = CDG(K) + CLI * EPS(I)/57.295779
      BCF = BCF + BETA(I) * (1.0-XB(I))
C
C  HINGE CONTRIBUTIONS
      IF(IHINGE .EQ. 0) GO TO 120
      B2 = BETA(I+1)
      IF(L .LT. NWK) GO TO 90
      B2 = 0.00
      IJK = IJ(K)
      IF(CMU(K) .GT. 0.0001) B2 = BETA(IJK)
   90 IF((ABS(BETA(I)) .LT. 0.0001).AND.(ABS(B2) .LT. 0.0001)) GO TO 120
      CLI = 0.00
      CMI = 0.00
      DL = ALOG(DEL(I) * CHORD(K))
      CON = 0.6366198 * DEL(I) / 57.295779
      IF(ABS(BETA(I)) .LT. 0.0001) GO TO 100
      CLI = CON * BETA(I) * (2.00 - DL)
      CMI = BETA(I) * (0.50-DL/3.00)
  100 IF(ABS(B2) .LT. 0.0001) GO TO 110
      CLI = CLI + CON * B2 * (2.00 - DL)
      CMI = CMI + 2.00 * B2 * (0.7500 - DL/3.00)
  110 CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CON * DEL(I) * CMI - CLI * XB(I)
      CDG(K) = CDG(K) + CLI * EPS(I)/57.295779
  120 CONTINUE
C
C  COMPUTE THE SECTIONAL COEFFICIENTS
  130 CLMU(K) = CMU(K) * THETA(K)/57.295779
      CL(K) = CLG(K) + CLMU(K)
  140 CMMU(K) = -CMU(K) * THETA(K)/57.295779
      CMT(K) = CMU(K) * (ALPHA+TST(K)-THETAS(K)+BCF)/57.295779
      CM(K) = CMG(K) + CMMU(K) + CMT(K)
      XBCP(K) = 0.00
      XBCL(K) = 0.00
  150 IF(CLG(K) .NE. 0.00) XBCP(K) = -CMG(K) / CLG(K)
      IF(CL(K) .NE. 0.00) XBCL(K) = -(CMG(K)+CMMU(K)) / CL(K)
  160 CDMU(K) = CMU(K) * (THETA(K)/57.295779)**2 / 2.00
      CDI(K) = CDG(K) + CDMU(K) - CS(K)
      CT(K) = CMU(K) - CDI(K)
  170 CONTINUE
  180 RETURN
      END
```

```
*DECK SLOADX
      SUBROUTINE SLOADX(CPA,CPO,DEL,EPS,CMU,TH,NW,NJ,IJ,
     1              CLGO,CDGX,CDMUX,CSX,CDIX,NROWS,LOGIC)
C
C  THIS SUBROUTINE CALCULATES THE SECTIONAL CROSS-PRODUCT VALUES
C  OF THE NONLINEAR DRAG COEFFICIENTS
C
      DIMENSION CPA(400),CPO(400),DEL(400),EPS(400)
      DIMENSION CMU(40),TH(40),NW(40),NJ(40),IJ(40)
```

```
      DIMENSION CLGO(40),CDGX(40),CDMUX(40),CSX(40),CDIX(40)
C
      I = 0
      DO 80  K = 1,NROWS
C  LEADING EDGE EVD CONTRIBUTION
      I = I + 1
   20 CDGX(K) = DEL(I)*(CPA(I)+0.50*CPA(I+1))*EPS(I)/57.295779
   30 IF(LOGIC .EQ. 0)CSX(K) = 0.3490658 * DEL(I) * (CPO(I) + CPA(I))
      NWK = NW(K)
      DO 60 L = 2,NWK
      I = I + 1
   40 CPI1 = CPA(I+1)
C  DEFINE TRAILING EDGE CP VALUE
      IF(L .LT. NWK) GO TO 50
      CPI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 50
      IJK = IJ(K)
      CPI1 = CPA(IJK)
C  REGULAR EVD CONTRIBUTION
   50 CDGX(K) = CDGX(K) + 0.50*DEL(I)*(CPA(I)+CPI1)*EPS(I)/57.295779
   60 CONTINUE
C
C  COMPUTE THE REMAINING SECTIONAL COEFFICIENTS
   70 CDGX(K) = CDGX(K) + CLGO(K)/57.295779
      CDMUX(K) = CMU(K) * TH(K)/57.295779**2
      CDIX(K) = CDGX(K) + CDMUX(K) - CSX(K)
   80 CONTINUE
      RETURN
      END
```

```
*DECK SLOADG
      SUBROUTINE SLOADG(CP,DEL,BETA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,
     1              NROWS,IHINGE)
C
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF TOTAL VORTICITY
C  ON THE WING-JET SYSTEM
C
      DIMENSION CP(400),DEL(400),BETA(400)
      DIMENSION CHORD(40),D(40),CMU(40),NJ(40),IJ(40)
      DIMENSION CLG(40),CGAM(40)
C
      DO 70  K = 1,NROWS
C
C  COMPUTE THE SECTIONAL JET VORTICITY, INTEGRATED FROM T.E. TO INFINITY
   20 CGAM(K) = 0.00
      IF(CMU(K) .LT. 0.0001) GO TO 60
      II = IJ(K)
C  HINGE EVD CONTRIBUTION
      IF(IHINGE .EQ. 0) GO TO 30
      IF(BETA(II) .NE. 0.00) CGAM(K) = 0.6366198 * DEL(II) *
     1      BETA(II)/57.295779 * (2.00-ALOG(CHORD(K)*DEL(II)))
C  REGULAR EVD CONTRIBUTION
   30 NJK1 = NJ(K) - 1
      II = II - 1
      DO 50 L = 1,NJK1
      II = II + 1
   40 CGAM(K) = CGAM(K) + 0.50 * DEL(II) * (CP(II)+CP(II+1))
   50 CONTINUE
C  FAR-JET EVD CONTRIBUTION
      II = II + 1
      CGAM(K) = CGAM(K) +          D(K) / CHORD(K) * CP(II)
C
C  SUM UP THE WING AND JET CONTRIBUTIONS
   60 CGAM(K) = 0.50 * CHORD(K) * (CGAM(K)+CLG(K))
   70 CONTINUE
      RETURN
      END
```

```
*DECK LOADG1
      SUBROUTINE LOADG1(ALPHA,IJ,NW,NJ,CHORD,CMU,THETA,THETAS,TST,
     1     XB,DEL,BETA,EPS,CP,CLG,CL,CLMU,CMG,CM,CMMU,CMT,XBCP,XBCL,
     2     NROWS)
      DIMENSION IJ(40),NW(40),NJ(40),CHORD(40),CMU(40),THETA(40)
      DIMENSION THETAS(40),TST(40),XB(400),DEL(400),BETA(400),EPS(400)
      DIMENSION CP(400),CLG(40),CL(40),CLMU(40),CMG(40),CM(40),CMMU(40)
      DIMENSION CMT(40),XBCP(40),XBCL(40)
      I = 0
      DO 40 K = 1,NROWS
      I = I + 1
      CLI = DEL(I) * (CP(I) + 0.50*CP(I+1))
      CLG(K) = CLI
      CMG(K) = -DEL(I)**2 * (CP(I) + CP(I+1)/3.0) / 3.0
      BCF = 0.0
      NWK = NW(K)
      DO 30 L = 2,NWK
      I = I + 1
      CPI = CP(I)
      IF(L .EQ. NWK) GO TO 10
      CPI1 = CP(I+1)
      GO TO 20
   10 CPI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 20
      IJK = IJ(K)
      CPI1 = CP(IJK)
   20 CLI = 0.50 * DEL(I) * (CPI + CPI1)
      CLG(K) = CLG(K) + CLI
      CMG(K) = CMG(K) - CLI*XB(I) - (CPI+2.0*CPI1)*DEL(I)**2/6.0
      BCF = BCF + BETA(I) * (1.0 - XB(I))
   30 CONTINUE
      CLMU(K) = CMU(K) * THETA(K) / 57.295779
      CL(K) = CLG(K) + CLMU(K)
      CMMU(K) = -CMU(K) * THETA(K) / 57.295779
      CMT(K) = CMU(K) * (ALPHA+TST(K)-THETAS(K)+BCF)/57.295779
      CM(K) = CMG(K) + CMMU(K) + CMT(K)
      XBCP(K) = 0.0
      XBCL(K) = 0.0
      IF(CLG(K) .NE. 0.00) XBCP(K) = -CMG(K) / CLG(K)
      IF(CL(K) .NE. 0.00) XBCL(K) = -(CMG(K)+CMMU(K)) / CL(K)
   40 CONTINUE
      RETURN
      END
```

```
*DECK LOADG2
```

```
      SUBROUTINE LOADG2(ALPHA,IJ,NW,NJ,CHORD,CMU,THETA,THETAS,TST,
     1     XB,DEL,BETA,EPS,GAM1,GAM2,CDG,CDMU,CDI,CS,CT,NROWS,ITYPE)
      DIMENSION IJ(40),NW(40),NJ(40),CHORD(40),CMU(40),THETA(40)
      DIMENSION THETAS(40),TST(40),XB(400),DEL(400),BETA(400),EPS(400)
      DIMENSION GAM1(400),GAM2(400),CDG(40),CDMU(40),CDI(40),CS(40)
      DIMENSION CT(40)
      I = 0
      DO 160 K = 1,NROWS
      I = I + 1
      GO TO ( 10 , 20 , 10 ),ITYPE
   10 CLI = 2.0 * DEL(I) * (GAM1(I) + 0.5 * GAM1(I+1))
      CDG(K) = CLI * EPS(I) / 57.295779
      CS(K) = 0.6981316 * DEL(I) * GAM1(I)**2
      GO TO 30
   20 CDG(K) = 2.0 * DEL(I) * ((GAM1(I) + 0.5*GAM1(I+1))*EPS(I) +
     1             (GAM2(I)+0.5*GAM2(I+1))) / 57.295779
      CS(K) = 0.6981316 * DEL(I) * (2.0 * GAM1(I) * GAM2(I))
   30 NWK = NW(K)
      GO TO ( 40 , 80 , 40 ),ITYPE
   40 DO 70 L = 2,NWK
      I = I + 1
      GAM1I = GAM1(I)
      IF(L .EQ. NWK) GO TO 50
      GAM1I1 = GAM1(I+1)
      GO TO 60
   50 GAM1I1=  0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K).LT.0.0001)) GO TO 60
      IJK = IJ(K)
      GAM1I1 = GAM1(IJK)
   60 CLI = DEL(I) * (GAM1I + GAM1I1)
      CDG(K) = CDG(K) + CLI * EPS(I) / 57.295779
   70 CONTINUE
      GO TO 120
   80 DO 110 L = 2,NWK
      I = I + 1
      GAM1I = GAM1(I)
      GAM2I = GAM2(I)
      IF(L .EQ. NWK) GO TO 90
      GAM1I1 = GAM1(I+1)
      GAM2I1 = GAM2(I+1)
      GO TO 100
   90 GAM1I1 = 0.0
      GAM2I1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 100
      IJK = IJ(K)
      GAM1I1 = GAM1(IJK)
      GAM2I1 = GAM2(IJK)
  100 CLI = DEL(I) * ((GAM1I+GAM1I1)*EPS(I)+(GAM2I+GAM2I1))/57.295779
      CDG(K) = CDG(K) + CLI
  110 CONTINUE
  120 GO TO ( 130 , 140 , 130 ),ITYPE
  130 CDMU(K) = CMU(K) * (THETA(K)/57.295779)**2/2.0
      GO TO 150
  140 CDMU(K) = CMU(K) * THETA(K) / (57.295779)**2
  150 CDI(K) = CDG(K) + CDMU(K) - CS(K)
      CT(K) = CMU(K) - CDI(K)
  160 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GLOADG(GAM,DEL,BTA,CHORD,D,CMU,NW,IW,NJ,IJ,CGAM,NROWS)
      DIMENSION GAM(400),DEL(400),BTA(400),CHORD(40),D(40),CMU(40)
      DIMENSION NW(40),IW(40),NJ(40),IJ(40),CGAM(40)
      DO 70 K = 1,NROWS
      I = IW(K)
      CGAM(K) = DEL(I) * (GAM(I) + 0.5*GAM(I+1))
      NWK = NW(K)
      DO 40 L = 2,NWK
      I = I + 1
      GAMI = GAM(I)
      IF(L .EQ. NWK) GO TO 20
      GAMI1 = GAM(I+1)
      GO TO 30
   20 GAMI1 = 0.0
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 30
      IJK = IJ(K)
      GAMI1 = GAM(IJK)
   30 CGAM(K) = CGAM(K) + 0.5 * DEL(I) * (GAMI + GAMI1)
   40 CONTINUE
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 60
      I = IJ(K) - 1
      NJK1 = NJ(K) - 1
      DO 50 L = 1,NJK1
      I = I + 1
      CGAM(K) = CGAM(K) + 0.5 * DEL(I) * (GAM(I) + GAM(I+1))
   50 CONTINUE
      I = I + 1
      CGAM(K) = CGAM(K) + D(K) / CHORD(K) * GAM(I)
   60 CGAM(K) = CHORD(K) * CGAM(K)
   70 CONTINUE
      RETURN
      END
```

```
*DECK TLOAD
      SUBROUTINE TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,C,HO,XB,XLEAD,BETA,
     1      CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG,CCLJ,CCL,CCMG,CCMJ,CCMT,CCM,
     2      CMGMC,CMJMC,CMTMC,CMMC,CXCP,CXCL,CXCPB,CXCLB,CCJ,CLLG,CLLJ,
     3      CLL,ITYPE,IW,NW,NROWS,ISYMM)
C
C  THIS SUBROUTINE CALCULATES ALL OF THE TOTAL LOADING PARAMETERS
C  FOR A FUNDAMENTAL CASE
C
      DIMENSION Y(40),DELTA(40),C(40),HO(40),XLEAD(40),IW(40),NW(40)
      DIMENSION XB(400),BETA(400),ITYPE(400)
      DIMENSION CLG(40),CLMU(40),CMG(40),CMMU(40),CMT(40),CMU(40)
C
C  INITIALIZE THE TOTAL COEFFICIENTS
   20 CCLG = 0.00
      CCLJ = 0.00
      CCMG = 0.00
      CCMJ = 0.00
      CCMT = 0.00
      CXCP = 0.00
      CXCL = 0.00
      CXCPB = 0.00
      CXCLB = 0.00
      CCJ  = 0.00
      CLLG = 0.00
      CLLJ = 0.00
      CLL  = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 110 K = 1,NROWS
   30 CDEL = C(K) * DELTA(K)
      IF(ISYMM .LT. 0) GO TO 90
C
C  LIFT COEFFICIENTS
      CCLG = CCLG + CDEL * CLG(K)
   40 CCLJ = CCLJ + CDEL * CLMU(K)
C
C  PITCHING MOMENT COEFFICIENTS
      CCDEL = CDEL * C(K)
      XLB = XLEAD(K) / C(K)
C  COMPUTE LEADING EDGE HEIGHT ABOVE WING APEX
   50 I = IW(K) - 1
      NWK = NW(K)
      XDS = 0.00
      DO 70 L = 1,NWK
      I = I + 1
      IF(ITYPE(I) - 4)) 70 , 60 , 80
   60 XDS = XDS + XB(I) * BETA(I)/57.295779
   70 CONTINUE
   80 HLB = HO(K) - XLB * ALPHA/57.295779 - XDS
      CCMG = CCMG + CCDEL * (CMG(K) - CLG(K)*XLB)
      CCMJ = CCMJ + CCDEL * (CMMU(K) - CLMU(K)*XLB)
      CCMT = CCMT + CCDEL * (CMT(K) - CMU(K)*HLB)
   90 CCJ  = CCJ  + CDEL * CMU(K)
C
C  ROLLING MOMENT COEFFICIENTS
  100 IF(ISYMM .EQ. 0) GO TO 110
      CDELY = CDEL * Y(K)
      CLLG = CLLG + CDELY * CLG(K)
      CLLJ = CLLJ + CDELY * CLMU(K)
  110 CONTINUE
C
C  COMPUTE THE FINAL VALUES OF ALL THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
  120 CCLG = FACTOR * CCLG
      CCLJ = FACTOR * CCLJ
      CCL  = CCLG + CCLJ
  130 CCJ  = FACTOR * CCJ
  140 FACTOR = FACTOR / CREF
      CCMG = FACTOR * CCMG
      CCMJ = FACTOR * CCMJ
      CCMT = FACTOR * CCMT
      CCM  = CCMG + CCMJ + CCMT
      IF(ISYMM .LT. 0) GO TO 150
      IF(CCLG .NE. 0.00) CXCP = - CCMG / CCLG
      IF(CCL .NE. 0.00) CXCL = -(CCMG+CCMJ) / CCL
      CXCPB = CXCP * CREF
      CXCLB = CXCL * CREF
  150 FACTOR = XMC / CREF
      IF(ISYMM .LT. 0) FACTOR = 0.00
      CMGMC = CCMG + CCLG * FACTOR
      CMJMC = CCMJ + CCLJ * FACTOR
      CMTMC = CCMT - CCJ  * FACTOR * ALPHA/57.295779
      CMMC  = CMGMC + CMJMC + CMTMC
  160 IF(ISYMM .EQ. 0) GO TO 170
      FACTOR = -1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = -2.00 / AREA
      CLLG = FACTOR * CLLG
      CLLJ = FACTOR * CLLJ
      CLL  = CLLG + CLLJ
  170 RETURN
      END
```

```
*DECK TLOADO
      SUBROUTINE TLOADO(CREF,CCLG;CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,
     1              CLLG,CLLJ,FACT,CCLGO,CCLJO,CCLO,CCMGO,CCMJO,CCMTO,CCMO,
     2              CMGMCO,CMJMCO,CMTMCO,CMMCO,CXCPO,CXCLO,CXCPBO,CXCLBO,
     3              CLLGO,CLLJO,CLLO,NCASES,ISYMM)
C
C  THIS SUBROUTINE CALCULATES THE LINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0.  LINEAR QUANTITIES ARE MODULATED AND SUMMED
C  ACCORDING TO THE COMPOSITE CASE REQUIREMENTS.
C
      DIMENSION CCLG(10),CCLJ(10),CCMG(10),CCMJ(10),CCMT(10),
     1             CMGMC(10),CMJMC(10),CMTMC(10),CLLG(10),CLLJ(10),FACT(10)
C
C  INITIALIZE THE COEFFICIENTS
   20 CCLGO = 0.00
      CCLJO = 0.00
      CCMGO = 0.00
      CCMJO = 0.00
      CCMTO = 0.00
      CMGMCO = 0.00
      CMJMCO = 0.00
      CMTMCO = 0.00
      CLLGO = 0.00
      CLLJO = 0.00
      CXCPO = 0.00
      CXCLO = 0.00
      CXCPBO = 0.00
      CXCLBO = 0.00
C
C  MODULATE AND SUM THE TOTAL COEFFICIENTS
      DO 50 N = 1,NCASES
   30 CCLGO = CCLGO + CCLG(N) * FACT(N)
      CCLJO = CCLJO + CCLJ(N) * FACT(N)
      CCMGO = CCMGO + CCMG(N) * FACT(N)
      CCMJO = CCMJO + CCMJ(N) * FACT(N)
      CCMTO = CCMTO + CCMT(N) * FACT(N)
   40 CMGMCO = CMGMCO + CMGMC(N) * FACT(N)
      CMJMCO = CMJMCO + CMJMC(N) * FACT(N)
      CMTMCO = CMTMCO + CMTMC(N) * FACT(N)
      CLLGO = CLLGO + CLLG(N) * FACT(N)
      CLLJO = CLLJO + CLLJ(N) * FACT(N)
   50 CONTINUE
C  DEFINE THE REMAINING TOTAL COEFFICIENTS
      CCLO = CCLGO + CCLJO
   60 CCMO = CCMGO + CCMJO + CCMTO
      IF(ISYMM .LT. 0) GO TO 70
      IF(CCLGO .NE. 0.00) CXCPO = - CCMGO / CCLGO
      IF(CCLO .NE. 0.00) CXCLO = - (CCMGO+CCMJO) / CCLO
      CXCPBO = CXCPO * CREF
      CXCLBO = CXCLO * CREF
   70 CMMCO = CMGMCO + CMJMCO + CMTMCO
      CLLO = CLLGO + CLLJO
   80 RETURN
      END
```

```
*DECK TLOADX
      SUBROUTINE TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,CLO,
     1                    CCDG,CCDJ,CCS,CCDI,CDITZ,ALFINF,ALFINO,CCT,CCJ,
     2                    CNJ,CNJJ,CNG,CCY,XLEAD,TANLE,XMC,NROWS,ISYMM,
     3                    LOGIC)
C
C  THIS SUBROUTINE CALCULATES THE NONLINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0 BY SPANWISE INTEGRATION OF THE NONLINEAR
C  SECTIONAL COEFFICIENTS
C
      DIMENSION CHORD(40),DELTA(40),Y(40),CMU(40),XLEAD(40),TANLE(40)
      DIMENSION CDG(40),CDMU(40),CS(40),CDI(40),CL(40),CLO(40)
      DIMENSION ALFINF(40),ALFINO(40)
C
C  INITIALIZE THE COEFFICIENTS
   20 CCDG = 0.00
      CCDJ = 0.00
      CCS = 0.00
      CDITZ = 0.00
      CNJ = 0.00
      CNJJ = 0.00
      CNG = 0.00
      CCY = 0.00
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 50  K = 1,NROWS
   30 CDEL = CHORD(K) * DELTA(K)
      CCDG = CCDG + CDEL * CDG(K)
      CCDJ = CCDJ + CDEL * CDMU(K)
   40 CCS = CCS + CDEL * CS(K)
      CDITZ = CDITZ + CDEL * (CL(K) * ALFINO(K) + CLO(K) * ALFINF(K))
      IF(ISYMM .LT. 1) GO TO 50
      CNJ = CNJ + CDEL * Y(K) * (CMU(K)-CDMU(K))
      CNJJ = CNJJ + CDEL * Y(K) * CDMU(K)
      CNG = CNG+CDEL*(Y(K)*(CDG(K)-CS(K))-CS(K)*TANLE(K)*(XLEAD(K)-XMC))
      CCY = CCY + CDEL * CS(K) * TANLE(K)
   50 CONTINUE
C  COMPUTE THE FINAL VALUES OF THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
   60 CCDG = FACTOR * CCDG
      CCDJ = FACTOR * CCDJ
      CCS = FACTOR * CCS
      CDITZ = FACTOR * CDITZ / 2.00
      CCDI = CCDG + CCDJ - CCS
      CCT = CCJ - CCDI
   70 IF(ISYMM .LT. 1) RETURN
      FACTOR = 1.00 / AREA
      CNJ = -FACTOR * CNJ
      CNJJ = FACTOR * CNJJ
```

```
      CNG = FACTOR * CNG                                                TLDX0500
      CCY = FACTOR * CCY                                                TLDX0510
   80 RETURN                                                            TLDX0520
      END                                                               TLDX0530


*DECK TREFTZ                                                            TFTZ0010
      SUBROUTINE TREFTZ(Y,DELTA,CMU,GAMB,ALFINF,NROWS,LIKE)             TFTZ0020
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          TFTZ0030
      COMMON/SOLV3 / DUMMY(400)                                         TFTZ0040
      DIMENSION Y(40),DELTA(40),CMU(40),GAMB(40),ALFINF(40)             TFTZ0050
      DIMENSION E(40),B(40),C(40),DP(40),DM(40),DGAM(40)                TFTZ0060
      DIMENSION SQ1(42),G(42),H33(42)                                   TFTZ0070
      EQUIVALENCE (SQ1(1),DUMMY(1)),(G(1),DUMMY(43)),(H33(1),DUMMY(85)),TFTZ0080
     1(E(1),DUMMY(125)),(B(1),DUMMY(165)),(C(1),DUMMY(205)),(DP(1),DUMMYTFTZ0090
     2(245)),(DM(1),DUMMY(285)),(DGAM(1),DUMMY(325))                    TFTZ0100
      LOGICAL Z1,Z2,Z3,Z4,Z5,Z6,ZT1,ZT2                                 TFTZ0110
C                                                                       TFTZ0120
C  FIND COEFFICIENTS OF CURVES                                          TFTZ0130
      J=0                                                               TFTZ0140
      DO 70 I=1,NROWS                                                   TFTZ0150
      Z1=I.EQ.1                                                         TFTZ0160
      Z2=LIKE.GT.0.AND.I.EQ.NROWS                                       TFTZ0170
      Z3=(LIKE.LE.0.AND.I.EQ.NROWS).OR.(LIKE.GT.0.AND.I.NE.NROWS.AND.   TFTZ0180
     1Y(I).GT.0.0.AND.Y(I+1).LT.0.0)                                    TFTZ0190
      Z4=LIKE.GT.0 AND.I.GT.1.AND.Y(I).LT.0.0.AND.Y(I-1).GT.0.0         TFTZ0200
      Z5=(I.NE.NROWS).AND.((CMU(I).LT.0.0001.AND.CMU(I+1).GT.0.0).OR.   TFTZ0210
     1(CMU(I).GT 0.0.AND.CMU(I+1).LT.0.0001))                           TFTZ0220
      Z6=(I.GT.1).AND.((CMU(I).GT.0.0.AND.CMU(I-1).LT.0.0001).OR.       TFTZ0230
     1(CMU(I).LT.0.0001.AND.CMU(I-1).GT.0.0))                           TFTZ0240
      D1=SQRT(1.0-ABS(Y(I)))                                            TFTZ0250
      ZT1=Z1.OR.Z4.OR.Z6                                                TFTZ0260
      ZT2=Z2.OR.Z3.OR.Z5                                                TFTZ0270
      IF(J.EQ.1)GO TO 20                                                TFTZ0280
      IF(ZT1)J=1                                                        TFTZ0290
      IF(ZT1)GO TO 50                                                   TFTZ0300
      J=0                                                               TFTZ0310
   20 CONTINUE                                                          TFTZ0320
      IF(ZT2) GO TO 40                                                  TFTZ0330
      D2=SQRT(1.0-ABS(Y(I+1)))                                          TFTZ0340
      D3=SQRT(1.0-ABS(Y(I-1)))                                          TFTZ0350
      G1=GAMB(I)                                                        TFTZ0360
      G2=GAMB(I+1)                                                      TFTZ0370
      G3=GAMB(I-1)                                                      TFTZ0380
      D13=D1**3                                                         TFTZ0390
      D23=D2**3                                                         TFTZ0400
      D33=D3**3                                                         TFTZ0410
      D15=D1**5                                                         TFTZ0420
      D25=D2**5                                                         TFTZ0430
      D35=D3**5                                                         TFTZ0440
      F1=D23*D35-D33*D25                                                TFTZ0450
      F2=D13*D35-D33*D15                                                TFTZ0460
      F3=D13*D25-D23*D15                                                TFTZ0470
      E(I)=(G1*F1-G2*F2+G3*F3)/(D1*F1-D2*F2+D3*F3)                      TFTZ0480
      B(I)=((G1-E(I)*D1)*D25-(G2-E(I)*D2)*D15)/(D13*D25-D23*D15)        TFTZ0490
      C(I)=(G1-E(I)*D1-B(I)*D13)/D15                                    TFTZ0500
      IF(J.EQ.0) GO TO 50                                               TFTZ0510
   30 E(I-1)=E(I)                                                       TFTZ0520


      B(I-1)=B(I)                                                       TFTZ0530
      C(I-1)=C(I)                                                       TFTZ0540
      J=0                                                               TFTZ0550
      GO TO 50                                                          TFTZ0560
   40 E(I)=E(I-1)                                                       TFTZ0570
      B(I)=B(I-1)                                                       TFTZ0580
      C(I)=C(I-1)                                                       TFTZ0590
   50 CONTINUE                                                          TFTZ0600
      IF(Y(I).LT.0.0) GO TO 60                                          TFTZ0610
      DP(I)=SQRT(1.0-ABS(Y(I))-DELTA(I))                                TFTZ0620
      DM(I)=SQRT(1.0-ABS(Y(I))+DELTA(I))                                TFTZ0630
      GO TO 70                                                          TFTZ0640
   60 DP(I)=SQRT(1.0-ABS(Y(I))+DELTA(I))                                TFTZ0650
      DM(I)=SQRT(1.0-ABS(Y(I))-DELTA(I))                                TFTZ0660
   70 CONTINUE                                                          TFTZ0670
C                                                                       TFTZ0680
C  CALCULATE STRENGTH OF DISCRETE VORTICIES                             TFTZ0690
      DO 90 I=1,NROWS                                                   TFTZ0700
      IF(I.LT.NROWS.AND.((CMU(I).LT.0.0001.AND.CMU(I+1).GT.0.0).OR.     TFTZ0710
     1(CMU(I).GT.0.0.AND.CMU(I+1).LT.0.0001))) GO TO 80                 TFTZ0720
      DGAM(I)=0.0                                                       TFTZ0730
      GO TO 90                                                          TFTZ0740
   80 DGAM(I)=(E(I+1)-E(I))*DM(I)+(B(I+1)-B(I))*DM(I)**3+(C(I+1)-C(I))  TFTZ0750
     1*DM(I)**5                                                         TFTZ0760
   90 CONTINUE                                                          TFTZ0770
C                                                                       TFTZ0780
C  CALCULATE DOWNWASH                                                   TFTZ0790
      DO 110 I=1,NROWS                                                  TFTZ0800
      ALFINF(I)=0.0                                                     TFTZ0810
      F2=SQRT(1.0-Y(I))                                                 TFTZ0820
      F4=SQRT(1.0+Y(I))                                                 TFTZ0830
      DO 100 J=1,NROWS                                                  TFTZ0840
      IF(Y(J).LT.0.0) F2=F4                                             TFTZ0850
      CON=1.0                                                           TFTZ0860
      IF(Y(J).LT.0.0) CON=-1.0                                          TFTZ0870
      F1=ALOG(ABS((DP(J)-F2)*(DM(J)+F2))/((DP(J)+F2)*(DM(J)-F2))))      TFTZ0880
      S1=CON*F1/F2                                                      TFTZ0890
      S2=(2.0*(DP(J)-DM(J))+F2*F1)*CON                                  TFTZ0900
      S3=CON*(2.0*(DP(J)**3-DM(J)**3)/3.0)+F2**2*S2                     TFTZ0910
      DELG=-DGAM(J)/(Y(I)-Y(J)+DELTA(J))                                TFTZ0920
      IF(LIKE.GT.0) GO TO 100                                           TFTZ0930
      F3=ALOG(ABS(((DP(J)-F4)*(DM(J)+F4))/((DP(J)+F4)*(DM(J)-F4))))     TFTZ0940
      S2P=2.0*(DP(J)-DM(J))+F4*F3                                       TFTZ0950
      IF(LIKE.LT.0) SY=-1.0                                             TFTZ0960
      IF(LIKE.EQ.0) SY=1.0                                              TFTZ0970
      S1=S1+SY*F3/F4                                                    TFTZ0980
      S2=S2+SY*S2P                                                      TFTZ0990
      S3=S3+SY*((2.0*(DP(J)**3-DM(J)**3)/3.0)+F4**2*S2P)                TFTZ1000
      DELG=DELG-SY*DGAM(J)/(-Y(I)-Y(J)+DELTA(J))                        TFTZ1010
  100 ALFINF(I)=ALFINF(I)+0.1591549*(0.5*E(J)*S1+1.5*B(J)*S2+2.5*C(J)*  TFTZ1020
     1S3+DELG)                                                          TFTZ1030
  110 CONTINUE                                                          TFTZ1040
      IF (IGND.EQ.0) GO TO 180                                          TFTZ1050
      H11 =4.0* GHITE(NHITE)**2                                         TFTZ1060
      K1=0                                                              TFTZ1070
      NR1 = NROWS+1                                                     TFTZ1080
      NR2=NROWS+2                                                       TFTZ1090
      G(1)= 0.0                                                         TFTZ1100
      SQ1(1)= 1.0                                                       TFTZ1110
      DO 120 I = 2,NR1                                                  TFTZ1120
      SQ1(I)= Y(I-1)                                                    TFTZ1130
  120 G(I)= GAMB(I-1)                                                   TFTZ1140
      SQ1(NR2)= -1.0                                                    TFTZ1150
      IF(LIKE .LE. 0) SQ1(NR2)= 0.0                                     TFTZ1160
      G(NR2)= 0.0                                                       TFTZ1170
      IF(LIKE .EQ.0) G(NR2)=E(NROWS)+B(NROWS)+G(NROWS)                  TFTZ1180
      GO TO 150                                                         TFTZ1190
  130 K1=1                                                              TFTZ1200
      DO 140 I=1,NR2                                                    TFTZ1210
      IF(LIKE.LT.0) G(I)=-G(I)                                          TFTZ1220
  140 SQ1(I)=-SQ1(I)                                                    TFTZ1230
  150 DO 170 I=1,NROWS                                                  TFTZ1240
      DO 160 J=1,NR2                                                    TFTZ1250
      H22 = (SQ1(J)-Y(I))**2                                            TFTZ1260
  160 H33(J)= G(J)*(H11-H22)/(H11+H22)**2                               TFTZ1270
      CALL QUA (SQ1,H33,H44,NR2)                                        TFTZ1280
      IF(K1.EQ.1) H44=-H44                                              TFTZ1290
  170 ALFINF(I)= ALFINF(I) +0.1591549*H44                               TFTZ1300
      IF(K1.EQ.0.AND.LIKE.LE.0) GO TO 130                               TFTZ1310
  180 RETURN                                                            TFTZ1320
      END                                                               TFTZ1330


*DECK QUA                                                               QUA 0010
      SUBROUTINE QUA (X,Y,TA,NPT)                                       QUA 0020
      DIMENSION X(42),Y(42)                                             QUA 0030
      IF  ( NPT .LT. 4 )  GO TO 70                                      QUA 0040
      TA = 0.0                                                          QUA 0050
      DO 60   I = 1, NPT                                                QUA 0060
      K = I - 1                                                         QUA 0070
      IF  ( I - 2 ) 60 , 40 , 20                                        QUA 0080
   20 IF  ( I .LT. NPT )  GO TO 30                                      QUA 0090
      K = K - 1                                                         QUA 0100
   30 K = K - 1                                                         QUA 0110
   40 A = X(I)                                                          QUA 0120
      B = X(I-1)                                                        QUA 0130
      L = K + 1                                                         QUA 0140
      M = K + 2                                                         QUA 0150
      N = K + 3                                                         QUA 0160
      TAB = 0.0                                                         QUA 0170
      DO 50   J = 1, 4                                                  QUA 0180
      XL = X(L)                                                         QUA 0190
      XN = X(N)                                                         QUA 0200
      XM = X(M)                                                         QUA 0210
      XK = X(K)                                                         QUA 0220
      YK = Y(K)                                                         QUA 0230
      SUM = YK / ( ( XK - XM ) * ( XK - XN ) * ( XK - XL  ) )           QUA 0240
      SUM1 = ( ( A ** 4 ) - ( B ** 4 ) ) / 4.0                          QUA 0250
      SUM2 = ( XL + XM + XN ) * ( ( A ** 3 ) - ( B ** 3 ) ) / 3.0       QUA 0260
      SUM3 = ( XM * XN + XM * XL + XL * XN ) * ( A ** 2 - B ** 2 ) / 2.0QUA 0270
      SUM4 = ( XM * XN * XL ) * ( A - B )                               QUA 0280
      SUM = SUM * ( SUM1 - SUM2 + SUM3 - SUM4 )                         QUA 0290
      TAB=TAB+SUM                                                       QUA 0300
      ITEMP = K                                                         QUA 0310
      K = N                                                             QUA 0320
      N = M                                                             QUA 0330
      M = L                                                             QUA 0340
   50 L = ITEMP                                                         QUA 0350


      TA=TA+TAB                                                         QUA 0360
   60 CONTINUE                                                          QUA 0370
   70 RETURN                                                            QUA 0380
      END                                                               QUA 0390


*DECK STG3C                                                             ST3C0010
      OVERLAY(OVLY,3,1)                                                 ST3C0020
      PROGRAM STG3C                                                     ST3C0030
C                                                                       ST3C0040
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE, SPANWISE AND      ST3C0050
C  TOTAL LOADING FOR THE REQUIRED COMPOSITE CASES                       ST3C0060
C                                                                       ST3C0070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                ST3C0080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            ST3C0090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) ST3C0100
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     ST3C0110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  ST3C0120
     1              D(40),KK(400),ITYPE(400)                            ST3C0130
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             ST3C0140
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     ST3C0150
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTST3C0160
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    ST3C0170
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           ST3C0180
      COMMON/SOLV1/CP(400,10)                                           ST3C0190
      COMMON/COMPDS/FACTOR(10,24),NCC                                   ST3C0200
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          ST3C0210
      COMMON/LOAD1/ TWIST(40),HO(40),TH(40),THETS(40),                  ST3C0220
     1          BTA(400),EP(400),DPO(400)                               ST3C0230
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),   ST3C0240
     1          CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)ST3C0250
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),        ST3C0260
     1      CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),    ST3C0270
     2      CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),ST3C0280
     3      CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNJJ(10),CNG(10),ST3C0290
     4      CCY(10),CXCPB(10),CXCLB(10)                                 ST3C0300
      COMMON/LOAD4/ CLGO(40),CLMUO(40),CLO(40),CDMUO(40),CDGO(40),      ST3C0310
     1      CDIO(40),CSO(40),CTO(40),CMGO(40),CMMUO(40),CMTO(40),CMO(40),ST3C0320
     2          XBCPO(40),XBCLO(40),FACT(10)                            ST3C0330
      COMMON/LOAD5/ CGAM(40),CGAMO(40),ALFINF(40),ALFINO(40),DUMB(40)   ST3C0340
      COMMON/LOAD6/ COEFF(27)                                           ST3C0350
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HT,HH,VL,VH,     ST3C0360
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,   ST3C0370
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             ST3C0380
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XBB (41),ZCL(41),  ST3C0390
     1               RADIUS(41),AMAJ(41),AMIN(41),CSA(41),              ST3C0400
     2               V(40,7),W(40,7)                                    ST3C0410
      COMMON/KEEPEM/PUT(102)                                            ST3C0420
      EQUIVALENCE (PUT(1),CCLO),(PUT(6),CMMCO),(PUT(11),CLLO),          ST3C0430
     1         (PUT(16),CDITZO),(PUT(17),CDITZX),(PUT(26),CCYO),        ST3C0440
     2         (PUT(27),CCYX)                                           ST3C0450
      DIMENSION CDMUX(40),CDGX(40),CDIX(40),CSX(40),CPA(400),DUM(400)   ST3C0460
      EQUIVALENCE (CP(1),CPA(1)),(EP(1),DUM(1))                         ST3C0470
      LOGICAL PRINT                                                     ST3C0480
C                                                                       ST3C0490
C  CALCULATE AND PRINT THE CHORDWISE LOADING FOR ALL COMPOSITE CASES    ST3C0500
      PRINT = (IPRINT .GT. 0)                                           ST3C0510
      M = KCASE                                                         ST3C0520
```

```
      LOGIC = 0                                                         ST3C0530
      I = 0                                                             ST3C0540
      II = NWT                                                          ST3C0550
      NC1 = NCASES + 1                                                  ST3C0560
      IWRITE = 2*NMAX + NSEG + M                                        ST3C0570
      CALL HEADER(ILINES)                                               ST3C0580
      ILINES = ILINES + 9                                               ST3C0590
      WRITE(6, 20 ) M                                                   ST3C0600
   20 FORMAT(1H0,39X,10(5H*****)/                                       ST3C0610
     1        40X,44H*  WING CHORDWISE LOADING FOR COMPOSITE CASE,I3,   ST3C0620
     2        3H  */40X,10(5H*****))                                    ST3C0630
      WRITE(6, 30 ) (N,N=1,10),(FACTOR(N,M),N=1,10)                     ST3C0640
   30 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),  ST3C0650
     1       3X),5X,2HA(,I2,1H)/10X,10F12.6)                            ST3C0660
      WRITE(6, 40 )                                                     ST3C0670
   40 FORMAT(1H0,7X,47H*** NOTE ***  EACH LEADING EDGE CP VALUE IS THE, ST3C0680
     1        43H AVERAGE VALUE OF THE SINGULAR DISTRIBUTION)           ST3C0690
      IF(IHINGE .NE. 0) WRITE(6, 50 )                                   ST3C0700
   50 FORMAT(1H ,21X,47HIF A HINGE IS DEFLECTED THE LOADING IS SINGULAR,ST3C0710
     1    58H AND THE CP(A=0) VALUE IS FOR THE REGULAR EVD PORTION ONLY)ST3C0720
      WRITE(6, 60 )                                                     ST3C0730
   60 FORMAT(1H ,21X,41HDO NOT PLOT THESE LOADING POINTS DIRECTLY)      ST3C0740
      DO 240 K = 1,NROWS                                                ST3C0750
C  ON THE WING                                                          ST3C0760
      NWK = NW(K)                                                       ST3C0770
      DO 90   L = 1,NWK                                                 ST3C0780
      I = I + 1                                                         ST3C0790
   70 CPO(I) = 0.00                                                     ST3C0800
      BTA(I) = 0.00                                                     ST3C0810
      DUMK(I) = 0.00                                                    ST3C0820
      DO 80   N = 1,NCASES                                              ST3C0830
      FACT(N) = FACTOR(N,M)                                             ST3C0840
      CPO(I) = CPO(I) + CP(I,N) * FACT(N)                               ST3C0850
      BTA(I) = BTA(I) + BETA(I,N) * FACT(N)                             ST3C0860
   80 CONTINUE                                                          ST3C0870
   90 CONTINUE                                                          ST3C0880
      IF(NC1 .GT. 10) GO TO 110                                         ST3C0890
      DO 100 N = NC1,10                                                 ST3C0900
      FACT(N) = 0.00                                                    ST3C0910
  100 CONTINUE                                                          ST3C0920
C  ON THE JET                                                           ST3C0930
  110 NJK = NJ(K)                                                       ST3C0940
      IF(CMU(K) .LT. 0.0001) GO TO 150                                  ST3C0950
      DO 140 L = 1,NJK                                                  ST3C0960
      II = II + 1                                                       ST3C0970
  120 CPO(II) = 0.00                                                    ST3C0980
      BTA(II) = 0.00                                                    ST3C0990
      DO 130 N = 1,NCASES                                               ST3C1000
      CPO(II) = CPO(II) + CP(II,N) * FACT(N)                            ST3C1010
      BTA(II) = BTA(II) + BETA(II,N) * FACT(N)                          ST3C1020
  130 CONTINUE                                                          ST3C1030
  140 CONTINUE                                                          ST3C1040
  150 IF(PRINT) GO TO 240                                               ST3C1050
      J1 = IW(K)                                                        ST3C1060
      J2 = IW(K) + NWK - 1                                              ST3C1070
      NEXT = 3 + (2+NOALFA) * (NWK/10+1)                                ST3C1080
      IF(CMU(K) .GT. 0.0) NEXT = NEXT + (2+NOALFA) * (NJ(K)/10+1) + 1   ST3C1090
      ILINES = ILINES + NEXT                                            ST3C1100
      IF((ILINES .LT. 56) .OR. (K .EQ. 1)) GO TO 170                    ST3C1110
      WRITE(6, 160 )                                                    ST3C1120
  160 FORMAT(1H1)                                                       ST3C1130
      ILINES = 1                                                        ST3C1140
  170 WRITE(6, 180 ) K,Y(K),CHORD(K)                                    ST3C1150
  180 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/    ST3C1160
     1        2X,4HWING)                                                ST3C1170
      WRITE(6, 190 ) (XB(J),J=J1,J2)                                    ST3C1180
  190 FORMAT(1H ,7X,2HXB,10F12.6,3(/10X,10F12.6))                       ST3C1190
      WRITE(6, 200 ) (CPO(J),J=J1,J2)                                   ST3C1200
  200 FORMAT(1H ,6X, 3HCPO,    10F12.6,3(/10X,10F12.6))                 ST3C1210
      IF(NOALFA .LT. 1) GO TO 220                                       ST3C1220
      WRITE(6, 210 ) (CPA(J),J=J1,J2)                                   ST3C1230
  210 FORMAT(1H ,6X, 3HCPA,    10F12.6,3(/10X,10F12.6))                 ST3C1240
  220 IF(CMU(K) .LT. 0.0001) GO TO 240                                  ST3C1250
      J1 = IJ(K)                                                        ST3C1260
      J2 = IJ(K) + NJK - 1                                              ST3C1270
      WRITE(6, 230 )                                                    ST3C1280
  230 FORMAT(1H ,4H JET)                                                ST3C1290
      WRITE(6, 190 ) (XB(J),J=J1,J2)                                    ST3C1300
      WRITE(6, 200 ) (CPO(J),J=J1,J2)                                   ST3C1310
      IF(NOALFA .GT. 0) WRITE(6, 210 ) (CPA(J),J=J1,J2)                 ST3C1320
  240 CONTINUE                                                          ST3C1330
C  COMPUTE AND PRINT SPANWISE AND TOTAL LOADINGS FOR EACH COMPOSITE CASEST3C1340
C  DEFINE THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 1                  ST3C1350
      IF(NOALFA .EQ. 0) GO TO 300                                       ST3C1360
      DO 260 K = 1,NROWS                                                ST3C1370
  250 TWIST(K) = TST(K,1)                                               ST3C1380
      TH(K) = THETA(K,1)                                                ST3C1390
      THETS(K) = THS(K,1)                                               ST3C1400
  260 CONTINUE                                                          ST3C1410
      DO 280 I = 1,NEWMAX                                               ST3C1420
  270 BTA(I) = BETA(I,1)                                                ST3C1430
      EP(I) = EPS(I,1)                                                  ST3C1440
  280 CONTINUE                                                          ST3C1450
C  COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 1                         ST3C1460
      ALPHA = 1.0                                                       ST3C1470
  290 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,               ST3C1480
     1       XB,DEL,BTA,EP,CPA,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,   ST3C1490
     2       CDI,CDMU,CDG,CS,CTO,NROWS,IHINGE)                          ST3C1500
C  COMPUTE SECTIONAL VORTICITY FOR ALPHA = 1                            ST3C1510
      CALL SLOADG(CPA,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)  ST3C1520
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 1                 ST3C1530
      CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)                  ST3C1540
C  MODULATE AND SUM THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 0        ST3C1550
  300 DO 330 K = 1,NROWS                                                ST3C1560
      TWIST(K) = 0.00                                                   ST3C1570
      TH(K) = 0.00                                                      ST3C1580
      THETS(K) = 0.00                                                   ST3C1590
      HO(K) = 0.00                                                      ST3C1600
      DUMB(K) = 0.0                                                     ST3C1601
      DO 320 N = 1,NCASES                                               ST3C1610
  310 TWIST(K) = TWIST(K) + TST(K,N) * FACT(N)                          ST3C1620
      TH(K) = TH(K) + THETA(K,N) * FACT(N)                              ST3C1630
      THETS(K) = THETS(K) + THS(K,N) * FACT(N)                          ST3C1640
      HO(K) = HO(K) + HL(K,N) * FACT(N)                                 ST3C1650
  320 CONTINUE                                                          ST3C1660
  330 CONTINUE                                                          ST3C1670
      DO 370 I = 1,NEWMAX                                               ST3C1680
  340 BTA(I) = 0.00                                                     ST3C1690
      EP(I) = 0.00                                                      ST3C1700
      DO 360 N = 1,NCASES                                               ST3C1710
  350 BTA(I) = BTA(I) + BETA(I,N) * FACT(N)                             ST3C1720
      EP(I) = EP(I) + EPS(I,N) * FACT(N)                                ST3C1730
  360 CONTINUE                                                          ST3C1740
  370 CONTINUE                                                          ST3C1750
C  COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 0                         ST3C1760
      ALPHA = 0.00                                                      ST3C1770
      IF(NOALFA .GT. 0) ALPHA = FACT(1)                                 ST3C1780
  380 CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,               ST3C1790
     1       XB,DEL,BTA,EP,CPO,CLO,CLGO,CLMUO,CMO,CMGO,CMMUO,CMTO,      ST3C1800
     2       XBCPO,XBCLO,CDIO,CDMUO,CDGO,CSO,CTO,NROWS,IHINGE)          ST3C1810
C  COMPUTE SECTIONAL VORTICITY FOR ALPHA = 0                            ST3C1820
  390 CALL SLOADG(CPO,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLGO,CGAMO,NROWS,IHINGE)ST3C1830
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 0                 ST3C1840
  400 CALL TREFTZ(Y,DELTA,CMU,CGAMO,ALFINO,NROWS,ISYMM)                 ST3C1850
C  COMPUTE SECTIONAL NONLINEAR CROSS-PRODUCT COEFFICIENTS               ST3C1860
      IF(NOALFA .EQ. 0) GO TO 420                                       ST3C1870
  410 CALL SLOADX(CPA,CPO,DEL,EP,CMU,TH,NW,NJ,IJ,                       ST3C1880
     1               CLGO,CDGX,CDMUX,CSX,CDIX,NROWS,LOGIC)              ST3C1890
C  COMPUTE TOTAL LINEAR COEFFICIENTS FOR ALPHA = 0                      ST3C1900
  420 CALL TLOADO(CREF,CCLG,CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,      ST3C1910
     1       CLLG,CLLJ,FACT,CCLGO,CCLJO,CCLO,CCMGO,CCMJO,CCMTO,CCMO,    ST3C1920
     2       CMGMCO,CMJMCO,CMTMCO,CMMCO,CXCPO,CXCLO,CXCPBO,CXCLBO,      ST3C1930
     3       CLLGO,CLLJO,CLLO,NCASES,ISYMM)                             ST3C1940
C  COMPUTE TOTAL NONLINEAR COEFFICIENTS FOR ALPHA = 0                   ST3C1950
  430 CALL TLOADN(AREA,CHORD,DELTA,Y,CMU,CDGO,CDMUO,CSO,CDIO,CLO,DUMB,  ST3C1960
     1          CCDGO,CCDJO,CCSO,CCDIO,CDITZO,DUMB,ALFINO,CCTO,CCJ(1),  ST3C1970
     2          CNJO,DUMMY,CNGO,CCYO,XLEAD,TANLE,XMC,NROWS,ISYMM,LOGIC) ST3C1980
C  COMPUTE TOTAL NONLINEAR CROSS-PRODUCT COEFFICIENTS                   ST3C1990
      IF(NOALFA .EQ. 0) GO TO 450                                       ST3C2000
  440 CALL TLOADN(AREA,CHORD,DELTA,Y,CMU,CDGX,CDMUX,CSX,CDIX,CL,CLO,    ST3C2010
     1          CCDGX,CCDJX,CCSX,CCDIX,CDITZX,ALFINF,ALFINO,DUMB,CCJ(1),ST3C2020
     2          DUMMY,CNJX,CNGX,CCYX,XLEAD,TANLE,XMC,NROWS,ISYMM,LOGIC) ST3C2030
C  PRINT THE SECTIONAL AND TOTAL COEFFICIENTS                           ST3C2040
  450 CALL HEADER(ILINES)                                               ST3C2050
      WRITE(6, 460 ) M                                                  ST3C2060
  460 FORMAT(1H0,39X,1H*,12(4H****)/                                    ST3C2070
     1          40X,43H*  WING SPANWISE LOADING FOR COMPOSITE CASE,I3,  ST3C2080
     2          3H  */40X,1H*,12(4H****))                               ST3C2090
      WRITE(6, 470 ) (N,N=1,10),(FACT(N),N=1,10)                        ST3C2100
  470 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(4X,2HA(,I1,1H),  ST3C2110
     1       2X),3X,2HA(,I2,1H)/10X,10F10.6/)                           ST3C2120
      WRITE(6, 480 )                                                    ST3C2130
  480 FORMAT(1H ,20X,29H........... LIFT ............,10H   *  *   ,    ST3C2140
     1       39H........... PITCHING MOMENT ..........,10H   *  *   ,   ST3C2150
     2       20H ... LIFT CENTER .../ 2X,7HSECTION,5X,1HY,8X,4HCLGO,6X, ST3C2160
     3       5HCLMUO,5X,3HCLO,4X,10H   *  *   ,3X,4HCMGO,6X,5HCMMUO,5X, ST3C2170
     4       4HCMTO,6X,3HCMO,4X,10H   *  *   ,3X,6HXCPO/C,4X,6HXCLO/C)  ST3C2180
      IF(NOALFA .GT. 0) WRITE(6, 490 )                                  ST3C2190
  490 FORMAT(1H ,                                                       ST3C2200
     1       22X,4HCLGA,6X,5HCLMUA,5X,3HCLA,4X,10H   *  *   ,3X,4HCMGA, ST3C2210
     2       6X,5HCMMUA,5X,4HCMTA,6X,3HCMA,4X,10H   *  *   ,3X,6HXCPA/C,ST3C2220
     3       4X,6HXCLA/C)                                               ST3C2230
      DO 520 K = 1,NROWS                                                ST3C2240
      WRITE(6, 500 ) K,Y(K),CLGO(K),CLMUO(K),CLO(K),CMGO(K),CMMUO(K),   ST3C2250
     1               CMTO(K),CMO(K),XBCPO(K),XBCLO(K)                   ST3C2260
  500 FORMAT(1H ,I5,4X,4F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)ST3C2270
      IF(NOALFA .GT. 0) WRITE(6, 510 ) CLG(K),CLMU(K),CL(K),            ST3C2280
     1               CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)        ST3C2290
  510 FORMAT(1H ,19X,3F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)  ST3C2300
  520 CONTINUE                                                          ST3C2310
      WRITE(6, 530 ) CCLGO,CCLJO,CCLO,CCMGO,CCMJO,CCMTO,CCMO,CXCPO,CXCLOST3C2320
  530 FORMAT(1H , 9X,4(10H----------),10H   *  *   ,4(10H----------),   ST3C2330
     1        10H   *  *   ,2(10H----------)/13X,5HTOTAL,2X,3F10.6,     ST3C2340
     2        10H   *  *   ,4F10.6,10H  (APEX)  , 2F10.6)               ST3C2350
      IF(NOALFA .GT. 0) WRITE(6, 540 ) CCLG(1),CCLJ(1),CCL(1),          ST3C2360
     1              CCMG(1),CCMJ(1),CCMT(1),CCM(1),CXCP(1),CXCL(1)      ST3C2370
  540 FORMAT(1H ,19X,3F10.6,                                            ST3C2380
     1        10H   *  *   ,4F10.6,10H  (APEX)  , 2F10.6)               ST3C2390
      WRITE(6, 550 ) CMGMCO,CMJMCO,CMTMCO,CMMCO,CXCPBO,CXCLBO           ST3C2400
  550 FORMAT(1H ,49X,                                                   ST3C2410
     1        10H   *  *   ,4F10.6,10H  (XMC)   , 2F10.6)               ST3C2420
      IF(NOALFA .GT. 0) WRITE(6, 560 )                                  ST3C2430
     1              CMGMC(1),CMJMC(1),CMTMC(1),CMMC(1),CXCPB(1),CXCLB(1)ST3C2440
  560 FORMAT(1H ,49X,10H   *  *   ,4F10.6,10H  (XMC)   , 2F10.6)        ST3C2450
      WRITE(6, 570 )                                                    ST3C2460
  570 FORMAT(1H1,25X,9(4H....),15H INDUCED DRAG .,9(4H....)/            ST3C2470
     1     7X,7HSECTION,6X,1HY,7X,4HCDGO,7X,5HCDMUO,6X,3HCSO,8X,3HCDO,  ST3C2480
     2   8X,6HGAMMAO,5X,6HALFINO,5X,3HCTO,8X,3HCMU)                     ST3C2490
      IF(NOALFA .GT. 0) WRITE(6, 580 )                                  ST3C2500
  580 FORMAT(1H ,27X,4HCDGA,7X,5HCDMUA,6X,3HCSA,8X,3HCDA/               ST3C2510
     1   28X,5HCDGA2,6X,6HCDMUA2,5X,4HCSA2,7X,4HCDA2,7X,6HGAMMAA,       ST3C2520
     2   5X,6HALFINA)                                                   ST3C2530
      DO 610 K = 1,NROWS                                                ST3C2540
      WRITE(6, 590 ) K,Y(K),CDGO(K),CDMUO(K),CSO(K),CDIO(K),            ST3C2550
     1       CGAMO(K),ALFINO(K),CTO(K),CMU(K)                           ST3C2560
  590 FORMAT(1H , 6X,I4 ,4X,F10.6,8F11.7)                               ST3C2570
      IF(NOALFA .GT. 0) WRITE(6, 600 )                                  ST3C2580
     1       CDGX(K),CDMUX(K),CSX(K),CDIX(K),CDG(K),CDMU(K),CS(K),      ST3C2590
     2       CDI(K),CGAM(K),ALFINF(K)                                   ST3C2600
  600 FORMAT(1H ,24X,4F11.7/25X,6F11.7)                                 ST3C2610
  610 CONTINUE                                                          ST3C2620
      WRITE(6, 620 ) CCDGO,CCDJO,CCSO,CCDIO,CDITZO,CCTO,CCJ(1)          ST3C2630
  620 FORMAT(1H ,24X,8(11H-----------)/18X,5HTOTAL,2X,4F11.7,11X,3F11.7)ST3C2640
      IF(NOALFA .GT. 0) WRITE(6, 630 ) CCDGX,                           ST3C2650
     1   CCDJX,CCSX,CCDIX,CDITZX,CCDG(1),CCDJ(1),CCS(1),CCDI(1),CDITZ(1)ST3C2660
  630 FORMAT(1H ,24X,4F11.7,11X,F11.7/25X,4F11.7,11X,F11.7)             ST3C2670
  PRINT A TABLE OF ALL TOTAL COEFFICIENTS FOR ALPHA = 0,ALPHA,ALPHA**2  ST3C2680
      CALL HEADER(ILINES)                                               ST3C2690
      WRITE(6, 640 )M                                                   ST3C2700
  640 FORMAT(1H0,32X,16(4H****)/                                        ST3C2710
     1        33X,43H*  WING TOTAL AERODYNAMIC COEFFICIENTS FOR ,       ST3C2720
     2        15HCOMPOSITE CASE ,I3,3H  */33X,16(4H****)//              ST3C2730
     3        42X,7HALPHA=0,8X,5HALPHA,10X,8HALPHA**2)                  ST3C2740
      IF(NOALFA .GT. 0) GO TO 680                                       ST3C2750
  650 FORMAT(1H0,25X,A8,F15.7)                                          ST3C2760
      WRITE(6, 650 ) COEFF(1),CCLGO,COEFF(2),CCLJO,COEFF(3),CCLO,       ST3C2770
     1        COEFF(4),CCDGO,COEFF(5),CCDJO,COEFF(6),CCSO,COEFF(7),CCDIO,ST3C2780
     2        COEFF(8),CDITZO,COEFF(9),CCJ(1),COEFF(10),CCMGO,          ST3C2790
     3        COEFF(11),CCMJO,COEFF(12),CCMTO,COEFF(13),CCMO,           ST3C2800
     4        COEFF(14),CXCPO,COEFF(15),CXCLO,COEFF(16),CXCPB,          ST3C2810
     5        COEFF(17),CXCLB,COEFF(18),CMGMCO,COEFF(19),CMJMCO,        ST3C2820
     6        COEFF(20),CMTMCO,COEFF(21),CMMCO,COEFF(22),CLLGO,         ST3C2830
     7        COEFF(23),CLLJO,COEFF(24),CLLO,COEFF(25),CNJO,            ST3C2840
     8        COEFF(26),CNGO,COEFF(27),CCYO                             ST3C2850
      GO TO 700                                                         ST3C2860
  660 FORMAT(1H0,25X,A8,3F15.6)                                         ST3C2870
  670 FORMAT(1H0,25X,A8,1X,3F15.7)                                      ST3C2880
  680 WRITE(6, 660 )COEFF( 1), CCLGO,CCLG(1)                            ST3C2890
      WRITE(6, 660 )COEFF( 2), CCLJO,CCLJ(1)                            ST3C2900
      WRITE(6, 660 )COEFF( 3), CCLO ,CCL(1)                             ST3C2910
```

```
      WRITE(6, 670 )COEFF( 4), CCDG0,CCDGX,CCDG(I)                       ST3C2920
      WRITE(6, 670 )COEFF( 5), CCDJ0,CCDJX,CCDJ(I)                       ST3C2930
      WRITE(6, 670 )COEFF( 6), CCS0 ,CCSX ,CCS(I)                        ST3C2940
      WRITE(6, 670 )COEFF( 7), CCDI0,CCDIX,CCDI(I)                       ST3C2950
      WRITE(6, 670 )COEFF( 8), CDITZ0,CDITZX,CDITZ(I)                    ST3C2960
      WRITE(6, 670 )COEFF( 9), CCJ(I)                                    ST3C2970
      WRITE(6, 660 )COEFF(10), CCMG0,CCMG(I)                             ST3C2980
      WRITE(6, 660 )COEFF(11), CCMJ0,CCMJ(I)                             ST3C2990
      WRITE(6, 660 )COEFF(12), CCMT0,CCMT(I)                             ST3C3000
      WRITE(6, 660 )COEFF(13), CCM0 ,CCM(I)                              ST3C3010
      WRITE(6, 660 )COEFF(14), CXCP0,CXCP(I)                             ST3C3020
      WRITE(6, 660 )COEFF(15), CXCL0,CXCL(I)                             ST3C3030
      WRITE(6, 660 )COEFF(16), CXCPB0,CXCPB(I)                           ST3C3040
      WRITE(6, 660 )COEFF(17), CXCLB0,CXCLB(I)                           ST3C3050
      WRITE(6, 660 )COEFF(18), CMGMC0,CMGMC(I)                           ST3C3060
      WRITE(6, 660 )COEFF(19), CMJMC0,CMJMC(I)                           ST3C3070
      WRITE(6, 660 )COEFF(20), CMTMC0,CMTMC(I)                           ST3C3080
      WRITE(6, 660 )COEFF(21), CMMC0,CMMC(I)                             ST3C3090
      WRITE(6, 660 )COEFF(22), CLLG0,CLLG(I)                             ST3C3100
      WRITE(6, 660 )COEFF(23), CLLJ0,CLLJ(I)                             ST3C3110
      WRITE(6, 660 )COEFF(24), CLL0,CLL(I)                               ST3C3120
      WRITE(6, 660 )COEFF(25), CNJ0,CNJX,CNJJ(I)                         ST3C3130
      WRITE(6, 660 )COEFF(26), CNG0,CNGX,CNG(I)                          ST3C3140
      WRITE(6, 660 )COEFF(27), CCY0,CCYX,CCY(I)                          ST3C3150
C  COMPUTE AND PRINT A TABLE OF THE VARIATION OF THE TOTAL COEFFICIENTS ST3C3160
C  WITH ANGLE OF ATTACK                                                 ST3C3170
  690 CALL TABLE(CCL0,CCL(I),CMMC0,CMMC(I),CLL0,CLL(I),CLLJ0,CLLJ(I),   ST3C3180
     1 CDITZ0,CDITZX,CDITZ(I),CCJ(I),CNG0,CNGX,CNG(I),CNJ0,CNJX,CNJJ(I),ST3C3190
     2           CCY0,CCYX,CCY(I),M)                                    ST3C3200
C  SAVE THE TOTAL STATIC WING COEFFICIENTS ON UNIT 9                    ST3C3210
      IF((NROWH+NROWV+NSEG) .EQ. 0) GO TO 710                           ST3C3220
  700 PUT(2) = CCL(I)                                                   ST3C3230
      PUT(7) = CMMC(I)                                                  ST3C3240
      PUT(12) = CLL(I)                                                  ST3C3250
      PUT(18) = CDITZ(I)                                                ST3C3260
      PUT(21) = CNG0 + CNJ0                                             ST3C3270
      PUT(22) = CNGX + CNJX                                             ST3C3280
      PUT(23) = CNG(I) + CNJJ(I)                                        ST3C3290
      PUT(28) = CCY(I)                                                  ST3C3300
      CALL WRITMS(9,PUT,102,IWRITE)                                     ST3C3310
  710 CONTINUE                                                          ST3C3320
      END                                                               ST3C3330
```

```
*DECK TABLE                                                             TABL0010
      SUBROUTINE TABLE(CL0,CLA,CM0,CMA,CLL0,CLLA,CLLJ0,CLLJA,           TABL0020
     1        CDI0,CDIX,CDIA2,CJ,CNG0,CNGX,CNGA2,CNJ0,CNJX,CNJA2,       TABL0030
     2        CY0,CYX,CYA2,MCASE)                                       TABL0040
C                                                                       TABL0050
C  THIS SUBROUTINE COMPUTES AND PRINTS A TABLE OF TOTAL COEFFICIENTS    TABL0060
C  FOR A RANGE OF ANGLES OF ATTACK                                      TABL0070
C                                                                       TABL0080
C  COMPUTE THE COEFFICIENTS                                             TABL0090
      CALL HEADER(ILINES)                                               TABL0100
      WRITE(6, 20 ) MCASE                                               TABL0110
   20 FORMAT(1H0,34X,61(1H*)/35X,35H*  TABULATED WING TOTAL COEFFICIENT,TABL0120
     120HS FOR COMPOSITE CASE,I3,3H  */35X,61(1H*)//                    TABL0130
     2  1X,5HALPHA,7X,3HCCL,8X,6HCCL**2,5X,7HCCM(MC),4X,4HCLLJ,7X,3HCLL,TABL0140
     3  12X,5HCDITZ,5X,2HCT,13X,3HCNJ,8X,2HCN,9X,3HCCY)                 TABL0150
      ALPHA = -11.00                                                    TABL0160
      DO 70  M = 1,41                                                   TABL0170
   30 ALPHA = ALPHA + 1.00                                              TABL0180
      ALPHA2 = ALPHA * ALPHA                                            TABL0190
C  LINEAR COEFFICIENTS                                                  TABL0200
   40 CL = CL0 + CLA * ALPHA                                            TABL0210
      CL2 = CL * CL                                                     TABL0220
      CM = CM0 + CMA * ALPHA                                            TABL0230
      CLLJ = CLLJ0 + CLLJA*ALPHA                                        TABL0240
      CLL = CLL0 + CLLA * ALPHA                                         TABL0250
C  NONLINEAR COEFFICIENTS                                               TABL0260
   50 CDI = CDI0 + CDIX * ALPHA + CDIA2 * ALPHA2                        TABL0270
      CT = CJ - CDI                                                     TABL0280
      CNG = CNG0 + CNGX*ALPHA + CNGA2*ALPHA2                            TABL0290
      CNJ = CNJ0 + CNJX*ALPHA + CNJA2*ALPHA2                            TABL0300
      CN = CNJ + CNG                                                    TABL0310
      CY = CY0 + CYX * ALPHA + CYA2 * ALPHA2                            TABL0320
C  PRINT THE TABLE                                                      TABL0330
      WRITE(6, 60 ) ALPHA,CL,CL2,CM,CLLJ,CLL,CDI,CT,CNJ,CN,CY           TABL0340
   60 FORMAT(1H ,F5.1,4X,5F11.7,4X,2F11.7,4X,3F11.7)                    TABL0350
   70 CONTINUE                                                          TABL0360
      RETURN                                                            TABL0370
      END                                                               TABL0380
```

```
*DECK STG3GE                                                            S3GE0010
      OVERLAY(OVLY,3,2)                                                 S3GE0020
      PROGRAM STG3GE                                                    S3GE0030
C                                                                       S3GE0040
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE, SPANWISE, AND     S3GE0050
C  TOTAL LOADING FOR THE M-TH COMPOSITE CASE IN GROUND PROXIMITY.       S3GE0060
C                                                                       S3GE0070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                S3GE0080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            S3GE0090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) S3GE0100
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     S3GE0110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),KI(400),DEL(400),  S3GE0120
     1              D(40),KK(400),ITYPE(400)                            S3GE0130
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             S3GE0140
      COMMON/FCASE2/IST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     S3GE0150
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTS3GE0160
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    S3GE0170
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           S3GE0180
      COMMON/COMPOS/FACTOR(10,24),NCC                                   S3GE0190
      COMMON/SOLV1/GAMMA(400,10)                                        S3GE0200
      COMMON/SOLV3/GAMSUM(400)                                          S3GE0210
      COMMON/LOAD1/TWIST(40),H0(40),TH(40),THETS(40),                   S3GE0220
     1              BTA(400),EP(400),CP0(400)                           S3GE0230
      COMMON/LOAD2/CLGA(40),CLMUA(40),CLA(40),CDMUA(40),CDGA(40),       S3GE0240
     1        CDIA(40),CSA(40),CMGA(40),CMMUA(40),CMTA(40),CMA(40),     S3GE0250
     2        XBCPA(40),XBCLA(40)                                       S3GE0260
      COMMON/LOAD3/CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),         S3GE0270
     1        CCMT(10),CCM(10),CMGMC(10),CMJMC(10),CMTMC(10),CMMC(10),  S3GE0280
     2        CXCP(10),CXCL(10),CCJ(10),CCDG(10),CCDJ(10),CCS(10),CCDI(10),S3GE0290
     3        CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNJJ(10),CNG(10),S3GE0300
     4        CCY(10),CXCPB(10),CXCLB(10)                               S3GE0310
      COMMON/LOAD4/CLG0(40),CLMU0(40),CL0(40),CDMU0(40),CDG0(40),       S3GE0320
     1        CDI0(40),CS0(40),CT0(40),CMG0(40),CMMU0(40),CMT0(40),CM0(40),S3GE0330
     2        XBCP0(40),XBCL0(40),FACT(10)                              S3GE0340
      COMMON/LOAD5/CGAM(40),CGAM0(40),ALFINF(40),ALFIN0(40),DUMMY(40)   S3GE0350
      COMMON/LOAD6/COEFF(27)                                            S3GE0360
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HT,HH,VL,VH,     S3GE0370
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    S3GE0380
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES              S3GE0390
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,KBB (41),ZCL(41),  S3GE0400
     1              RADIUS(41),AMAJ(41),AMIN(41),CSAA (41),             S3GE0410
     2              V(40,7),W(40,7)                                     S3GE0420
      COMMON/KEEPEM/PUT(102)                                            S3GE0430
      DIMENSION CLGA2(40),CMGA2(40),XBCPA2(40),CDGA2(40),CDMUA2(40),    S3GE0440
     1          CSA2(40),CDIA2(40),CDGA3(40),CPA(400),CPA2(400)         S3GE0450
      DIMENSION ONSET(400),GALPHA(400)                                  S3GE0460
      EQUIVALENCE (GAMSUM(1),ONSET(1)),(GALPHA(1),GAMMA(1,1))           S3GE0470
      LOGICAL PRINT                                                     S3GE0480
      PRINT = (IPRINT .GT. 0)                                           S3GE0490
      M = KCASE                                                         S3GE0510
      DUM = 0.0                                                         S3GE0520
      DUMB = 0.0                                                        S3GE0530
      DO 20  K = 1,NROWS                                                S3GE0540
   20 DUMMY(K) = 0.0                                                    S3GE0550
      I = 0                                                             S3GE0560
      II = NWT                                                          S3GE0570
      NCL = NCASES + 1                                                  S3GE0580
      CALL HEADER(ILINES)                                               S3GE0590
      ILINES = ILINES + 9                                               S3GE0600
      WRITE(6, 30 ) M                                                   S3GE0610
   30 FORMAT(1H0,29X,14(5H*****)/                                       S3GE0620
     1        30X,44H*  WING CHORDWISE LOADING FOR COMPOSITE CASE,I3,   S3GE0630
     2        23H IN GROUND PROXIMITY  */30X,14(5H*****))               S3GE0640
      WRITE(6, 40 )(N,N=1,10),(FACTOR(N,M),N=1,10)                      S3GE0650
   40 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),  S3GE0660
     1       3X),5X,2HA(,I2,1H)/10X,10F12.6)                            S3GE0670
      WRITE(6, 50 )                                                     S3GE0680
   50 FORMAT(1H0,7X,47H*** NOTE ***  EACH LEADING EDGE CP VALUE IS THE, S3GE0690
     1        43H AVERAGE VALUE OF THE SINGULAR DISTRIBUTION,/,         S3GE0700
     2        22X,41HDO NOT PLOT THESE LOADING POINTS DIRECTLY)         S3GE0710
      DO 60 N = 1,10                                                    S3GE0720
   60 FACT(N) = FACTOR(N,M)                                             S3GE0730
   70 DO 270 K = 1,NROWS                                                S3GE0740
C  ON THE WING                                                          S3GE0750
      NWK = NW(K)                                                       S3GE0760
      DO 120 L = 1,NWK                                                  S3GE0770
      I = I + 1                                                         S3GE0780
      IREAD = 2 * I - 1                                                 S3GE0790
      GSUM = 0.0                                                        S3GE0800
      DO 80 N = 1,NCASES                                                S3GE0810
   80 GSUM  = GSUM  + GAMMA(I,N) * FACT(N)                              S3GE0820
C  READ THE U PERTURBATION VELOCITIES                                   S3GE0830
      U = 0.0                                                           S3GE0840
      CALL READMS(9,ONSET,NEWMAX,IREAD)                                 S3GE0850
      DO 100 N = 1,NCASES                                               S3GE0860
      DO 90  J = 1,NEWMAX                                               S3GE0870
   90 U = U + GAMMA(J,N) * FACT(N) * ONSET(J)                           S3GE0880
  100 CONTINUE                                                          S3GE0890
      UALPHA = 0.0                                                      S3GE0900
      DO 110 J = 1,NEWMAX                                               S3GE0910
  110 UALPHA = UALPHA + GAMMA(J,1) * ONSET(J)                           S3GE0920
C  FORM THE 3 PARTS OF THE PRESSURE COEFFICIENT                         S3GE0930
      CP0(I) = 2.0 * (1.0 + U) * GSUM                                   S3GE0940
      CPA(I) = 2.0 * ((UALPHA * GSUM) + (1.0 + U) * GAMMA(I,1))         S3GE0950
      CPA2(I) = 2.0 * UALPHA * GAMMA(I,1)                               S3GE0960
  120 CONTINUE                                                          S3GE0970
      IF(PRINT)GO TO 200                                                S3GE0980
      J1 = IW(K)                                                        S3GE0990
      J2 = IW(K) + NWK - 1                                              S3GE1000
      NEXT = 3 + 4 * (NWK/10 + 1)                                       S3GE1010
      IF(CMU(K) .GT. 0.0) NEXT = NEXT + 4 * (NJ(K)/10 + 1) + 1          S3GE1020
      ILINES = ILINES + NEXT                                            S3GE1030
      IF(ILINES .LT. 56)GO TO 140                                       S3GE1040
      WRITE(6, 130 )                                                    S3GE1050
  130 FORMAT(1H1)                                                       S3GE1060
      ILINES = 1                                                        S3GE1070
  140 WRITE(6, 150 )K,Y(K),CHORD(K)                                     S3GE1080
  150 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY = ,F10.6,5X,8HCHORD = ,F10.6/  S3GE1090
     1        2X,4HWING)                                                S3GE1100
      WRITE(6, 160 )(XB(J),J=J1,J2)                                     S3GE1110
  160 FORMAT(1H ,7X,2HXB,10F12.6,3(/10X,10F12.6))                       S3GE1120
      WRITE(6, 170 )(CP0(J),J=J1,J2)                                    S3GE1130
  170 FORMAT(1H ,6X,3HCP0,10F12.6,3(/10X,10F12.6))                      S3GE1140
      WRITE(6, 180 )(CPA(J),J=J1,J2)                                    S3GE1150
  180 FORMAT(1H ,6X,3HCPA,10F12.6,3(/10X,10F12.6))                      S3GE1160
      WRITE(6, 190 )(CPA2(J),J=J1,J2)                                   S3GE1170
  190 FORMAT(1H ,5X,4HCPA2,10F12.6,3(/10X,10F12.6))                     S3GE1180
C  ON THE JET                                                           S3GE1190
  200 IF(CMU(K) .LT. 0.0001)GO TO 270                                   S3GE1200
      NJK = NJ(K)                                                       S3GE1210
      DO 250 L = 1,NJK                                                  S3GE1220
      II = II + 1                                                       S3GE1230
      IREAD = 2 * II - 1                                                S3GE1240
      GSUM = 0.0                                                        S3GE1250
      DO 210 N = 1,NCASES                                               S3GE1260
  210 GSUM = GSUM + GAMMA(II,N) * FACT(N)                               S3GE1270
C  READ THE U PERTURBATION VELOCITIES                                   S3GE1280
      U = 0.0                                                           S3GE1290
      CALL READMS(9,ONSET,NEWMAX,IREAD)                                 S3GE1300
      DO 230 N = 1,NCASES                                               S3GE1310
      DO 220 J = 1,NEWMAX                                               S3GE1320
  220 U = U + GAMMA(J,N) * FACT(N) * ONSET(J)                           S3GE1330
  230 CONTINUE                                                          S3GE1340
      UALPHA = 0.0                                                      S3GE1350
      DO 240 J = 1,NEWMAX                                               S3GE1360
  240 UALPHA = UALPHA + GAMMA(J,1) * ONSET(J)                           S3GE1370
C  FORM THE 3 PARTS OF THE PRESSURE COEFFICIENT                         S3GE1380
      CP0(II) = 2.0 * (1.0 + U) * GSUM                                  S3GE1390
      CPA(II) = 2.0 * ((UALPHA * GSUM) + (1.0 + U) * GAMMA(II,1))       S3GE1400
      CPA2(II) = 2.0 * UALPHA * GAMMA(II,1)                             S3GE1410
  250 CONTINUE                                                          S3GE1420
      IF(PRINT)GO TO 270                                                S3GE1430
      J1 = IJ(K)                                                        S3GE1440
      J2 = IJ(K) + NJK - 1                                              S3GE1450
      WRITE(6, 260 )                                                    S3GE1460
  260 FORMAT(1H ,4H JET)                                                S3GE1470
      WRITE(6, 160 )(XB(J),J=J1,J2)                                     S3GE1480
      WRITE(6, 170 )(CP0(J),J=J1,J2)                                    S3GE1490
      WRITE(6, 180 )(CPA(J),J=J1,J2)                                    S3GE1500
      WRITE(6, 190 )(CPA2(J),J=J1,J2)                                   S3GE1510
  270 CONTINUE                                                          S3GE1520
      DO 290 I=1,NEWMAX                                                 S3GE1530
```

```
      GAMSUM(I) = 0.0                                                   S3GE1540
      DO 280 N = 1,NCASES                                               S3GE1550
  280 GAMSUM(I) = GAMSUM(I) + GAMMA(I,N) * FACT(N)                      S3GE1560
  290 CONTINUE                                                          S3GE1570
C  CALCULATE THE SECTIONAL COEFFICIENTS FOR ALPHA**2                    S3GE1580
      ALPHA = 0.0                                                       S3GE1590
      DO 300 K = 1,NROWS                                                S3GE1600
      TH(K) = THETA(K,1)                                                STGE1610
      THETS(K) = 0.0                                                    S3GE1620
  300 TWIST(K) = 0.0                                                    S3GE1630
      DO 310 I = 1,NEWMAX                                               S3GE1640
      BTA(I) = BETA(I,1)                                                S3GE1650
  310 EP(I) = EPS(I,1)                                                  S3GE1660
      CALL LOADG1(ALPHA,IJ,NW,NJ,CHORD,DUMMY,TH,THETS,TWIST,XB,DEL,BTA, S3GE1670
     1       EP,CPA2,CLGA2,CLGA2,DUMMY,CMGA2,CMGA2,DUMMY,DUMMY          S3GE1680
     2       XBCPA2,XBCPA2,NROWS)                                       S3GE1690
      IDENT = 1                                                         S3GE1700
      CALL LOADG2(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,   S3GE1710
     1       EP,GALPHA,GAMSUM,CDGA2,CDMUA2,CDIA2,CSA2,CTO,NROWS,IDENT)  S3GE1720
C  CALCULATE THE SECTIONAL COEFFICIENTS FOR ALPHA                       S3GE1730
      ALPHA = 1.0                                                       S3GE1740
      DO 320 K = 1,NROWS                                                S3GE1750
      TWIST(K) = TST(K,1)                                               S3GE1760
      TH(K) = THETA(K,1)                                                S3GE1770
  320 THETS(K) = THS(K,1)                                               S3GE1780
      CALL LOADG1(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,   S3GE1790
     1       EP,CPA,CLGA,CLA,CLMUA,CMGA,CMA,CMMUA,CMTA,XBCPA,XBCLA,NROWS) S3GE1800
C  MODULATE AND SUM THE FUNDAMENTAL CASE SECTIONAL PARAMETERS           S3GE1810
      DO 340 K = 1,NROWS                                                S3GE1820
      TWIST(K) = 0.0                                                    S3GE1830
      TH(K) = 0.0                                                       S3GE1840
      THETS(K) = 0.0                                                    S3GE1850
      HO(K) = 0.0                                                       S3GE1860
      DO 330 N = 1,NCASES                                               S3GE1870
      TWIST(K) = TWIST(K) + TST(K,N) * FACT(N)                          S3GE1880
      TH(K) = TH(K) + THETA(K,N) * FACT(N)                              S3GE1890
      THETS(K) = THETS(K) + THS(K,N) * FACT(N)                          S3GE1900
  330 HO(K) = HO(K) + HL(K,N) * FACT(N)                                 S3GE1910
  340 CONTINUE                                                          S3GE1920
      DO 360 I = 1,NEWMAX                                               S3GE1930
      BTA(I) = 0.0                                                      S3GE1940
      EP(I) = 0.0                                                       S3GE1950
      DO 350 N = 1,NCASES                                               S3GE1960
      BTA(I) = BTA(I) + BETA(I,N) * FACT(N)                             S3GE1970
  350 EP(I) = EP(I) + EPS(I,N) * FACT(N)                                S3GE1980
  360 CONTINUE                                                          S3GE1990
      IDENT = 2                                                         S3GE2000
      CALL LOADG2(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,   S3GE2010
     1       EP,GALPHA,GAMSUM,CDGA,CDMUA,CDIA,CSA,CTO,NROWS,IDENT)      S3GE2020
      CALL GLOADG(GALPHA,DEL,BTA,CHORD,D,CMU,NW,IW,NJ,IJ,CGAM,NROWS)    S3GE2030
C  CALCULATE THE SECTIONAL COEFFICIENTS FOR ALPHA = 0                   S3GE2050
      ALPHA = 0.0                                                       S3GE2060
      IF(NOALFA .GT. 0) ALPHA = FACT(1)                                 S3GE2061
      CALL LOADG1(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,   S3GE2070
     1       EP,CPO,CLGO,CLO,CLMUO,CMGO,CMO,CMMUO,CMTO,XBCPO,XBCLO,NROWS) S3GE2080
      ALPHA = 0.0                                                       S3GE2081
      IDENT = 3                                                         S3GE2090
      CALL LOADG2(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,   S3GE2100
     1       EP,GAMSUM,GALPHA,CDGO,CDMUO,CDIO,CSO,CTO,NROWS,IDENT)      S3GE2110
      CALL GLOADG(GAMSUM,DEL,BTA,CHORD,D,CMU,NW,IW,NJ,IJ,CGAMO,NROWS)   S3GE2130
      CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)                  S3GE2140
      CALL TREFTZ(Y,DELTA,CMU,CGAMO,ALFINO,NROWS,ISYMM)                 S3GE2150
C  CALCULATE THE TOTAL COEFFICIENTS FOR LIFT, MOMENT, AND LIFT CENTER   S3GE2160
C                                                                       S3GE2170
C  ALPHA**2 TERMS                                                       S3GE2180
      DO 370 I = 1,NEWMAX                                               S3GE2190
  370 BTA(I) = 0.0                                                      S3GE2200
      ALPHA = 0.0                                                       S3GE2210
      CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,DUMMY,XB,XLEAD,BTA,  S3GE2220
     1       CLGA2,DUMMY,CMGA2,DUMMY,DUMMY,CMU,CCLG(3),DUM,CCLG(3),CCMG(3)S3GE2230
     2   ,DUM,DUM,CCMG(3),CMGMC(3),DUM,DUM,CMGMC(3),                    S3GE2240
     3       CXCP(3),CXCP(3),CXCPB(3),CXCPB(3),CCJ(3),CLLG(3),CLLJ(3),  S3GE2250
     4       CLL(3),ITYPE,IW,NW,NROWS,ISYMM)                            S3GE2260
C  ALPHA TERMS                                                          S3GE2270
      ALPHA = 1.0                                                       S3GE2280
      CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,DUMMY,XB,XLEAD,BTA,  S3GE2290
     1       CLGA,CLMUA,CMGA,CMMUA,CMTA,CMU,CCLG(2),CCLJ(2),CCL(2),CCMG(2)S3GE2300
     2       CCMJ(2),CCMT(2),CCM(2),CMGMC(2),CMJMC(2),CMTMC(2),CMMC(2), S3GE2310
     3       CXCP(2),CXCL(2),CXCPB(2),CXCLB(2),CCJ(2),CLLG(2),CLLJ(2),  S3GE2320
     4       CLL(2),ITYPE,IW,NW,NROWS,ISYMM)                            S3GE2330
C  ZERO TERMS                                                           S3GE2340
      ALPHA = 0.0                                                       S3GE2350
      DO 390 I = 1,NEWMAX                                               S3GE2360
      BTA(I) = 0.0                                                      S3GE2370
      DO 380 N = 1,NCASES                                               S3GE2380
  380 BTA(I) = BTA(I) + BETA(I,N) * FACT(N)                             S3GE2390
  390 CONTINUE                                                          S3GE2400
      CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,HO,XB,XLEAD,BTA,     S3GE2410
     1       CLGO,CLMUO,CMGO,CMMUO,CMTO,CMU,CCLG(1),CCLJ(1),CCL(1),CCMG(1)S3GE2420
     2       CCMJ(1),CCMT(1),CCM(1),CMGMC(1),CMJMC(1),CMTMC(1),CMMC(1), S3GE2430
     3       CXCP(1),CXCL(1),CXCPB(1),CXCLB(1),CCJ(1),CLLG(1),CLLJ(1),  S3GE2440
     4       CLL(1),ITYPE,IW,NW,NROWS,ISYMM)                            S3GE2450
C  CALCULATE THE TOTAL DRAG COEFFICIENTS                                S3GE2460
      LOGIC = 1                                                         S3GE2470
C  ALPHA**2 TERMS                                                       S3GE2480
      CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGA2,CDMUA2,CSA2,CDIA2,CLA,   S3GE2490
     1       DUMMY,CCDG(3),CCDJ(3),CCS(3),CCDI(3),CDITZ(3),DUMMY,ALFINF, S3GE2500
     2       DUMB,CCJ(3),DUM,CNJ(3),CNG(3),CCY(3),XLEAD,TANLE,XMC,NROWS, S3GE2510
     3       ISYMM,LOGIC)                                               S3GE2520
C  ALPHA TERMS                                                          S3GE2530
      CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGA,CDMUA,CSA,CDIA,CLA,       S3GE2540
     1       CLO,CCDG(2),CCDJ(2),CCS(2),CCDI(2),CDITZ(2),ALFINF,ALFINO, S3GE2550
     2       DUMB,CCJ(2),DUM,CNJ(2),CNG(2),CCY(2),XLEAD,TANLE,XMC,NROWS, S3GE2560
     3       ISYMM,LOGIC)                                               S3GE2570
C  ZERO TERMS                                                           S3GE2580
      CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGO,CDMUO,CSO,CDIO,CLO,       S3GE2590
     1       DUMMY,CCDG(1),CCDJ(1),CCS(1),CCDI(1),CDITZ(1),DUMMY,ALFINO, S3GE2600
     2       CT,CCJ(1),CNJ(1),DUM,CNG(1),CCY(1),XLEAD,TANLE,XMC,NROWS,  S3GE2610
     3       ISYMM,LOGIC)                                               S3GE2620
C                                                                       S3GE2630
C  PRINT THE SECTIONAL AND TOTAL COEFFICIENTS                           S3GE2640
      CALL HEADER(ILINES)                                               S3GE2650
      WRITE(6, 400 )M                                                   S3GE2660
  400 FORMAT(1H0,29X,16(4H****),5H*****/                                S3GE2670
     1          30X,43H*  WING SPANWISE LOADING FOR COMPOSITE CASE,I3,  S3GE2680
     2          23H IN GROUND PROXIMITY  */30X,16(4H****),5H*****)      S3GE2690
      WRITE(6, 40 )(N,N=1,10),(FACT(N),N=1,10)                          S3GE2700
      WRITE(6, 410 )                                                    S3GE2710
  410 FORMAT(1H0,20X,29H........... LIFT ...........,11H   *  *    ,    S3GE2720
     1          39H........... PITCHING MOMENT ...........,10H   *  *  , S3GE2730
     2          20H ... LIFT CENTER .../,2X,7HSECTION,5X,1HY,8X,4HCLGO,6X, S3GE2740
     3          5HCLMUO,5X,3HCLO,4X,10H   *  *   ,3X,4HCMGO,6X,5HCMMUO,5X, S3GE2750
     4          4HCMTO,6X,3HCMO,4X,10H   *  *   ,3X,6HXCPO/C,4X,6HXCLO/C,/,S3GE2760
     5          23X,4HCLGA,6X,5HCLMUA,5X,3HCLA,4X,10H   *  *   ,3X,4HCMGA, S3GE2770
     6          6X,5HCMMUA,5X,4HCMTA,6X,3HCMA,4X,10H   *  *   ,/,       S3GE2780
     7          22X,5HCLGA2,15X,4HCLA2,4X,10H   *  *   ,2X,             S3GE2790
     8           5HCMGA2,25X,4HCMA2,4X,10H   *  *   )                   S3GE2800
      DO 450 K = 1,NROWS                                                S3GE2810
      WRITE(6, 420 )K,Y(K),CLGO(K),CLMUO(K),CLO(K),CMGO(K),CMMUO(K),    S3GE2820
     1              CMTO(K),CMO(K),XBCPO(K),XBCLO(K)                    S3GE2830
  420 FORMAT(1H ,I5,4X,4F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)S3GE2840
      WRITE(6, 430 )CLGA(K),CLMUA(K),CLA(K),CMGA(K),CMMUA(K),CMTA(K),   S3GE2850
     1              CMA(K)                                              S3GE2860
  430 FORMAT(1H ,19X,3F10.6,10H   *  *   ,4F10.6,10H   *  *   )         S3GE2870
      WRITE(6, 440 )CLGA2(K),CLGA2(K),CMGA2(K),CMGA2(K)                 S3GE2880
  440 FORMAT(1H ,19X,F10.6,10X,F10.6,10H   *  *   ,F10.6,20X,F10.6,     S3GE2890
     1          10H   *  *   )                                          S3GE2900
  450 CONTINUE                                                          S3GE2910
      WRITE(6, 460 )CCLG(1),CCLJ(1),CCL(1),CCMG(1),CCMJ(1),CCMT(1),CCM(1S3GE2920
     1),            CXCP(1),CXCL(1)                                     S3GE2930
  460 FORMAT(1H ,9X,4(10H ---------),10H   *  *   ,4(10H ---------),    S3GE2940
     1          10H   *  *   ,2(10H ---------),/,13X,5HTOTAL,2X,3F10.6, S3GE2950
     2          10H   *  *   ,4F10.6,10H  (APEX)  ,2F10.6)              S3GE2960
      WRITE(6, 470 )CCLG(2),CCLJ(2),CCL(2),CCMG(2),CCMJ(2),CCMT(2),CCM(2S3GE2970
     1))                                                                S3GE2980
  470 FORMAT(1H ,19X,3F10.6,10H   *  *   ,4F10.6,10H  (APEX)  )         S3GE2990
      WRITE(6, 480 )CCLG(3),CCLG(3),CCMG(3),CCMG(3)                     S3GE3000
  480 FORMAT(1H ,19X,F10.6,10X,F10.6,10H   *  *   ,F10.6,20X,F10.6,     S3GE3010
     1          10H  (APEX)  )                                          S3GE3020
      WRITE(6, 490 )CMGMC(1),CMJMC(1),CMTMC(1),CMMC(1),CXCPB(1),CXCLB(1)S3GE3030
  490 FORMAT(1H ,49X,10H   *  *   ,4F10.6,10H  (XMC)   ,2F10.6)         S3GE3040
      WRITE(6, 490 )CMGMC(2),CMJMC(2),CMTMC(2),CMMC(2)                  S3GE3050
      WRITE(6, 500 )CMGMC(3),CMGMC(3)                                   S3GE3060
  500 FORMAT(1H ,49X,10H   *  *   ,F10.6,20X,F10.6,10H  (XMC)   )       S3GE3070
      WRITE(6, 510 )                                                    S3GE3080
  510 FORMAT(1H1,25X,9(4H....),15H INDUCED DRAG .,9(4H....),/,          S3GE3090
     1        7X,7HSECTION,6X,1HY,7X,4HCDGO,7X,5HCDMUO,6X,3HCSO,8X,3HCDO, S3GE3100
     2        8X,6HGAMMAO,5X,6HALFINO,5X,3HCTO,8X,3HCMU,/,              S3GE3110
     3        28X,4HCDGA,7X,5HCDMUA,6X,3HCSA,8X,3HCDA,/,                S3GE3120
     4        28X,5HCDGA2,6X,6HCDMUA2,5X,4HCSA2,7X,4HCDA2,7X,6HGAMMA2,  S3GE3130
     5        5X,6HALFINA/)                                             S3GE3140
      DO 560 K = 1,NROWS                                                S3GE3150
      WRITE(6, 520 )K,Y(K),CDGO(K),CDMUO(K),CSO(K),CDIO(K),CGAMO(K),    S3GE3160
     1              ALFINO(K),CTO(K),CMU(K)                             S3GE3170
  520 FORMAT(1H ,6X,I4,4X,F10.6,8F11.7)                                 S3GE3180
      WRITE(6, 530 )CDGA(K),CDMUA(K),CSA(K),CDIA(K)                     S3GE3190
  530 FORMAT(1H ,24X,4F11.7)                                            S3GE3200
      WRITE(6, 540 )CDGA2(K),CDMUA2(K),CSA2(K),CDIA2(K),CGAM(K),ALFINF(KS3GE3210
     1))                                                                S3GE3220
  540 FORMAT(1H ,24X,6F11.7)                                            S3GE3230
  560 CONTINUE                                                          S3GE3260
      WRITE(6, 570 )CCDG(1),CCDJ(1),CCS(1),CCDI(1),CDITZ(1),CT,CMU(1)   S3GE3270
  570 FORMAT(1H ,24X,8(11H ----------)/18X,5HTOTAL,2X,4F11.7,11X,3F11.7)S3GE3280
      WRITE(6, 580 )CCDG(2),CCDJ(2),CCS(2),CCDI(2),CDITZ(2)             S3GE3290
  580 FORMAT(1H ,24X,4F11.7,11X,F11.7)                                  S3GE3300
      WRITE(6, 580 )CCDG(3),CCDJ(3),CCS(3),CCDI(3),CDITZ(3)             S3GE3310
C  PRINT A TABLE OF ALL TOTAL COEFFICIENTS                              S3GE3340
      CALL HEADER(ILINES)                                               S3GE3350
      WRITE(6, 600 ) M                                                  S3GE3360
  600 FORMAT(1H0,22X,3H***,16(5H*****)/                                 S3GE3370
     1        23X,42H*  WING TOTAL AERODYNAMIC COEFFICIENTS FOR,        S3GE3380
     2        15H COMPOSITE CASE,I3,23H IN GROUND PROXIMITY  */         S3GE3390
     3        23X,3H***,16(5H*****))                                    S3GE3400
      WRITE(6, 610 )                                                    S3GE3410
  610 FORMAT(1H0,41X,7HALPHA=0,8X,5HALPHA,10X,8HALPHA**2)               S3GE3420
  620 FORMAT(1H0,25X,A8,4F15.6)                                         S3GE3430
  630 FORMAT(1H0,25X,A8,1X,4F15.7)                                      S3GE3440
      WRITE(6, 620 )COEFF(1),CCLG(1),CCLG(2),CCLG(3)                    S3GE3450
      WRITE(6, 620 )COEFF(2),CCLJ(1),CCLJ(2)                            S3GE3460
      WRITE(6, 620 )COEFF(3),CCL(1),CCL(2),CCLG(3)                      S3GE3470
      WRITE(6, 630 )COEFF(4),CCDG(1),CCDG(2),CCDG(3)                    S3GE3480
      WRITE(6, 630 )COEFF(5),CCDJ(1),CCDJ(2),CCDJ(3)                    S3GE3490
      WRITE(6, 630 )COEFF(6),CCS(1),CCS(2),CCS(3)                       S3GE3500
      WRITE(6, 630 )COEFF(7),CCDI(1),CCDI(2),CCDI(3)                    S3GE3510
      WRITE(6, 630 )COEFF(8),CDITZ(1),CDITZ(2),CDITZ(3)                 S3GE3520
      WRITE(6, 630 )COEFF(9),CCJ(1)                                     S3GE3530
      WRITE(6, 620 )COEFF(10),CCMG(1),CCMG(2),CCMG(3)                   S3GE3540
      WRITE(6, 620 )COEFF(11),CCMJ(1),CCMJ(2)                           S3GE3550
      WRITE(6, 620 )COEFF(12),CCMT(1),CCMT(2)                           S3GE3560
      WRITE(6, 620 )COEFF(13),CCM(1),CCM(2),CCMG(3)                     S3GE3570
      WRITE(6, 620 )COEFF(14),CXCP(1)                                   S3GE3580
      WRITE(6, 620 )COEFF(15),CXCL(1)                                   S3GE3590
      WRITE(6, 620 )COEFF(16),CXCPB(1)                                  S3GE3600
      WRITE(6, 620 )COEFF(17),CXCLB(1)                                  S3GE3610
      WRITE(6, 620 )COEFF(18),CMGMC(1),CMGMC(2),CMGMC(3)                S3GE3620
      WRITE(6, 620 )COEFF(19),CMJMC(1),CMJMC(2)                         S3GE3630
      WRITE(6, 620 )COEFF(20),CMTMC(1),CMTMC(2)                         S3GE3640
      WRITE(6, 620 )COEFF(21),CMMC(1),CMMC(2),CMGMC(3)                  S3GE3650
      WRITE(6, 620 )COEFF(22),CLLG(1),CLLG(2),CLLG(3)                   S3GE3660
      WRITE(6, 620 )COEFF(23),CLLJ(1),CLLJ(2)                           S3GE3670
      WRITE(6, 620 )COEFF(24),CLL(1),CLL(2),CLLG(3)                     S3GE3680
      WRITE(6, 620 )COEFF(25),CNJ(1),CNJ(2),CNJ(3)                      S3GE3690
      WRITE(6, 620 )COEFF(26),CNG(1),CNG(2),CNG(3)                      S3GE3700
      WRITE(6, 620 )COEFF(27),CCY(1),CCY(2),CCY(3)                      S3GE3710
C  COMPUTE AND PRINT A TABLE OF THE VARIATION OF THE TOTAL COEFFICIENTS S3GE3720
C  WITH ANGLE OF ATTACK IN GROUND EFFECT                                S3GE3730
      CALL TABLGE(CCL(1),CCL(2),CCLG(3),CMMC(1),CMMC(2),CMGMC(3),CLL(1),S3GE3740
     1 CLL(2),CLLG(3),CLLJ(1),CLLJ(2),CDITZ(1),CDITZ(2),CDITZ(3),CCJ(1),S3GE3750
     2 CNG(1),CNG(2),CNG(3),CNJ(1),CNJ(2),CNJ(3),CCY(1),CCY(2),CCY(3),M)S3GE3760
C  SAVE THE TOTAL STATIC WING COEFFICIENTS ON UNIT 9                    S3GE3770
      IF((NROWH+NROWV+NSEG) .EQ. 0) GO TO 650                           S3GE3780
      IWRITE = 2*NMAX + NSEG + M                                        S3GE3790
      PUT(1) = CCL(1)                                                   S3GE3800
      PUT(2) = CCL(2)                                                   S3GE3810
      PUT(3) = CCLG(3)                                                  S3GE3820
      PUT(6) = CMMC(1)                                                  S3GE3830
      PUT(7) = CMMC(2)                                                  S3GE3840
      PUT(8) = CMGMC(3)                                                 S3GE3850
      PUT(11) = CLL(1)                                                  S3GE3860
      PUT(12) = CLL(2)                                                  S3GE3870
      PUT(13) = CLLG(3)                                                 S3GE3880
      DO 640 I = 1,3                                                    S3GE3890
      PUT(15+I) = CDITZ(I)                                              S3GE3900
      PUT(20+I) = CNG(I) + CNJ(I)                                       S3GE3910
      PUT(25+I) = CCY(I)                                                S3GE3920
  640 CONTINUE                                                          S3GE3930
      CALL WRITMS(9,PUT,102,IWRITE)                                     S3GE3960
  650 CONTINUE                                                          S3GE3970
      END                                                               S3GE3980
```

```
      SUBROUTINE TABLGE(CLO,CLA,CLA2,CMO,CMA,CMA2,CLLO,CLLA,CLLA2,CLLJO,
     1  CLLJA,CDIO,CDIX,CDIA2,CJ,CNGO,CNGX,CNGA2,CNJO,CNJX,CNJA2,CYO,
     2  CYX,CYA2,MCASE)
C
C  THIS SUBROUTINE COMPUTES AND PRINTS A TABLE OF TOTAL COEFFICIENTS
C  FOR A RANGE OF ANGLES OF ATTACK FOR A WING IN GROUND EFFECT
C
C  COMPUTE THE COEFFICIENTS
      CALL HEADER(ILINES)
      WRITE(6, 20 ) MCASE
   20 FORMAT(1H0,26X,78(1H*)/27X,35H*   TABULATED WING TOTAL COEFFICIENT,
     120HS FOR COMPOSITE CASE,I3,20H IN GROUND EFFECT   */27X,78(1H*)//
     2  1X,5HALPHA,7X,3HCLL,8X,6HCLL**2,5X,7HCCM(MC),4X,4HCLLJ,7X,3HCLL,
     3  12X,5HCDITZ,6X,2HCT,13X,3HCNJ,8X,2HCN,9X,3HCCY)
      ALPHA = -11.00
      DO 40  M = 1,41
      ALPHA = ALPHA + 1.00
      ALPHA2 = ALPHA * ALPHA
      ALPHA3 = ALPHA2 * ALPHA
C  LINEAR COEFFICIENTS
      CLLJ = CLLJO + CLLJA*ALPHA
C  NONLINEAR COEFFICIENTS
      CL = CLO + CLA * ALPHA + CLA2 * ALPHA2
      CL2 = CL * CL
      CM = CMO + CMA * ALPHA + CMA2 * ALPHA2
      CLL = CLLO + CLLA * ALPHA + CLLA2 * ALPHA2
      CDI = CDIO + CDIX * ALPHA + CDIA2 * ALPHA2
      CT = CJ - CDI
      CNG = CNGO + CNGX*ALPHA + CNGA2*ALPHA2
      CNJ = CNJO + CNJX*ALPHA + CNJA2*ALPHA2
      CN = CNJ + CNG
      CY = CYO + CYX*ALPHA + CYA2*ALPHA2
C  PRINT THE TABLE
      WRITE(6, 30 ) ALPHA,CL,CL2,CM,CLLJ,CLL,CDI,CT,CNJ,CN,CY
   30 FORMAT(1H ,F5.1,4X,5F11.7,4X,2F11.7,4X,3F11.7)
   40 CONTINUE
      RETURN
      END
```

```
*DECK STG3VT
      OVERLAY(OVLY,3,3)
      PROGRAM STG3VT
C
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE, SPANWISE AND
C  TOTAL LOADING FOR ONE COMPOSITE CASE AT FIVE ANGLES OF ATTACK
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),
     1              D(40),KK(400),ITYPE(400)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),
     1              XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
```

```
      COMMON/SOLV1/GAMMA(400,10)
      COMMON/SOLV3/ONSET(400)
      COMMON/COMPOS/FACTOR(10,24),NCC
      COMMON/GEFECT/GHITE(10),IGRND,IGNC,NHITE
      COMMON/LOAD1/ TWIST(40),HO(40),TH(40),THETS(40),
     1            BTA(400),EP(400),CPO(400)
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),
     1            CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)
      COMMON/LOAD4/ CLGO(40),CLMUO(40),CLO(40),CDMUO(40),CDGO(40),
     1      CDIO(40),CSO(40), CTI(40),CMGO(40),CMMUO(40),CMTO(40),CMO(40),
     2            XBCPO(40),XBCLO(40),FACT(10)
      COMMON/LOAD5/ CGAM(40),CGAMO(40),ALFINF(40),ALFINO(40),DUMB(40)
      COMMON/LOAD6/ COEFF(27)
      COMMON /FFLD5/ GAMJ(400,5)
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HI,HH,VL,VH,
     1               NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2               ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,KBB (41),ZCL(41),
     1                RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2                V(40,7),W(40,7)
      COMMON/JET13/CJ(10),THO(10)
      COMMON/KEEPEM/CLW(5),CMW(5),CLLW(5),COW(5),CNW(5),CYW(5),CLOW,
     1        CMOW,CLLPW,CLLRW,CNPW,CNRW,CYPW,CYRW,CLH(5),CMH(5),CLLH(5),
     2        COH(5),CLOH,CMOH,CLLPH,CYV(5),CYBV,CNV(5),CNBV,CLLV(5),
     3        CLLBV,COV(5),CLLPV,CLLRV,CNPV,CNRV,CYPV,CYRV,CMF(5),CNF(5),
     4        CMOF,CNRF
      DIMENSION CP(400),GAMSUM(400)
      EQUIVALENCE (GAMSUM(1),ONSET(1))
      LOGICAL PRINT,GE
      PRINT = (IPRINT .GT. 0)
      GE = (IGND .NE. 0)
      M = KCASE
      ALPHA = -5.0
C  CYCLE ANGLE OF ATTACK OVER THE RANGE OF 0.0 TO 20.0 DEGREES
      DO 470 NNN = 1,5
      ALPHA = ALPHA + 5.0
      LOGIC = 0
      I = 0
      NCI = NCASES + 1
      CALL HEADER(ILINES)
      ILINES = ILINES + 9
      IF(GE)WRITE(6, 20 )M,ALPHA
   20 FORMAT(1H ,36X,9(5H*****)/
     1              37X,39H*  CHORDWISE LOADING FOR COMPOSITE CASE,I3,3H  */
     2              37X,36H*     IN GROUND EFFECT  AT  ALPHA = ,F6.1,3H  */
     3              37X,1H*,11X,20HWITH VECTORED THRUST, 12X,1H*/
     4              37X,9(5H*****))
      IF(.NOT. GE) WRITE(6, 30 )M,ALPHA
   30 FORMAT(1H ,36X,9(5H*****)/
     1              37X,39H*  CHORDWISE LOADING FOR COMPOSITE CASE,I3,3H  */
     2              37X,36H*  WITH VECTORED THRUST AT ALPHA =  ,F6.1,3H  */
     3              37X,9(5H*****))
      FACT(1) = ALPHA + FACTOR(1,M)
      DO 40 N = 2,NCASES
   40 FACT(N) = FACTOR(N,M)
      IF(NCI .GT. 10) GO TO 60
      DO 50  N = NCI,10
      FACT(N) = 0.00
   50 CONTINUE
   60 WRITE(6, 70 )(N,N=1,10),(FACT(N),N=1,10)
```

```
   70 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),
     1       3X),5X,2HA(,I2,1H)/10X,10F12.6)
      WRITE(6, 80 )
   80 FORMAT(1H0,7X,47H*** NOTE ***  EACH LEADING EDGE CP VALUE IS THE,
     1          43H AVERAGE VALUE OF THE SINGULAR DISTRIBUTION)
      IF(IHINGE .NE. 0) WRITE(6, 90 )
   90 FORMAT(1H ,21X,47HIF A HINGE IS DEFLECTED THE LOADING IS SINGULAR,
     1         53H AND THE CP VALUE IS FOR THE REGULAR EVC PORTION ONLY)
      WRITE(6, 100 )
  100 FORMAT(1H ,21X,41HDO NOT PLOT THESE LOADING POINTS DIRECTLY)
C  CALCULATE AND PRINT THE CHORDWISE LOADING
      DO 220 K = 1,NROWS
      DUMB(K) = 0.0
C  ON THE WING
      NWK = NW(K)
      DO 160  L = 1,NWK
      I = I + 1
      IREAD = 2 * I - 1
      BTA(I) = 0.00
      EP(I) = 0.00
      CPO(I) = 0.0
      DO 110 N = 1,NCASES
      BTA(I) = BTA(I) + BETA(I,N) * FACT(N)
      EP(I) = EP(I) + EPS(I,N) * FACT(N)
      CPO(I) = CPO(I) + GAMMA(I,N) * FACT(N)
  110 CONTINUE
      IF(.NOT. GE) GO TO 140
      U = 1.0
      CALL READMS(9,ONSET,NEWMAX,IREAD)
      DO 130 N = 1,NCASES
      DO 120 J = 1,NEWMAX
      U = U + GAMMA(J,N) * FACT(N) * ONSET(J)
  120 CONTINUE
  130 CONTINUE
      CP(I) = 2.0 * U * CPO(I)
      GO TO 150
  140 CP(I) = 2.00*CPO(I)
  150 CP(I) = CP(I) + 2.00*GAMJ(I,NNN)
  160 CONTINUE
      IF(.NOT. GE) GO TO 500
      DO 480 I = 1,NEWMAX
      GAMSUM(I) = GAMJ(I,NNN)
      DO 490 N = 1,NCASES
  490 GAMSUM(I) = GAMSUM(I) + GAMMA(I,N) * FACT(N)
  480 CONTINUE
  500 IF(PRINT) GO TO 220
      J1 = IW(K)
      J2 = IW(K) + NWK - 1
      NEXT = 3 + 2 * (NWK/10+1)
      ILINES = ILINES + NEXT
      IF((ILINES .LT. 56) .OR. (K .EQ. 1)) GO TO 180
      WRITE(6, 170 )
  170 FORMAT(1H1)
      ILINES = 1
  180 WRITE(6, 190 ) K,Y(K),CHORD(K)
  190 FORMAT(1H0,35X,7HSECTION,I3,5X,3HY =,F10.6,5X,7HCHORD =,F10.6/
     1         2X,4HWING)
      WRITE(6, 200 ) (XB(J),J=J1,J2)
  200 FORMAT(1H ,7X,2HXB,10F12.6,3(/10X,10F12.6))
      WRITE(6, 210 )(CP(J),J=J1,J2)
```

```
  210 FORMAT(1H ,7X,2HCP,10F12.6,3(/10X,10F12.6))
  220 CONTINUE
C  MODULATE AND SUM THE FUNDAMENTAL CASE SECTIONAL PARAMETERS
  230 DO 260 K = 1,NROWS
      TWIST(K) = 0.00
      TH(K) = 0.00
      THETS(K) = 0.00
      HO(K) = 0.00
      DO 250 N = 1,NCASES
  240 TWIST(K) = TWIST(K) + TST(K,N) * FACT(N)
      TH(K) = TH(K) + THETA(K,N) * FACT(N)
      THETS(K) = THETS(K) + THS(K,N) * FACT(N)
      HO(K) = HO(K) + HL(K,N) * FACT(N)
  250 CONTINUE
  260 CONTINUE
C  COMPUTE SECTIONAL COEFFICIENTS
      IF(GE) GO TO 510
      CALL SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,
     1       XB,DEL,BTA,EP,CP,CL,CLG,CLMU,CM,CMG,CMMU,CMT,
     2       XBCP,XBCL,CDI,CDMU,CDG,CS,CT,NROWS,IHINGE)
C  COMPUTE SECTIONAL VORTICITY
      CALL SLOADG(CP,DEL,BTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)
      GO TO 270
  510 CALL LOADG1(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,
     1       EP,CP,CLG,CL,CLMU,CMG,CM,CMMU,CMT,XBCP,XBCL,NROWS)
      IDENT = 1
      CALL LOADG2(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,XB,DEL,BTA,
     1       EP,GAMSUM,GAMSUM,CDG,CDMU,CDI,CS,CT,NROWS,IDENT)
      CALL GLOADG(GAMSUM,DEL,BTA,CHORD,D,CMU,NW,IW,NJ,IJ,CGAM,NROWS)
  270 CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)
C  COMPUTE TOTAL COEFFICIENTS ASSOCIATED WITH LIFT
      CALL TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,HO,XB,XLEAD,BTA,
     1       CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG,CCLJ,CCL,CCMG,CCMJ,CCMT,CCM,
     2       CMGMC,CMJMC,CMTMC,CMMC,CXCP,CXCL,CXCPB,CXCLB,CCJ,CLLG,CLLJ,
     3       CLL,ITYPE,IW,NW,NROWS,ISYMM)
C  COMPUTE TOTAL COEFFICIENTS ASSOCIATED WITH DRAG
      CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,DUMB,
     1       CCDG,CCDJ,CCS,CCDI,CDITZ,DUMB,ALFINF,CCT,CCJ,
     2       CNJ,CNJJ,CNG,CCY,XLEAD,TANLE,XMC,NROWS,ISYMM,LOGIC)
C  ADD CONTRIBUTIONS DUE TO VECTORED JET REACTIONS
      CALL TLODVT(ALPHA,CREF,XMC,CCLJ,CCL,CCMG,CCMJ,CCMT,CCM,
     1       CMJMC,CMTMC,CMMC,CXCL,CXCLB,CLLJ,CLL,CCJ,
     2     CCDJ,CCDI,CCT,CNJ,CCY)
      WRITE(6, 280 )M
  280 FORMAT(1H1,37X,11(4H****) / 38X,23H*  SPANWISE LOADING FOR,
     1         15H COMPOSITE CASE, I3, 3H  *)
      IF(GE) WRITE(6, 290 ) ALPHA   *
  290 FORMAT(1H ,37X,34H*  IN GROUND  EFFECT  AT  ALPHA = ,F6.1, 4H    *)
      IF(.NOT. GE)WRITE(6, 300 ) ALPHA
  300 FORMAT(1H ,37X,35H*  WITH VECTORED THRUST AT ALPHA = ,F6.1,3H   *)
      IF(GE) WRITE(6, 310 )
  310 FORMAT(1H ,37X,1H*, 11X,20HWITH VECTORED THRUST, 11X,1H*)
      WRITE(6, 320 )
  320 FORMAT(1H ,37X,11(4H****))
      WRITE(6, 330 ) (N,N=1,10),(FACT(N),N=1,10)
  330 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(4X,2HA(,I1,1H),
     1       2X),3X,2HA(,I2,1H)/10X,10F10.6/)
      WRITE(6, 340 )
  340 FORMAT(1H ,20X,29H............ LIFT .............,11H    *  *
     1         39H............ PITCHING MOMENT ...........,10H    *  *
```

```
2        20H ... LIFT CENTER .../ 2X,7HSECTION,5X,1HY,9X,3HCLG,6X,      S3VT1840
3        4HCLMU,6X,2HCL,5X,10H   *  *   ,4X,3HCMG,6X,4HCMMU,6X,         S3VT1850
4        3HCMT,7X,2HCM,5X,10H   *  *   ,4X,5HXCP/C,4X,5HXCL/C)         S3VT1860
      DO 360 K = 1,NROWS                                                S3VT1870
      WRITE(6, 350 )K,Y(K),CLG(K),CLMU(K),CL(K),CMG(K),CMMU(K),CMT(K),  S3VT1880
     1              CM(K),XBCP(K),XBCL(K)                               S3VT1890
  350 FORMAT(1H ,I5,4X,4F10.6,10H   *  *   ,4F10.6,10H   *  *   ,2F10.6)S3VT1900
  360 CONTINUE                                                          S3VT1910
      WRITE(6, 370 )CCLG,CCLJ,CCL,CCMG,CCMJ,CCMT,CCM,CXCP,CXCL          S3VT1920
  370 FORMAT(1H , 9X,4(10H ----------),10H   *  *   ,4(10H ----------), S3VT1930
     1        10H   *  *   ,2(10H ----------)/13X,5HTOTAL,2X,3F10.6,     S3VT1940
     2        10H   *  *   ,4F10.6,10H  (APEX)  , 2F10.6)               S3VT1950
      WRITE(6, 380 )CMGMC,CMJMC,CMTMC,CMMC,CXCPB,CXCLB                  S3VT1960
  380 FORMAT(1H ,49X,                                                   S3VT1970
     1        10H   *  *   ,4F10.6,10H  (XMC)   , 2F10.6)               S3VT1980
      WRITE(6, 390 )                                                    S3VT1990
  390 FORMAT(1H1,25X,9(4H....),14H INDUCED DRAG , 5(3H...) /            S3VT2000
     1    7X,7HSECTION,6X,1HY,8X,3HCDG,8X,4HCDMU,7X,2HCS,9X,2HCD,       S3VT2010
     2     9X,5HGAMMA,6X,5HALFIN,6X,2HCT,8X,3HCMU)                      S3VT2020
      DO 410 K = 1,NROWS                                                S3VT2030
      WRITE(6, 400 )K,Y(K),CDG(K),CDMU(K),CS(K),CDI(K),CGAM(K),ALFINF(K)S3VT2040
  400 FORMAT(1H , 6X,I4 ,4X,F10.6,8F11.7)                               S3VT2050
  410 CONTINUE                                                          S3VT2060
      WRITE(6,420)CCDG,CCDJ,CCS,CCDI,CDITZ,CCT,CCJ                      S3VT2070
  420 FORMAT(1H ,24X,8(11H ----------)/18X,5HTOTAL,2X,4F11.7,11X,3F11.7)S3VT2080
C  PRINT A TABLE OF ALL TOTAL COEFFICIENTS                              S3VT2090
      CALL HEADER(ILINES)                                               S3VT2100
      WRITE(6, 430 ) M                                                  S3VT2110
  430 FORMAT(1H0,28X,15(4H****),3H*** /                                 S3VT2120
     1 29X,57H*  WING TOTAL AERODYNAMIC COEFFICIENTS FOR COMPOSITE CASE,S3VT2130
     2 I3,3H  *)                                                        S3VT2140
      IF(GE) WRITE(6, 440 ) ALPHA                                       S3VT2150
  440 FORMAT(1H ,28X,42H*    IN GROUND EFFECT WITH VECTORED THRUST,     S3VT2160
     1 11H AT ALPHA =, F5.1, 5H    * / 29X,15(4H****),3H***)            S3VT2170
      IF(.NOT. GE) WRITE(6, 450 ) ALPHA                                 S3VT2180
  450 FORMAT(1H ,28X,1H*,12X,31HWITH VECTORED THRUST AT ALPHA =,        S3VT2190
     1 F5.1,13X,1H* / 29X,15(4H****),3H***)                             S3VT2200
  460 FORMAT(1H0,52X,A8,F15.7)                                          S3VT2210
      WRITE(6, 460 )COEFF(1),CCLG,COEFF(2),CCLJ,COEFF(3),CCL,           S3VT2220
     1              COEFF(4),CCDG,COEFF(5),CCDJ,COEFF(6),CCS,           S3VT2230
     2              COEFF(7),CCDI,COEFF(8),CDITZ,COEFF(9),CCJ,          S3VT2240
     3              COEFF(10),CCMG,COEFF(11),CCMJ,COEFF(12),            S3VT2250
     4              CCMT,COEFF(13),CCM,COEFF(14),CXCP,COEFF(15),        S3VT2260
     5              CXCL,COEFF(16),CXCPB,COEFF(17),CXCLB,COEFF(18),     S3VT2270
     6              CMGMC,COEFF(19),CMJMC,COEFF(20),CMTMC,COEFF(21),    S3VT2280
     7              CMMC,COEFF(22),CLLG,COEFF(23),CLLJ,COEFF(24),       S3VT2290
     8              CLL,COEFF(25),CNJ,COEFF(26),CNG,COEFF(27),CCY       S3VT2300
C  STORE THE TOTAL STATIC WING COEFFICIENTS IN THE PROPER SAVE ARRAYS   S3VT2310
      CLW(NNN) = CCL                                                    S3VT2320
      CMW(NNN) = CMMC                                                   S3VT2330
      CLLW(NNN) = CLL                                                   S3VT2340
      CDW(NNN) = CDITZ                                                  S3VT2350
      CNW(NNN) = CNG + CNJ                                              S3VT2360
      CYW(NNN) = CCY                                                    S3VT2370
  470 CONTINUE                                                          S3VT2380
      CONTINUE                                                          S3VT2390
      END                                                               S3VT2400


*DECK TLODVT                                                            TLVT0010


      SUBROUTINE TLODVT(ALPHA,CREF,XMC,                                 TLVT0020
     1          CCLJ,CLL,CCMG,CCMJ,CCMT,CCM,                            TLVT0030
     2           CMJMC,CMTMC,CMMC,CXCL,CXCLB,CLLJ,CLL,CLJ,              TLVT0040
     3    CCDJ,CCDI,CCT,CNJ,CCY)                                        TLVT0050
C                                                                       TLVT0060
C  THIS SUBROUTINE COMPUTES THE REACTION FORCE AND MOMENT CONTRIBUTIONS TLVT0070
C  DUE TO ALL VECTORED JETS                                             TLVT0080
C                                                                       TLVT0090
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                TLVT0100
      COMMON/JET12/UJ(10),DDD(10),THTA(10),HH(10),ZMX(10),X0(10),Y0(10),TLVT0110
     1            Z0(10),DIHE(10),ISY(10),JETRE(10),NJETS               TLVT0120
      COMMON/JET13/CJ(10),THO(10)                                       TLVT0130
C                                                                       TLVT0140
C  SUM UP THE REACTIONS FOR ALL JETS                                    TLVT0150
      B = 2.0                                                           TLVT0151
      CCJ  = 0.0                                                        TLVT0160
      CCLJ = 0.0                                                        TLVT0170
      CCDJ = 0.0                                                        TLVT0180
      CCMJ = 0.0                                                        TLVT0190
      CCMT = 0.0                                                        TLVT0200
      CMJMC= 0.0                                                        TLVT0210
      CMTMC= 0.0                                                        TLVT0220
      CLLJ = 0.0                                                        TLVT0230
      CNJ  = 0.0                                                        TLVT0240
      CCYJ = 0.0                                                        TLVT0250
      A = ALPHA / 57.295779                                             TLVT0260
      SA = SIN(A)                                                       TLVT0270
      CA = COS(A)                                                       TLVT0280
      X1 = XMC * CA                                                     TLVT0290
      Z1 = XMC * SA                                                     TLVT0300
      DO 30  N = 1,NJETS                                                TLVT0310
      AT = (ALPHA + THO(N)) / 57.295779                                 TLVT0320
      D = DIHE(N) / 57.295779                                           TLVT0330
      STA = SIN(AT)                                                     TLVT0340
      CTA = COS(AT)                                                     TLVT0350
      CLJ = CJ(N) * STA * COS(D)                                        TLVT0360
      CTJ = CJ(N) * CTA                                                 TLVT0370
      CDJ = CJ(N) * (1.0 - CTA)                                         TLVT0371
      CYJ =-CJ(N) * STA * SIN(D)                                        TLVT0380
      CCJ  = CCJ  + CJ(N)                                               TLVT0390
      X2 = (X0(N)-XMC)*CA + Z0(N)*SA                                    TLVT0400
      Z2 = Z0(N)*CA - (X0(N)-XMC)*SA                                    TLVT0410
      IF(ISY(N) .NE. 0) GO TO 20                                        TLVT0420
      CLJ = 2.0 * CLJ                                                   TLVT0430
      CTJ = 2.0 * CTJ                                                   TLVT0440
      CDJ = 2.0 * CDJ                                                   TLVT0441
      CYJ = 0.0                                                         TLVT0450
      CCJ = CCJ + CJ(N)                                                 TLVT0460
   20 CCLJ = CCLJ + CLJ                                                 TLVT0470
      CCDJ = CCDJ + CDJ                                                 TLVT0480
      CCMJ = CCMJ - CLJ * (X1+X2)                                       TLVT0490
      CCMT = CCMT - CTJ * (Z1+Z2)                                       TLVT0500
      CMJMC= CMJMC - CLJ * X2                                           TLVT0510
      CMTMC = CMTMC - CTJ * Z2                                          TLVT0520
      CCDI = CCDI + CDJ                                                 TLVT0521
      IF(ISY(N) .EQ. 0) GO TO 30                                        TLVT0530
      CLLJ = CLLJ - CLJ * Y0(N) + CYJ * Z2                              TLVT0540
      CNJ = CNJ - CTJ * Y0(N) - CYJ * X2                                TLVT0550
      CCYJ = CCYJ + CYJ                                                 TLVT0560
   30 CONTINUE                                                          TLVT0570
C  COMPUTE THE FINAL COEFFICIENTS                                       TLVT0580
      CCL = CCL + CCLJ                                                  TLVT0590
      CCMJ = CCMJ / CREF                                                TLVT0591
      CCMT = CCMT / CREF                                                TLVT0592
      CCM = CCM + CCMJ + CCMT                                           TLVT0600
      CMJMC = CMJMC / CREF                                              TLVT0610
      CMTMC = CMTMC / CREF                                              TLVT0611
      CMMC = CMMC + CMJMC + CMTMC                                       TLVT0620
      IF(CCL .NE. 0.0) CXCL = -(CCMG+CCMJ) / CCL                        TLVT0630
      CXCLB = CXCL * CREF                                               TLVT0640
      CLLJ = CLLJ / B                                                   TLVT0640
      CLL = CLL + CLLJ                                                  TLVT0650
      CCT = CCJ - CCDI                                                  TLVT0660
      CNJ = CNJ / B                                                     TLVT0661
      CCY = CCY + CCYJ                                                  TLVT0670
      RETURN                                                            TLVT0680
      END                                                               TLVT0690


*DECK STAGE4                                                            STG40010
      OVERLAY(OVLY,4,0)                                                 STG40020
      PROGRAM STAGE4                                                    STG40030
C                                                                       STG40040
C  THIS PROGRAM CONTROLS THE EXECUTION OF UTILITY ROUTINES AND          STG40050
C  BOUNDARY CONDITION SETUP FOR STABILITY DERIVATIVE RUNS               STG40060
C                                                                       STG40070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            STG40080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) STG40090
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     STG40100
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                STG40110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  STG40120
     1            D(40),KK(400),ITYPE(400)                              STG40130
      COMMON/FCASE2/TST(40,10),HL(40,10),OJ(40),ACTE(40),AC(20,40),     STG40140
     1            XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NWTSTG40150
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    STG40160
      COMMON/SOLV1/CP(400,10)                                           STG40170
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          STG40180
      DIMENSION CPREAD(400)                                             STG40190
      LOGICAL PASS                                                      STG40200
C                                                                       STG40210
C  IF THIS IS A NEW RUN RETRIEVE THE OLD GEOMETRY PARAMETERS            STG40220
      PASS = (NEWCMU .GT. 1) .OR. (NHITE .GT. 1)                        STG40230
      ISIZE = NMAX                                                      STG40240
      IF(LOGIC .GT. 1) GO TO 30                                         STG40250
      IF(PASS) CALL RETREV(EPS,BETA,TST,THETA,THS,CPREAD,               STG40260
     1      NMAX,NJT,ISIZE,NCASES)                                      STG40270
C  DEFINE THE FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING          STG40280
   20 CALL BCPICH(XCG,CREF,XI,DEL,EPS,BETA,CHORD,KK,THETA,THS,TST,HL,   STG40290
     1      NW,IW,NJ,IJ,NWT,NMAX,NROWS,NCASES)                          STG40300
      IF(PASS) GO TO 60                                                 STG40310
      CALL SAVEIT(EPS,BETA,TST,THETA,THS,CPREAD,NMAX,NJT,ISIZE,NCASES)  STG40320
      GO TO 60                                                          STG40330
   30 ISIZE = NEWMAX                                                    STG40340
C  DEFINE THE FUNDAMENTAL CASES FOR YAWING DERIVATIVES                  STG40350
      NC1 = NCASES - 1                                                  STG40360
   40 CALL BCYAW(EPS,BETA,THETA,THS,Y,KK,NWT,NMAX,NROWS,NC1)            STG40370
C  DEFINE THE LAST FUNDAMENTAL CASE FOR ROLLING DERIVATIVES             STG40380
   50 CALL BCROLL(EPS,BETA,THETA,THS,TST,Y,NW,NWT,NMAX,NROWS,NCASES)    STG40390


   60 CONTINUE                                                          STG40400
      END                                                               STG40410


*DECK SAVEIT                                                            SAVI0010
      SUBROUTINE SAVEIT(EPS,BETA,TST,THETA,THS,DUMMY,                   SAVI0020
     1      NMAX,NJT,ISIZE,NCASES)                                      SAVI0030
      DIMENSION EPS(400,10),BETA(400,10),DUMMY(ISIZE)                   SAVI0040
      DIMENSION TST(40,10),THETA(40,10),THS(40,10)                      SAVI0050
C                                                                       SAVI0060
   20 IWRITE = 2 * (NMAX+NJT) + NCASES                                  SAVI0070
C  STORE EPS                                                            SAVI0080
      DO 40 N = 1,NCASES                                                SAVI0090
      IWRITE = IWRITE + 1                                               SAVI0100
      DO 30 I = 1,ISIZE                                                 SAVI0110
      DUMMY(I) = EPS(I,N)                                               SAVI0120
   30 CONTINUE                                                          SAVI0130
      CALL WRITMS(1,DUMMY,ISIZE,IWRITE)                                 SAVI0140
   40 CONTINUE                                                          SAVI0150
C  STORE BETA                                                           SAVI0160
      DO 60 N = 1,NCASES                                                SAVI0170
      IWRITE = IWRITE + 1                                               SAVI0180
      DO 50 I = 1,ISIZE                                                 SAVI0190
      DUMMY(I) = BETA(I,N)                                              SAVI0200
   50 CONTINUE                                                          SAVI0210
      CALL WRITMS(1,DUMMY,ISIZE,IWRITE)                                 SAVI0220
   60 CONTINUE                                                          SAVI0230
C  STORE TST                                                            SAVI0240
      IWRITE = IWRITE + 1                                               SAVI0250
      CALL WRITMS(1,TST,400,IWRITE)                                     SAVI0260
C  STORE THETA                                                          SAVI0270
      IWRITE = IWRITE + 1                                               SAVI0280
      CALL WRITMS(1,THETA,400,IWRITE)                                   SAVI0290
C  STORE THS                                                            SAVI0300
      IWRITE = IWRITE + 1                                               SAVI0310
      CALL WRITMS(1,THS,400,IWRITE)                                     SAVI0320
      RETURN                                                            SAVI0330
      END                                                               SAVI0340


*DECK RETREV                                                            RETR0010
      SUBROUTINE RETREV(EPS,BETA,TST,THETA,THS,DUMMY,                   RETR0020
     1      NMAX,NJT,ISIZE,NCASES)                                      RETR0030
      DIMENSION EPS(400,10),BETA(400,10),DUMMY(ISIZE)                   RETR0040
      DIMENSION TST(40,10),THETA(40,10),THS(40,10)                      RETR0050
C                                                                       RETR0060
      IREAD = 2 * (NMAX + NJT) + NCASES                                 RETR0070
C  RETRIEVE EPS                                                         RETR0080
      DO 20 N = 1,NCASES                                                RETR0090
      IREAD = IREAD + 1                                                 RETR0100
      CALL READMS(1,DUMMY,ISIZE,IREAD)                                  RETR0110
      DO 20 I = 1,ISIZE                                                 RETR0120
      EPS(I,N) = DUMMY(I)                                               RETR0130
   20 CONTINUE                                                          RETR0140
C  RETRIEVE BETA                                                        RETR0150
      DO 30 N = 1,NCASES                                                RETR0160
```

```
      IREAD = IREAD + 1                                                 RETR0170
      CALL READMS(1,DUMMY,ISIZE,IREAD)                                  RETR0180
      DO 30 I = 1,ISIZE                                                 RETR0190
      BETA(I,N) = DUMMY(I)                                              RETR0200
   30 CONTINUE                                                          RETR0210
C  RETRIEVE TST                                                         RETR0220
      IREAD = IREAD + 1                                                 RETR0230
      CALL READMS(1,TST,400,IREAD)                                      RETR0240
C  RETRIEVE THETA                                                       RETR0250
      IREAD = IREAD + 1                                                 RETR0260
      CALL READMS(1,THETA,400,IREAD)                                    RETR0270
C  RETRIEVE THS                                                         RETR0280
      IREAD = IREAD + 1                                                 RETR0290
      CALL READMS(1,THS,400,IREAD)                                      RETR0300
      RETURN                                                            RETR0310
      END                                                               RETR0320


*DECK BCPICH                                                            BCPI0010
      SUBROUTINE BCPICH(XLG,CREF,XI,DEL,EPS,BETA,C,KK,THETA,THS,TST,HL, BCPI0020
     1                  NW,IW,NJ,IJ,NWT,NMAX,NROWS,N)                   BCPI0030
C                                                                       BCPI0040
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR                  BCPI0050
C  THE PITCHING RATE DERIVATIVE FUNDAMENTAL CASE                        BCPI0060
C                                                                       BCPI0070
      DIMENSION XI(400),DEL(400),C(40),KK(400)                          BCPI0080
      DIMENSION NW(40),IW(40),NJ(40),IJ(40)                             BCPI0090
      DIMENSION EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)        BCPI0100
      DIMENSION TST(40,10),HL(40,10)                                    BCPI0110
C                                                                       BCPI0120
C  DEFINE THE CAMBER ANGLES WHICH RESULT FROM PITCHING                  BCPI0130
      DO 30 I = 1,NWT                                                   BCPI0140
      KKI = KK(I)                                                       BCPI0150
   20 EPS(I,N)=(2.0*(XI(I)+(DEL(I)/2.0)*C(KKI)-XCG)/CREF)*57.295779     BCPI0160
      BETA(I,N) = 0.00                                                  BCPI0170
   30 CONTINUE                                                          BCPI0180
      IF(NWT .EQ. NMAX) GO TO 60                                        BCPI0190
      NWT1 = NWT + 1                                                    BCPI0200
      DO 50 I = NWT1,NMAX                                               BCPI0210
   40 EPS(I,N) = 0.00                                                   BCPI0220
      BETA(I,N) = 0.00                                                  BCPI0230
   50 CONTINUE                                                          BCPI0240
C                                                                       BCPI0250
C  DEFINE THE JET ANGLES WHICH RESULT FROM PITCHING                     BCPI0260
   60 DO 80 K = 1,NROWS                                                 BCPI0270
   70 THETA(K,N) = 0.00                                                 BCPI0280
      IJK = IJ(K)                                                       BCPI0290
      IF(NJ(K).GT.0) THETA(K,N)=(2.0*(XI(IJK)-XCG)/CREF)*57.295779      BCPI0300
      THS(K,N) = 0.00                                                   BCPI0310
      TST(K,N) = 0.00                                                   BCPI0320
      HL(K,N) = 0.00                                                    BCPI0330
   80 CONTINUE                                                          BCPI0340
      RETURN                                                            BCPI0350
      END                                                               BCPI0360


*DECK BCROLL                                                            BCRO0010


      SUBROUTINE BCROLL(EPS,BETA,THETA,THS,TST,Y,                       BCRO0020
     1                  NW,NWT,NMAX,NROWS,NCASES)                       BCRO0030
C                                                                       BCRO0040
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR                  BCRO0050
C  THE ROLLING RATE DERIVATIVE FUNDAMENTAL CASE                         BCRO0060
C                                                                       BCRO0070
      DIMENSION EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10),       BCRO0080
     1          TST(40,10)                                              BCRO0090
      DIMENSION Y(40),NW(40)                                            BCRO0100
C                                                                       BCRO0110
C  DEFINE THE TWIST AND CAMBER ANGLES WHICH RESULT FROM ROLLING         BCRO0120
      N = NCASES                                                        BCRO0130
      I = 0                                                             BCRO0140
      DO 50 K = 1,NROWS                                                 BCRO0150
   25 TST(K,N) = Y(K) * 57.295779                                       BCRO0160
      THETA(K,N) = TST(K,N)                                             BCRO0170
      THS(K,N) = 0.00                                                   BCRO0180
      NWK = NW(K)                                                       BCRO0190
      DO 40 L = 1,NWK                                                   BCRO0200
      I = I + 1                                                         BCRO0210
   30 EPS(I,N) = TST(K,N)                                               BCRO0220
   40 CONTINUE                                                          BCRO0230
   50 CONTINUE                                                          BCRO0240
C                                                                       BCRO0250
C  DEFINE THE ANGLES ON THE JET                                         BCRO0260
      IF(NMAX .EQ. NWT) RETURN                                          BCRO0270
      NWT1 = NWT + 1                                                    BCRO0280
      DO 70 I = NWT1,NMAX                                               BCRO0290
   60 EPS(I,N) = 0.00                                                   BCRO0300
   70 CONTINUE                                                          BCRO0310
      RETURN                                                            BCRO0320
      END                                                               BCRO0330


*DECK BCYAW                                                             BCYA0010
      SUBROUTINE BCYAW (EPS,BETA,THETA,THS,Y,KK,                        BCYA0020
     1                  NWT,NMAX,NROWS,NCASES)                          BCYA0030
C                                                                       BCYA0040
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR ALL OF THE       BCYA0050
C  YAWING RATE DERIVATIVE FUNDAMENTAL CASES                             BCYA0060
C                                                                       BCYA0070
      COMMON /DERIV/ UO(40),CLQ,CMQ,CMQMC                               BCYA0080
      DIMENSION EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)        BCYA0090
      DIMENSION Y(40),KK(400)                                           BCYA0100
C                                                                       BCYA0110
C  DEFINE THE SECTIONAL NORMALIZED VELOCITY INDUCED BY YAWING           BCYA0120
      DO 20 K = 1,NROWS                                                 BCYA0130
      UO(K) = Y(K)                                                      BCYA0140
   20 CONTINUE                                                          BCYA0150
C                                                                       BCYA0160
C  DEFINE THE ANGLES FOR ALL FUNDAMENTAL CASES                          BCYA0170
      DO 90  N = 1,NCASES                                               BCYA0180
      DO 40 I = 1,NWT                                                   BCYA0190
      KKI = KK(I)                                                       BCYA0200
   30 EPS(I,N) = -UO(KKI) * EPS(I,N)                                    BCYA0210
      BETA(I,N) = 0.00                                                  BCYA0220
   40 CONTINUE                                                          BCYA0230
C                                                                       BCYA0240
      DO 60 K = 1,NROWS                                                 BCYA0250
   50 THETA(K,N) = -UO(K) * THETA(K,N)                                  BCYA0260
      THS(K,N) = 0.00                                                   BCYA0270
   60 CONTINUE                                                          BCYA0280
      IF(NMAX .EQ. NWT) GO TO 90                                        BCYA0290
      NWT1 = NWT + 1                                                    BCYA0300
      DO 80 I = NWT1,NMAX                                               BCYA0310
   70 EPS(I,N) = 0.00                                                   BCYA0320
      BETA(I,N) = 0.00                                                  BCYA0330
   80 CONTINUE                                                          BCYA0340
   90 CONTINUE                                                          BCYA0350
      RETURN                                                            BCYA0360
      END                                                               BCYA0370


*DECK STAGE5                                                            STG50010
      OVERLAY(OVLY,5,0)                                                 STG50020
      PROGRAM STAGE5                                                    STG50030
C                                                                       STG50040
C  THIS SUBROUTINE CONTROLS THE EXECUTION OF THE NON-PLANAR             STG50050
C  JET/WING FLOW FIELD PROGRAM                                          STG50060
C                                                                       STG50070
      COMMON/SOLVI/GAMMA(400,10)                                        STG50080
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                 STG50090
      COMMON/FFLD2/G1(400),G(400),GAMTRL(400)                           STG50100
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IK(40) STG50110
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),     STG50120
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)          STG50130
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         STG50140
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               STG50150
     2             XTR(40),XTL(40),CHR(40),CHL(40)                      STG50160
      COMMON/FCASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     STG50170
     1             XHB(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHTSTG50180
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)    STG50190
      COMMON/COMPOS/FACTOR(10,24),NCC                                   STG50200
      COMMON/TAILI/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HX,HH,VL,VH,     STG50210
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    STG50220
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES              STG50230
      DIMENSION VXJ(600),VYJ(600),VZJ(600)                              STG50240
C                                                                       STG50250
C  INITIAL LOADINGS AND CAMBERS                                         STG50260
      DO 20 J = 1,NEWMAX                                                STG50270
   20 G1(J) = 0.0                                                       STG50280
      DO 30 J = 1,NWT                                                   STG50290
   30 ALFAC(J) = 0.0                                                    STG50300
      DO 40 J = 1,NROWS                                                 STG50310
      ZL(J) = 0.0                                                       STG50320
      ALFINF(J) = 0.0                                                   STG50330
   40 ALFAT(J) = 0.0                                                    STG50340
C  SUM LOADINGS AND CAMBERS                                             STG50350
      DO 80 I = 1,10                                                    STG50360
      F = FACTOR(I,KCASE)                                               STG50370
      IF(F .EQ. 0.0)GO TO 80                                            STG50380
      DO 50  J = 1,NEWMAX                                               STG50390
   50 G1(J) = G1(J) + F * GAMMA(J,I)                                    STG50400
      DO 60  J = 1,NWT                                                  STG50410
   60 ALFAC(J) = ALFAC(J) + F * EPS(J,I) / 57.295779                    STG50420
      DO 70  J = 1,NROWS                                                STG50430


      ZL(J) = ZL(J) + F * HL(J,I)                                       STG50440
      ALFINF(J) = ALFINF(J) + F * AINF(J,I)                             STG50450
   70 ALFAT(J) = ALFAT(J) + F * THETA(J,I) / 57.295779                  STG50460
   80 CONTINUE                                                          STG50470
C  CYCLE ANGLE OF ATTACK                                                STG50480
C  FIND THE PLACE TO READ THE VECTORED JET INDUCED VELOCITIES           STG50490
      IREAD = 2 * (NMAX + NJT) + 3 * NTOTAL + 1                         STG50500
      DO 110 N = 1,5                                                    STG50510
      CALL HEADER(ILINES)                                               STG50520
      ALPHA = AL(N)                                                     STG50530
      CALL LINEAR(ALPHA,N)                                              STG50540
      ALPHA = AL(N) / 57.295779                                         STG50550
      CALL VORTEX(ALPHA)                                                STG50560
      CALL TRAIL(ALPHA)                                                 STG50570
      CALL COORS(ALPHA)                                                 STG50580
      ILINES = ILINES + 12                                              STG50590
C  READ VECTORED JET INDUCED VELOCITIES                                 STG50600
      IF(LOGIC .NE. 2) GO TO 90                                         STG50610
      CALL READMS(8,VXJ,600,IREAD)                                      STG50620
      IREAD = IREAD + 1                                                 STG50630
      CALL READMS(8,VYJ,600,IREAD)                                      STG50640
      IREAD = IREAD + 1                                                 STG50650
      CALL READMS(8,VZJ,600,IREAD)                                      STG50660
      IREAD = IREAD + 1                                                 STG50670
   90 DO 100 J = 1,NTPNP                                                STG50680
      ILINES = ILINES +1                                                STG50690
      SQ=SQUIG(J)                                                       STG50700
      ET=ETA(J)                                                         STG50710
      ZE=ZETA(J)                                                        STG50720
      JJ = NEWMAX + J                                                   STG50730
      CALL INDVEL(SQ,ET,ZE,VXJ(JJ),VYJ(JJ),VZJ(JJ),J,N,KCASE,ALPHA,     STG50740
     1     ILINES)                                                      STG50750
  100 CONTINUE                                                          STG50760
      CALL RHS1(ALPHA,N)                                                STG50770
  110 CONTINUE                                                          STG50780
      CONTINUE                                                          STG50790
      END                                                               STG50800


*DECK LINEAR                                                            LINR0010
      SUBROUTINE LINEAR(ALPHA,N)                                        LINR0020
C                                                                       LINR0030
C  THIS SUBROUTINE ADDS THE ANGLE OF ATTACK AND JET-INDUCED LOADS       LINR0040
C  TO THE THE WING LOADING                                              LINR0050
C                                                                       LINR0060
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                 LINR0070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            LINR0080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IX(40) LINR0090
      COMMON/SOLVI/GAMMA(400,10)                                        LINR0100
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),     LINR0110
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)          LINR0120
      COMMON/FFLD2/G1(400),G(400),GAMTRL(400)                           LINR0130
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         LINR0140
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               LINR0150
     2             XTR(40),XTL(40),CHR(40),CHL(40)                      LINR0160
      COMMON /FFLD5/ GAMJ(400,5)                                        LINR0170
      DO 20 I = 1,NEWMAX                                                LINR0180
   20 G(I) = G1(I) + ALPHA * GAMMA(I,1)                                 LINR0190
```

```
      IF((JETFLG .NE. 2) .OR. (LOGIC .EQ. 1)) GO TO 40
      DO 30 I = 1,NEWMAX
   30 G(I) = G(I) + GAMJ(I,N)
   40 DO 50 K = 1,NROWS
      ALFI(K) = ALFINF(K) + AINF(K,I) * ALPHA
   50 CONTINUE
      IF(TCR .LT. 0.00001)RETURN
      DO 60 I = 1,NWT
   60 G(I) = G(I) * (1.0 + TCR)
      RETURN
      END


*DECK VORTEX
      SUBROUTINE VORTEX(ALPHA)
C
C  THIS SUBROUTINE INTEGRATES EACH EVD TO ITS EQUIVALENT DISCRETE VORTEX
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XGC(400),XI(400),DEL(400),
     1             D(40),KK(400),ITYPE(400)
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)
      COMMON/FFLD2/GI(400),G(400),GAMTRL(400)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      BLOWIT = 0.0
      DO 20 K = 1,NROWS
   20 BLOWIT = BLOWIT + CMU(K)
      IF(BLOWIT .LT. 0.01)GO TO 40
      DO 30 K = 1,NROWS
   30 NN(K) = NJ(K)
      GO TO 60
   40 DO 50 K = 1,NROWS
   50 NN(K) = 0
   60 DO 70 I = 1,NEWMAX
   70 XC(I) = XGC(I)
      DO 100 K = 1,NROWS
C  ON THE WING
      I = IW(K)
C  LEADING EDGE ELEMENT
      G(I) = DEL(I) * CHORD(K) * (G(I) + 0.5 * G(I+1))
      LAST = IW(K) + NW(K) - 2
C  REGULAR WING ELEMENT
      I1 = I + 1
      DO 80 I = I1,LAST
      G(I) = DEL(I) * CHORD(K) *(G(I+1) + G(I)) / 2.0
   80 CONTINUE
      LAST = LAST + 1
      I = LAST
      IF(NN(K) .EQ. 0)GP = 0.00
      IF(NN(K) .NE. 0)GP = G(IJ(K))
      G(I) = DEL(I) * CHORD(K) * (GP + G(I)) / 2.0
C  ON THE JET
      IF(NN(K) .EQ. 0)GO TO 100
      I1 = IJ(K)
      LASTJ = IJ(K) + NJ(K) - 2
      DO 90 I = I1,LASTJ
      G(I) = DEL(I) * CHORD(K) * (G(I+1) + G(I)) / 2.0
   90 CONTINUE
C  INFINITY ELEMENT
      I = LASTJ + 1
      G(I) = G(I) * XC(I) * CHORD(K)
  100 CONTINUE
      RETURN
      END


*DECK TRAIL
      SUBROUTINE TRAIL(ALPHA)
C
C  THIS SUBROUTINE SUMS THE STRENGTHS OF THE TRAILING VORTICES
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)
      COMMON/FFLD2/GI(400),G(400),GAMTRL(400)
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),
     2             XTR(40),XTL(40),CHR(40),CHL(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),
     1             D(40),KK(400),ITYPE(400)
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE
      DO 40 K= 1,NROWS
      GAMSUM = 0.00
C  ON THE WING
      I1 = IW(K)
      LAST = IW(K) + NW(K) - 1
      DO 20 I = I1,LAST
      GAMTRL(I) = GAMSUM + G(I)
   20 GAMSUM = GAMTRL(I)
      IF(NN(K) .EQ. 0)GO TO 40
C  ON THE JET
      I1 = IJ(K)
      LAST = IJ(K) + NJ(K) - 1
      DO 30 I = I1,LAST
      GAMTRL(I) = GAMSUM + G(I)
   30 GAMSUM = GAMTRL(I)
   40 CONTINUE
      IF(IGND. GT. 0)GO TO 60
      DO 50  K = 1,NROWS
      IF(CMU(K) .GT. 0.0)GO TO 50
      LAST = IW(K) + NW(K) - 1
      ALFI(K) = 0.5 * (ALPHA + ALFAC(LAST))
   50 CONTINUE
      RETURN
   60 DO 70  K = 1,NROWS
   70 ALFI(K) = 0.0
      RETURN
      END


*DECK COORS
      SUBROUTINE COORS(ALPHA)
C
C  THIS SUBROUTINE SETS UP THE NON-PLANAR WING AND JET SHEET
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XGC(400),XI(400),DEL(400),
     1             D(40),KK(400),ITYPE(400)
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)
      COMMON/FFLD2/GI(400),G(400),GAMTRL(400)
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),
     2             XTR(40),XTL(40),CHR(40),CHL(40)
C
      SA = SIN(ALPHA)
      CA = COS(ALPHA)
C  ON THE WING
      DO 60 K = 1,NROWS
C  LEADING EDGE ELEMENT
      I = IW(K)
      XC(I) = XC(I) + 0.25 * DEL(I)
      X(I) = (XLR(K) + XC(I) * CHR(K)) * CA
      XN(I) = (XLL(K) + XC(I) * CHL(K)) * CA
      XSUM = X(I)
      XSUMN = XN(I)
      Z(I) = ZL(K) * CHORD(K) - (X(I) + XN(I)) *SA / 2.0
      ZSUM = Z(I)
      LAST = IW(K) + NW(K) - 1
C  REGULAR WING ELEMENTS
      I1 = I + 1
      DO 20 I = I1,LAST
      XC(I) = XC(I) + 0.25 * DEL(I)
      DIFF = XC(I) - XC(I-1)
      ANGLE = ALFAC(I-1) + ALPHA
      CANGLE = COS(ANGLE)
      X(I) = XSUM + DIFF * CHR(K) * CANGLE
      XN(I) = XSUMN + DIFF * CHL(K) * CANGLE
      Z(I) = ZSUM - SIN(ANGLE) * DIFF * CHORD(K)
      XSUM = X(I)
      XSUMN = XN(I)
      ZSUM = Z(I)
   20 CONTINUE
      IF(NN(K) .EQ. 0)GO TO 50
C  ON THE JET
      I = IJ(K)
C  FIRST JET ELEMENT
      XC(I) = XC(I) + 0.25 * DEL(I)
      DIFF = XC(I) - XC(LAST)
      ANGLE = ALFAC(LAST) + ALPHA
      CANGLE = COS(ANGLE)
      X(I) = XSUM + DIFF * CHR(K) * CANGLE
      XN(I) = XSUMN + DIFF * CHL(K) * CANGLE
      Z(I) = ZSUM - SIN(ANGLE) * DIFF * CHORD(K)
      W = -ALFAT(K) - ALPHA
      XSUM = X(I)
      XSUMN = XN(I)
      ZSUM = Z(I)
      LASTJ= IJ(K) + NJ(K) - 1
C  REGULAR JET ELEMENTS
      I1 = I + 1
      DO 40 I = I1,LASTJ
      IF(I .EQ. LASTJ)GO TO 30
      XC(I) = XC(I) + 0.25 * DEL(I)
   30 DIFF = XC(I) - XC(I-1)
      X(I) = XSUM + (XC(I) - 1.0) * CHR(K)
      XN(I) = XSUMN + (XC(I) - 1.0) * CHL(K)
      WB = W
      W = (2.0 * (GAMTRL(I-1) - GAMTRL(LAST)) / (CHORD(K) * CMU(K))) -
     1    ALFAT(K) - ALPHA
      Z(I) = ZSUM + DIFF * TAN((W + WB) / 2.0) * CHORD(K)
   40 ZSUM = Z(I)
      GO TO 60
   50 DIFF = 1.0 - XC(LAST)
      ANGLE = ALFAC(LAST) + ALPHA
      CANGLE = COS(ANGLE)
      ZT(K) = ZSUM - SIN(ANGLE) * DIFF * CHORD(K)
      XTR(K) = XSUM + DIFF * CHR(K) * CANGLE
      XTL(K) = XSUMN + DIFF * CHL(K) * CANGLE
   60 CONTINUE
      IF((IFFLD .NE. -9) .AND.(IFFLD .NE. 9))RETURN
      WRITE(6, 70 )
   70 FORMAT(1H0,54X,22H** FLOW FIELD PRINT **,//,
     1        30X,2HX1,11X,2HX2,11X,1HZ,9X,5HGAMMA,9X,6HGAMTRL,9X,
     2        5HALFAC,/)
      WRITE(6, 80 )(X(I),XN(I),Z(I),G(I),GAMTRL(I),ALFAC(I),I = 1,NWT)
   80 FORMAT(30X,6F12.6)
      IF(NWT .EQ. NEWMAX)RETURN
      I1 = NWT + 1
      WRITE(6, 90 )(X(I),XN(I),Z(I),G(I),GAMTRL(I),I = I1,NEWMAX)
   90 FORMAT(30X,5F12.6)
      RETURN
      END


*DECK INDVEL
      SUBROUTINE INDVEL(SQ,ET,ZE,VXJ,VYJ,VZJ,NP,N,KCASE,ALPHA,ILINES)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),
     2             XTR(40),XTL(40),CHR(40),CHL(40)
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XGC(400),XI(400),DEL(400),
     1             D(40),KK(400),ITYPE(400)
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),
     1     IHK(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),
     2             CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),
     3             CAMH(80),CAMV(20),DXH(80),DXV(20),DOWNH(200),
     4     SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)
      IF(NP .NE. 1)GO TO 40
      WRITE(6, 20 )KCASE,AL(N)
   20 FORMAT(1H0,T42,2H**,9(5H*****),/
     1     T42,42H*  FLOW FIELD SOLUTION FOR COMPOSITE CASE ,I2,3H  *,/
     2     T42,2H**,9(5H*****)//T54,8HALPHA = ,F6.2,
```

```
     3      9H DEGREES/)                                                INDV0240
      WRITE(6, 30 )                                                     INDV0250
   30 FORMAT(T21,1HX,T33,1HY,T45,1HZ,T55,6HU/UINF,T67,6HV/UINF,T79,      INDV0260
     1  6HW/UINF,T89,8HDOWNWASH,T101,8HSIDEWASH,/T91,5HANGLE,T103,      INDV0270
     2  5HANGLE,/T89,9H(DEGREES),T101,9H(DEGREES),/)                    INDV0280
      CA=COS(ALPHA)                                                     INDV0290
      SA=SIN(ALPHA)                                                     INDV0300
   40 SQP = SQ                                                          INDV0310
      ZEP = ZE                                                          INDV0320
      SQ = SQP * CA + ZEP * SA                                          INDV0330
      ZE = ZEP * CA - SQP * SA                                          INDV0340
      USUM = 0.00                                                       INDV0350
      VSUM = 0.00                                                       INDV0360
      WSUM = 0.00                                                       INDV0370
      DO 130 K = 1,NROWS                                                INDV0380
      FINISH = 0                                                        INDV0390
C   ON THE WING                                                         INDV0400
      I1 = IW(K)                                                        INDV0410
      LAST = IW(K) + NW(K) - 1                                          INDV0420
      DO 80 I = I1,LAST                                                 INDV0430
      X1 = X(I)                                                         INDV0440
      X2 = XN(I)                                                        INDV0450
      Z1 = Z(I)                                                         INDV0460
      Y1 = Y(K) + DELTA(K)                                              INDV0470
      Y2 = Y(K) - DELTA(K)                                              INDV0480
      IF(I .EQ. LAST)GO TO 50                                           INDV0490
      X3 = X(I+1)                                                       INDV0500
      X4 = XN(I+1)                                                      INDV0510
      Z2 = Z(I+1)                                                       INDV0520
      GO TO 70                                                          INDV0530
   50 IF(NN(K) .EQ. 0)GO TO 60                                          INDV0540
      X3 = X(IJ(K))                                                     INDV0550
      X4 = XN(IJ(K))                                                    INDV0560
      Z2 = Z(IJ(K))                                                     INDV0570
      GO TO 70                                                          INDV0580
   60 X3 = XTR(K)                                                       INDV0590
      X4 = XTL(K)                                                       INDV0600
      Z2 = ZT(K)                                                        INDV0610
   70 CALL BSLAW(X1,X2,X3,X4,Y1,Y2,Z1,Z2,USUM,VSUM,WSUM,FINISH,         INDV0620
     1            I,K,SQ,ET,ZE),RETURNS(180,200)                        INDV0630
   80 CONTINUE                                                          INDV0640
      IF(NN(K) .EQ. 0)GO TO 120                                         INDV0650
C   ON THE JET                                                          INDV0660
      I1 = IJ(K)                                                        INDV0670
      LASTJ = IJ(K) + NJ(K) - 1                                         INDV0680
      DO 110 I = I1,LASTJ                                               INDV0690
      X1 = X(I)                                                         INDV0700
      X2 = XN(I)                                                        INDV0710
      Z1 = Z(I)                                                         INDV0720
      IF(I .EQ. LASTJ)GO TO 90                                          INDV0730
      X3 = X(I+1)                                                       INDV0740
      X4 = XN(I+1)                                                      INDV0750
      Z2 = Z(I+1)                                                       INDV0760
      GO TO 100                                                         INDV0770
   90 FINISH = 1                                                        INDV0780
      X3 = 2.0 * X1                                                     INDV0790
      X4 = 2.0 * X2                                                     INDV0800
      Z2 = Z1 - TAN(ALFI(K)) * (X1 + X2) / 2.0                          INDV0810
  100 CALL BSLAW(X1,X2,X3,X4,Y1,Y2,Z1,Z2,USUM,VSUM,WSUM,FINISH,         INDV0820
     1            I,K,SQ,ET,ZE),RETURNS(180,200)                        INDV0830
```

```
  110 CONTINUE                                                          INDV0840
      GO TO 130                                                         INDV0850
  120 FINISH = 2                                                        INDV0860
      X1 = XTR(K)                                                       INDV0870
      X2 = XTL(K)                                                       INDV0880
      Z1 = ZT(K)                                                        INDV0890
      X3 = 2.0 * X1                                                     INDV0900
      X4 = 2.0 * X2                                                     INDV0910
      Z2 = Z1 - TAN(ALFI(K)) * (X1 + X2) / 2.0                          INDV0920
      CALL BSLAW(X1,X2,X3,X4,Y1,Y2,Z1,Z2,USUM,VSUM,WSUM,FINISH,         INDV0930
     1            LAST,K,SQ,ET,ZE),RETURNS(180,200)                     INDV0940
  130 CONTINUE                                                          INDV0950
C   ADD VECTORED JET INDUCED VELOCITIES                                 INDV0960
      IF(LOGIC .NE. 2) GO TO 140                                        INDV0970
      USUM = USUM + VXJ                                                 INDV0980
      VSUM = VSUM + VYJ                                                 INDV0990
      WSUM = WSUM + VZJ                                                 INDV1000
  140 EP = -ATAN(WSUM / (1.0 + USUM)) * 57.295779                       INDV1010
      BE = ATAN(VSUM / (1.0 + USUM)) * 57.295779                        INDV1020
      DOWN(NP) = -EP / 57.295779                                        INDV1030
      SIDE(NP) = BE / 57.295779                                         INDV1040
      IF(ILINES .LT. 55)GO TO 160                                       INDV1050
      WRITE(6, 150 )                                                    INDV1060
  150 FORMAT(1H1)                                                       INDV1070
      WRITE(6, 30 )                                                     INDV1080
      ILINES = 5                                                        INDV1090
  160 WRITE(6, 170 )NP,SQP,ET,ZEP,USUM,VSUM,WSUM,EP,BE                  INDV1100
  170 FORMAT(8X,I4,8F12.6)                                              INDV1110
      RETURN                                                            INDV1120
  180 WRITE(6, 190 )NP,SQP,ET,ZEP,I                                     INDV1130
  190 FORMAT(8X,I4,3F12.6,5X,40HCONTROL POINT IS ON BOUND VORTEX ELEMENTINDV1140
     1       ,8H NUMBER ,I3)                                            INDV1150
      DOWN(NP) = 0.00                                                   INDV1160
      SIDE(NP) = 0.00                                                   INDV1170
      RETURN                                                            INDV1180
  200 WRITE(6, 210 )NP,SQP,ET,ZEP,I                                     INDV1190
  210 FORMAT(8X,I4,3F12.6,5X,35HCONTROL POINT IS ON TRAILING VORTEX,    INDV1200
     1       16H ELEMENT NUMBER ,I3)                                    INDV1210
      DOWN(NP) = 0.00                                                   INDV1220
      SIDE(NP) = 0.00                                                   INDV1230
      RETURN                                                            INDV1240
      END                                                               INDV1250
```

```
*DECK BSLAW                                                             BSLA0010
      SUBROUTINE BSLAW(X1,X2,X3,X4,Y1,Y2,Z1,Z2,USUM,VSUM,WSUM,FINISH,   BSLA0020
     1                 I,K,SQ,ET,ZE),RETURNS(A,B)                       BSLA0030
C                                                                       BSLA0040
C  THIS SUBROUTINE CALCULATES THE VELOCITIES INDUCED BY A NON-PLANAR    BSLA0050
C  HORSESHOE VORTEX USING THE BIOT-SAVART LAW.                          BSLA0060
C                                                                       BSLA0070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            BSLA0080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BSLA0090
      COMMON/FFLD1/XC(400),X(400),XN(400),Z(400),ALFAC(400),ZL(40),     BSLA0100
     1             ZT(40),ALFAT(40),ALFINF(40),ALFI(40),NN(40)          BSLA0110
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         BSLA0120
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               BSLA0130
     2             XTR(40),XTL(40),CHR(40),CHL(40)                      BSLA0140
```

```
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          BSLA0150
      COMMON/FFLD7/GI(400),G(400),GAMTRL(400)                           BSLA0160
      DIMENSION VA(3),VB(3),VC(3),V1(3),V2(3),V3(3),V4(3)               BSLA0170
      LOGICAL NSYMM,XSYMM,IMAGE                                         BSLA0180
      DOT(A,B,C,D,E,F)=A*D+B*E+C*F                                      BSLA0190
      CROSS(A,B,C,D,E,F)=SQRT((B*F-E*C)**2+(D*C-A*F)**2+(A*E-D*B)**2)   BSLA0200
      VAL(A,B,C)=SQRT(A*A+B*B+C*C)                                      BSLA0210
      NSYMM = ISYMM .GT. 0                                              BSLA0220
      XSYMM = ISYMM .LT. 0                                              BSLA0230
      IMAGE = IGND .EQ. 0                                               BSLA0240
      SIGN = 1.0                                                        BSLA0250
      ASYMM = 1.0                                                       BSLA0260
      GROUND = 1.0                                                      BSLA0270
      VA(1)=X4-X2                                                       BSLA0280
      VA(2) = 0.0                                                       BSLA0290
      VA(3)=Z2-Z1                                                       BSLA0300
      VB(1)=X1-X2                                                       BSLA0310
      VB(2)=Y1-Y2                                                       BSLA0320
      VB(3)=0.0                                                         BSLA0330
      VC(1)=X3-X1                                                       BSLA0340
      VC(2) = 0.0                                                       BSLA0350
      VC(3)=Z2-Z1                                                       BSLA0360
      V1(1)=SQ-X4                                                       BSLA0370
      V2(1)=SQ-X2                                                       BSLA0380
      V3(1)=SQ-X1                                                       BSLA0390
      V4(1)=SQ-X3                                                       BSLA0400
      BL=VAL(VB(1),VB(2),VB(3))                                         BSLA0410
      TL1=VAL(VC(1),VC(2),VC(3))                                        BSLA0420
      TL2=VAL(VA(1),VA(2),VA(3))                                        BSLA0430
   20 V1(2)=ET-Y2                                                       BSLA0440
      V2(2) = V1(2)                                                     BSLA0450
      V3(2)=ET-Y1                                                       BSLA0460
      V4(2) = V3(2)                                                     BSLA0470
   30 V1(3)=ZE-Z2                                                       BSLA0480
      V2(3)=ZE-Z1                                                       BSLA0490
      V3(3)=V2(3)                                                       BSLA0500
      V4(3)=V1(3)                                                       BSLA0510
      R2T1=VAL(V4(1),V4(2),V4(3))                                       BSLA0520
      R2T2=VAL(V1(1),V1(2),V1(3))                                       BSLA0530
      R1T1=VAL(V3(1),V3(2),V3(3))                                       BSLA0540
      R1T2=VAL(V2(1),V2(2),V2(3))                                       BSLA0550
      IF((R1T1 .EQ. 0.0) .OR. (R1T2 .EQ. 0.0)) GO TO 140                BSLA0560
      RB=CROSS(V2(1),V2(2),V2(3),V3(1),V3(2),V3(3))/BL                  BSLA0570
      IF(RB .EQ. 0.0)GO TO 140                                          BSLA0580
      CT1B=DOT(V2(1),V2(2),V2(3),VB(1),VB(2),VB(3))/(R1T2*BL)           BSLA0590
      CT2B=DOT(V3(1),V3(2),V3(3),VB(1),VB(2),VB(3))/(R1T1*BL)           BSLA0600
      RT1=CROSS(V3(1),V3(2),V3(3),V4(1),V4(2),V4(3))/TL1                BSLA0610
      IF(RT1 .EQ. 0.0)GO TO 150                                         BSLA0620
      CT1T1=DOT(V3(1),V3(2),V3(3),VC(1),VC(2),VC(3))/(TL1*R1T1)         BSLA0630
      IF(FINISH .GT. 0)GO TO 40                                         BSLA0640
      CT2T1=DOT(V4(1),V4(2),V4(3),VC(1),VC(2),VC(3))/(TL1*R2T1)         BSLA0650
      GO TO 50                                                          BSLA0660
   40 CT2T1 = -1.0                                                      BSLA0670
   50 RT2 = CROSS(V2(1),V2(2),V2(3),V1(1),V1(2),V1(3)) / TL2            BSLA0680
      IF(RT2 .EQ. 0.0)GO TO 150                                         BSLA0690
      CT1T2=DOT(V2(1),V2(2),V2(3),VA(1),VA(2),VA(3))/(TL2*R1T2)         BSLA0700
      IF(FINISH .GT. 0)GO TO 60                                         BSLA0710
      CT2T2=DOT(V1(1),V1(2),V1(3),VA(1),VA(2),VA(3))/(TL2*R2T2)         BSLA0720
      GO TO 70                                                          BSLA0730
   60 CT2T2 = -1.0                                                      BSLA0740
```

```
   70 AB=VB(2)*V2(3)-V2(2)*VB(3)                                        BSLA0750
      BB=V2(1)*VB(3)-VB(1)*V2(3)                                        BSLA0760
      CB=V2(2)*VB(1)-V2(1)*VB(2)                                        BSLA0770
      AT1=VC(2)*V3(3)-V3(2)*VC(3)                                       BSLA0780
      BT1=V3(1)*VC(3)-VC(1)*V3(3)                                       BSLA0790
      CT1=VC(1)*V3(2)-V3(1)*VC(2)                                       BSLA0800
      AT2=VA(2)*V2(3)-VA(3)*V2(2)                                       BSLA0810
      BT2=V2(1)*VA(3)-VA(1)*V2(3)                                       BSLA0820
      CT2=VA(1)*V2(2)-V2(1)*VA(2)                                       BSLA0830
      DB=VAL(AB,BB,CB)                                                  BSLA0840
      DT1=VAL(AT1,BT1,CT1)                                              BSLA0850
      DT2=VAL(AT2,BT2,CT2)                                              BSLA0860
      IF(FINISH .LT. 2)GO TO 80                                         BSLA0870
      QB = 0.0                                                          BSLA0880
      GO TO 90                                                          BSLA0890
   80 QB = G(I) * (CT1B - CT2B) / (12.5663708 * RB)                     BSLA0900
   90 QT1 = GAMTRL(I) * (CT1T1 - CT2T1) / (12.5663708 * RT1)            BSLA0910
      QT2 = -GAMTRL(I) * (CT1T2 - CT2T2) / (12.5663708 * RT2)           BSLA0920
      USUM=USUM+ASYMM*(AB*QB/DB+AT1*QT1/DT1+AT2*QT2/DT2)*GROUND         BSLA0930
      VSUM=VSUM+ASYMM*SIGN*(BB*QB/DB+BT1*QT1/DT1+BT2*QT2/DT2)           BSLA0940
      WSUM=WSUM+ASYMM*(CB*QB/DB+CT1*QT1/DT1+CT2*QT2/DT2)*GROUND         BSLA0950
      IF(IMAGE)GO TO 110                                                BSLA0960
      IF(GROUND .LT. 0.0)GO TO 100                                      BSLA0970
      ZSAVE = ZE                                                        BSLA0980
      ZE = -2.0 * GHITE(NHITE) - ZE                                     BSLA0990
      GROUND = -1.0                                                     BSLA1000
      GO TO 30                                                          BSLA1010
  100 GROUND = 1.0                                                      BSLA1020
      ZE = ZSAVE                                                        BSLA1030
  110 IF(NSYMM)GO TO 130                                                BSLA1040
      IF(SIGN .LT. 0.0)GO TO 120                                        BSLA1050
      ET = -ET                                                          BSLA1060
      SIGN = -1.0                                                       BSLA1070
      IF(XSYMM)ASYMM = -1.0                                             BSLA1080
      GO TO 20                                                          BSLA1090
  120 SIGN = 1.0                                                        BSLA1100
      ASYMM = 1.0                                                       BSLA1110
      ET = - ET                                                         BSLA1120
  130 RETURN                                                            BSLA1130
  140 RETURN A                                                          BSLA1140
  150 RETURN B                                                          BSLA1150
      END                                                               BSLA1160
```

```
*DECK RHS1                                                              RHS10010
      SUBROUTINE RHS1(ALPHA,N)                                          RHS10020
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     RHS10030
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    RHS10040
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES              RHS10050
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),   RHS10060
     1       IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),     RHS10070
     2             CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),        RHS10080
     3             CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),         RHS10090
     4       SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)RHS10100
      COMMON/TAIL4/B(100,5)                                             RHS10110
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZLL(41),    RHS10120
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),               RHS10130
     2              V(40,7),W(40,7)                                     RHS10140
```

```
      COMMON/SLIPIT/SSLIP                                               RHS10140
      LOGICAL L1,L2                                                     RHS10150
      L1 = NROWH .EQ. 0                                                 RHS10160
      L2 = NROWV .EQ. 0                                                 RHS10170
   20 IF(L1)GO TO 40                                                    RHS10180
      DO 30 I = 1,NMAXH                                                 RHS10190
   30 B(I,N) =  DOWN(I) + CAMH(I) + ALPHA                               RHS10200
   40 IF(L2) GO TO 60                                                   RHS10210
      SSS = SIN(SSLIP/57.295779)                                        RHS10211
      DO 50 I=1,NMAXV                                                   RHS10220
      J = NMAXH + I                                                     RHS10230
   50 B(J,N) = SIDE(J) + CAMV(I) - SSS                                  RHS10241
   60 IF(NSEG .EQ. 0) RETURN                                            RHS10250
      N1 = NTOTAL                                                       RHS10260
      DO 70 K = 1,NSEG                                                  RHS10270
      N1 = N1 + 1                                                       RHS10280
      V(K,N) = SIDE(N1)                                                 RHS10290
   70 W(K,N) = DOWN(N1)                                                 RHS10300
      RETURN                                                            RHS10310
      END                                                               RHS10320
```

```
*DECK STAGE6
      OVERLAY(OVLY,6,0)                                                 STG60010
      PROGRAM STAGE6                                                    STG60020
C                                                                       STG60030
C  THIS SUBROUTINE CONTROLS GEOMETRY PREPARATION, DOWNWASH/SIDEWASH     STG60040
C  CALCULATIONS AND LOADING FOR THE HORIZONTAL AND VERTICAL TAIL        STG60050
C                                                                       STG60060
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     STG60070
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    STG60080
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             STG60090
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),    STG60100
     1       IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),     STG60110
     2              CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),        STG60120
     3              CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),         STG60130
     4       SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)STG60140
      COMMON/TAIL3/XH(80),XV(20),XCH(80),XCV(20),XLH(20),XTH(20),       STG60150
     1              XLV(10),XTV(10),YACH,ZACV                           STG60160
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         STG60170
     1              IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               STG60180
     2              XTR(40),XTL(40),CHR(40),CHL(40)                     STG60190
      COMMON/SOLV3/DUMMY(400)                                           STG60200
      DIMENSION W(100),WS(100),WD(100)                                  STG60210
      LOGICAL HOR,VERT                                                  STG60220
      HOR = NROWH .GT. 0                                                STG60230
      VERT = NROWV .GT. 0                                               STG60240
      GO TO ( 20 , 30 , 40 , 50 , 60 , 70 ),ISENSE                      STG60250
C  CALCULATE TRAPEZOIDAL PLANFORM GEOMETRIES                             STG60260
   20 IF(HOR) CALL TAILG(ARH,TRH,SWH,SPANH,NROWH,YH,XLH,XTH,CHH,        STG60270
     1                     AREAH,CREFH,XMH,YACH,2)                      STG60280
      IF(VERT) CALL TAILG(ARV,TRV,SWV,SPANV,NROWV,ZV,XLV,XTV,CHV,       STG60290
     1                     AREAV,CREFV,XMV,ZACV,3)                      STG60300
C  PRINT OUT TAIL GEOMETRY                                              STG60310
      CALL TAILP                                                        STG60320
C  DIVIDE UP THE TAIL AND DETERMINE THE DOWNWASH POINT LOCATIONS        STG60330
      CALL LATICE                                                       STG60340
C  SET UP THE LEFT HAND SIDE MATRIX FOR THE TAIL AND STORE IT ON UNIT 8 STG60350
                                                                        STG60360
```

```
      CALL LHS(W,WS,WD,NTOTAL)                                          STG60370
      GO TO 80                                                          STG60380
   30 N = 1                                                             STG60390
      CALL RHS2(N)                                                      STG60400
      NSIDE = N - 1                                                     STG60410
      GO TO 80                                                          STG60420
C  DETERMINE THE TAIL LOADING FOR THE LINEAR CASE                       STG60430
   40 CALL TAILL(1,DUMMY,NTOTAL)                                        STG60440
      GO TO 80                                                          STG60450
C  DETERMINE THE TAIL LOADING FOR THE NON-LINEAR TAIL CASE              STG60460
   50 NSIDE = 5                                                         STG60470
      CALL TAILL(2,DUMMY,NTOTAL)                                        STG60480
      GO TO 80                                                          STG60490
C  SET UP THE RIGHT HAND SIDE FOR THE ROLLING STABILITY CASE            STG60500
   60 NSIDE = 1                                                         STG60510
      CALL RHS2(N)                                                      STG60520
      GO TO 80                                                          STG60530
C  DETERMINE THE TAIL ROLLING (P) STABILITY DERIVATIVES                 STG60540
   70 NSIDE = 1                                                         STG60550
      CALL TAILL(3,DUMMY,NTOTAL)                                        STG60560
   80 CONTINUE                                                          STG60570
      END                                                               STG60580
```

```
*DECK TAILG
      SUBROUTINE TAILG(ARATIO,TRATIO,SWD,B,NROW,Y,X,XE,C,A,MAC,XMAC,    TALG0010
     1 YMAC,ITYPE)                                                      TALG0020
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     TALG0030
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    TALG0040
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             TALG0050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            TALG0060
      DIMENSION Y(21),X(21),Z(21),XE(21),C(21)                          TALG0070
      REAL MAC                                                          TALG0080
      LIKE = ISYMM                                                      TALG0090
      IF((ITYPE .EQ.2) .AND. (NROWV .GT. 0))LIKE = 1                    TALG0100
      SWR=SWD/57.295779                                                 TALG0110
      IF(ITYPE.NE.3)GO TO 20                                            TALG0120
      B=2.0*B                                                           TALG0130
      ARATIO=2.0*ARATIO                                                 TALG0140
   20 CROOT=2.0*B/((1.0+TRATIO)*ARATIO)                                 TALG0150
      B2=B/2.0                                                          TALG0160
      A=B2*(1.0+TRATIO)*CROOT                                           TALG0170
      XLB2=0.25*(1.0-TRATIO)*CROOT+B2*TAN(SWR)                          TALG0180
      MAC=2.0*CROOT*(1.0+TRATIO+TRATIO*TRATIO)/(3.0*(1.0+TRATIO))       TALG0190
      IF(TRATIO.EQ.1.00)GO TO 30                                        TALG0200
      YMAC=B2*(1.0-MAC/CROOT)/(1.0-TRATIO)                              TALG0210
      XMAC=0.25*MAC+XLB2*YMAC/B2                                        TALG0220
      GO TO 40                                                          TALG0230
   30 XMAC=0.25*MAC                                                     TALG0240
      YMAC=0.0                                                          TALG0250
   40 IF(ITYPE.EQ.3)GO TO 70                                            TALG0260
      IF(LIKE.GT.0)GO TO 90                                             TALG0270
   50 DO 60 K=1,NROW                                                    TALG0280
      X(K)=XLB2*Y(K)/B2                                                 TALG0290
      C(K)=CROOT*(1.0-(1.0-TRATIO)*Y(K)/B2)                             TALG0300
   60 XE(K)=X(K)+C(K)                                                   TALG0310
      RETURN                                                            TALG0320
   70 DO 80 K=1,NROW                                                    TALG0330
                                                                        TALG0340
```

```
      X(K)=XLB2*Y(K)/B2                                                 TALG0350
      C(K)=CROOT*(1.0-(1.0-TRATIO)*Y(K)/B2)                             TALG0360
   80 XE(K)=X(K)+C(K)                                                   TALG0370
      RETURN                                                            TALG0380
   90 DO 100 K=1,NROW                                                   TALG0390
      YBAR=ABS(Y(K))                                                    TALG0400
      X(K)=XLB2*YBAR/B2                                                 TALG0410
      C(K)=CROOT*(1.0-(1.0-TRATIO)*YBAR/B2)                             TALG0420
  100 XE(K)=X(K)+C(K)                                                   TALG0430
      RETURN                                                            TALG0440
      END                                                               TALG0450
```

```
*DECK TAILP
      SUBROUTINE TAILP                                                  TALP0010
C  THIS SUBROUTINE PRINTS OUT A SUMMARY OF HORIZONTAL AND VERTICAL TAIL TALP0020
C  GEOMETRIC PARAMETERS                                                 TALP0030
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     TALP0040
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     TALP0050
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    TALP0060
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             TALP0070
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),    TALP0080
     1       IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),     TALP0090
     2              CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),        TALP0100
     3              CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),         TALP0110
     4       SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)TALP0120
      COMMON/TAIL3/XH(80),XV(20),XCH(80),XCV(20),XLH(20),XTH(20),       TALP0130
     1              XLV(10),XTV(10),YACH,ZACV                           TALP0140
      CALL HEADER(ILINES)                                               TALP0150
      IF(NROWH .EQ. 0)GO TO 60                                          TALP0160
      WRITE(6, 20 )                                                     TALP0170
   20 FORMAT(1H0,46X,1H*,7(5H*****),/,                                  TALP0180
     1            47X,36H* HORIZONTAL TAIL GEOMETRY SUMMARY *,/,        TALP0190
     2            47X,1H*,7(5H*****),/)                                 TALP0200
      XMCH = XMH + HL                                                   TALP0210
      HLENTH = XMCH - XMC                                               TALP0220
      VOLH = (HLENTH * AREAH) / (CREF * AREA)                           TALP0230
      WRITE(6, 30 )ARH,TRH,SWH,SPANH,AREAH,CREFH,XMCH,YACH,HLENTH,HH,   TALP0240
     1             VOLH,NROWH                                           TALP0250
   30 FORMAT(51X,15HASPECT RATIO = ,F12.6/52X,14HTAPER RATIO = ,F12.6/  TALP0260
     1        58X,8HSWEEP = ,F12.6/59X,7HSPAN = ,F12.6/59X,7HAREA = ,   TALP0270
     2        F12.6/60X,6HMAC = ,F12.6/60X,6HXMC = ,F12.6/60X,6HYMC = , TALP0280
     3        F12.6/57X,9HLENGTH = ,F12.6/57X,9HHEIGHT = ,F12.6/57X,    TALP0290
     4        9HVOLUME = ,F12.6/58X,8HNROWH = ,3X,I2//                  TALP0300
     5        40X,3HROW,3X,5HYLEAD,7X,5HXLEAD,7X,6HXTRAIL,7X,5HCHORD)   TALP0310
      ILINES = ILINES + 19                                              TALP0320
      DO 50 K = 1,NROWH                                                 TALP0330
      XLEAD = XLH(K) + HL                                               TALP0340
      XTRAIL = XTH(K) + HL                                              TALP0350
      WRITE(6, 40 )K,YH(K),XLEAD,XTRAIL,CHH(K)                          TALP0360
   40 FORMAT(39X,I3,4F12.6)                                             TALP0370
      ILINES = ILINES + 1                                               TALP0380
   50 CONTINUE                                                          TALP0390
   60 IF(NROWV .EQ. 0)RETURN                                            TALP0400
      SPANV = 0.5 * SPANV                                               TALP0410
      ARV = 0.5 * ARV                                                   TALP0420
      AREAV = 0.5 * AREAV                                               TALP0430
      IF(ZACV .EQ. 0.0) ZACV = VH + 0.5 * SPANV                         TALP0440
                                                                        TALP0450
```

```
      XMCV = XMV + VL                                                   TALP0460
      VLENTH = XMCV - XMC                                               TALP0470
      VOLV = (VLENTH * AREAV) / (SPAN * AREA)                           TALP0480
      IF((ILINES + NROWV + 18) .GT. 60) GO TO 80                        TALP0490
      WRITE(6, 70 )                                                     TALP0500
   70 FORMAT(1H0/)                                                      TALP0510
      GO TO 90                                                          TALP0520
   80 CALL HEADER(ILINES)                                               TALP0530
   90 WRITE(6, 100 )                                                    TALP0540
  100 FORMAT(1H0,47X,4H****,6(5H*****),/,                               TALP0550
     1            48X,34H* VERTICAL TAIL GEOMETRY SUMMARY */,           TALP0560
     2            48X,4H****,6(5H*****),/)                              TALP0570
      WRITE(6, 110 )ARV,TRV,SWV,SPANV,AREAV,CREFV,XMCV,ZACV,VLENTH,VH,  TALP0580
     1             VOLV,NROWV                                           TALP0590
  110 FORMAT(51X,15HASPECT RATIO = ,F12.6/52X,14HTAPER RATIO = ,F12.6/  TALP0600
     1        58X,8HSWEEP = ,F12.6/59X,7HSPAN = ,F12.6/59X,7HAREA = ,   TALP0610
     2        F12.6/60X,6HMAC = ,F12.6/60X,6HXMC = ,F12.6/60X,6HZMC =   TALP0620
     3        F12.6/57X,9HLENGTH = ,F12.6/57X,9HHEIGHT = ,F12.6/57X,    TALP0630
     4        9HVOLUME = ,F12.6/58X,8HNROWV = ,3X,I2//                  TALP0640
     5        40X,3HROW,3X,5HZLEAD,7X,5HXLEAD,7X,6HXTRAIL,7X,5HCHORD)   TALP0650
      DO 120 K = 1,NROWV                                                TALP0660
      ZLEAD = ZV(K) + VH                                                TALP0670
      XLEAD = XLV(K) + VL                                               TALP0680
      XTRAIL = XTV(K) + VL                                              TALP0690
      WRITE(6, 40 )K,ZLEAD,XLEAD,XTRAIL,CHV(K)                          TALP0700
  120 CONTINUE                                                          TALP0710
      RETURN                                                            TALP0720
      END                                                               TALP0730
```

```
*DECK LATICE
      SUBROUTINE LATICE                                                 LATT0010
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         LATT0020
     1              IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),               LATT0030
     2              XTR(40),XTL(40),CHR(40),CHL(40)                     LATT0040
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     LATT0050
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,    LATT0060
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             LATT0070
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),    LATT0080
     1       IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),     LATT0090
     2              CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),        LATT0100
     3              CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),         LATT0110
     4       SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)LATT0120
      COMMON/TAIL3/XH(80),XV(20),XCH(80),XCV(20),XLH(20),XTH(20),       LATT0130
     1              XLV(10),XTV(10),YACH,ZACV                           LATT0140
      COMMON/SLIPIT/SSLIP                                               LATT0150
      LOGICAL L1,L2,L3,L4                                               LATT0151
      L1 = NROWH .EQ. 0                                                 LATT0160
      L2 = NROWV .EQ. 0                                                 LATT0170
      L3 = ICAMH .NE. 0                                                 LATT0180
      L4 = ICAMV .NE. 0                                                 LATT0190
      SSS = SIN(SSLIP/57.295779)                                        LATT0200
      CSS = COS(SSLIP/57.295779)                                        LATT0201
      IF(L1)GO TO 50                                                    LATT0202
      IC1 = 0                                                           LATT0210
      IC2 = 0                                                           LATT0220
      DO 40 K = 1,NROWH                                                 LATT0230
      IXH = IHX(K)                                                      LATT0240
                                                                        LATT0250
```

```
      ICK = IHC(K)
      NH(K) = NX(IKK)
      NHK = NH(K)
      DO 20 I = 1,NHK
      IC1= IC1+ 1
      XGH(IC1)= XBT(I,IXK)
      IF(L3)CAMH(IC1) = CAMBER(I,ICK) / 57.295779
   20 CONTINUE
      DO 30 I = 1,NHK
      IC2 = IC2 + 1
      IROWH(IC2) = K
      IF(I .LT. NHK)DXH(IC2) = XGH(IC2+1) - XGH(IC2)
      IF(I .EQ. NHK)DXH(IC2) = 1.0 - XGH(IC2)
      XH(IC2) = XLH(K) + (XGH(IC2) + 0.25 * DXH(IC2)) * CHH(K)
      XCH(IC2) = XH(IC2) + 0.5 * DXH(IC2) * CHH(K)
      DX = HL + XCH(IC2)
      SQUIG(IC2) = DX * CSS + YH(K) * SSS
      ETA(IC2) = -DX * SSS + YH(K) * CSS
   30 ZETA(IC2) = HH
   40 CONTINUE
   50 IF(L2)RETURN
      IC1 = 0
      IC2 = 0
      DO 80 K = 1,NROWV
      IXK = IVK(K)
      ICK = IVC(K)
      NV(K) = NK(IXK)
      NVK = NV(K)
      DO 60 I = 1,NVK
      IC1 = IC1 + 1
      XGV(IC1) = XBT(I,IXK)
      IF(L4)CAMV(IC1) = CAMBER(I,ICK) / 57.295779
   60 CONTINUE
      DO 70 I = 1,NVK
      IC2 = IC2 + 1
      II = IC2 + NMAXH
      IROWV(IC2) = K
      IF(I .LT. NVK)DXV(IC2) = XGV(IC2+1) - XGV(IC2)
      IF(I .EQ. NVK)DXV(IC2) = 1.0 - XGV(IC2)
      XV(IC2) = XLV(K) + (XGV(IC2) + 0.25 * DXV(IC2)) * CHV(K)
      XCV(IC2) = XV(IC2) + 0.5 * DXV(IC2) * CHV(K)
      DX = VL + XCV(IC2)
      SQUIG(II) = DX * CSS
      ETA(II) = DX * SSS
   70 ZETA(II) = ZV(K) + VH
   80 CONTINUE
      RETURN
      END


*DECK LHS
      SUBROUTINE LHS(W,WS,WD,ITOTAL)
C     THIS SUBROUTINE SETS UP THE LEFT HAND SIDE OF THE MATRIX TO SOLVE
C     THE HORIZONTAL/VERTICAL TAIL PROBLEM.
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),
     1      IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),
     2             CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),
     3             CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),
     4      SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)
      COMMON/TAIL3/XH(80),XV(20),XCH(80),XCV(20),XLH(20),XTH(20),
     1             XLV(10),XTV(10),YACH,ZACV
      COMMON/TAIL4/B(100,5)
      DIMENSION VA(3),VB(3),VC(3),V1(3),V2(3),V3(3)
      DIMENSION W(ITOTAL),WS(ITOTAL),WD(ITOTAL)
      LOGICAL PRINT,STAB
      FCN(A,B) = SQRT(A*A + B*B)
      FCN1(A,B,C) = SQRT(A*A + B*B + C*C)
      DOT(A,B,C,D,E,F) = A*D + B*E + C*F
      CROSS(A,B,C,D,E,F) = SQRT((B*F - E*C)**2 + (D*C - A*F)**2 +
     1 (A*E - D*B)**2)
      PRINT = IPRINT .GE. 0
      STAB = LOGIC .EQ. 2
      IWRITE = 2 * (NMAX + NJT) - 2
      IF(PRINT) GO TO 30
      WRITE(6, 20 )
   20 FORMAT(1H1,57X,17HTAIL MATRIX PRINT)
      ILINES = 1
      LINES = (NMAXH+NMAXV)/10 + 3
      IF(STAB) LINES = LINES + (NMAXH+NMAXV)/10 + 1
   30 LIKE = ISYMM
      IF((NROWH .GT. 0) .AND. (NROWV .GT. 0)) LIKE = 1
      VB(1)=0.0
      FOURPI = 12.5663708
      A = 1.0
      DL = 0.0
      DH = 0.0
      IF(NROWV .EQ. 0)GO TO 40
      DL = HL - VL
      DH = HH - VH
C     THE LHS IS OF ORDER NTOTAL.
C     I = CONTROL STATION
C     J = INFLUENCING STATION
C     CYCLE THE CONTROL STATIONS
   40 DO 230 I = 1,NTOTAL
      IWRITE = IWRITE + 3
      IF(I .GT. NMAXH)GO TO 50
      SQ = XCH(I) + DL
      SQD = XH(I) + DL
      K = IROWH(I)
      ET = YH(K)
      ZE = DH
      GO TO 60
   50 II = I - NMAXH
      SQ = XCV(II)
      SQD = XV(II)
      K = IROWV(II)
      ET = 0.0
      ZE = ZV(K)
C     CYCLE THE INFLUENCING STATIONS
   60 DO 170 J = 1,NTOTAL
      W(J) = 0.0
      WS(J) = 0.0
      WD(J) = 0.0
   70 IF(J .GT. NMAXH)GO TO 80
      K = IROWH(J)
      VB(2) = 2.0 * DELTAH(K)
      VB(3) = 0.0
      V2(1) = SQ - (XH(J) + DL)
      VD21 = SQD - (XH(J) + DL)
      V2(2) = ET - (YH(K) - DELTAH(K))
      V2(3) = ZE - DH
      V3(1) = V2(1)
      VD31 = VD21
      V3(2) = ET - (YH(K) + DELTAH(K))
      V3(3) = V2(3)
      GO TO 90
   80 JJ = J - NMAXH
      K = IROWV(JJ)
      VB(2) = 0.0
      VB(3) = 2.0 * DELTAV(K)
      V2(1) = SQ - XV(JJ)
      VD21 = SQD - XV(JJ)
      V2(2) = ET
      V2(3) = ZE - (ZV(K) - DELTAV(K))
      V3(1) = V2(1)
      VD31 = VD21
      V3(2) = V2(2)
      V3(3) = ZE - (ZV(K) + DELTAV(K))
   90 RL = FCN1(V2(1),V2(2),V2(3))
      RR = FCN1(V3(1),V3(2),V3(3))
      BL = FCN(VB(2),VB(3))
      IF(J .LE. NMAXH)RB = FCN(V2(1),V2(3))
      IF(J .GT. NMAXH)RB = FCN(V2(1),V2(2))
      IF(RB .EQ. 0.0) RB = 1.0E10
      RTL = FCN(V2(2),V2(3))
      RTR = FCN(V3(2),V3(3))
      CT1B = DOT(V2(1),V2(2),V2(3),VB(1),VB(2),VB(3))/(RL*BL)
      CT2B = DOT(V3(1),V3(2),V3(3),VB(1),VB(2),VB(3))/(RR*BL)
      CT1L = V2(1)/RL
      CT1R = V3(1)/RR
      QB = (CT1B - CT2B)/(FOURPI * RB)
      QTL = (CT1L + 1.0)/(FOURPI * RTL)
      QTR = (CT1R + 1.0)/(FOURPI * RTR)
C  REPEAT THESE CALCULATIONS FOR TAIL INDUCED DRAG
      RL = FCN1(VD21,V2(2),V2(3))
      RR = FCN1(VD31,V3(2),V3(3))
      IF(J .LE. NMAXH)RBD = FCN(VD21,V2(3))
      IF(J .GT. NMAXH)RBD = FCN(VD21,V2(2))
      IF(RBD .EQ. 0.0) RBD = 1.0E10
      CT1B = DOT(VD21,V2(2),V2(3),VB(1),VB(2),VB(3))/(RL*BL)
      CT2B = DOT(VD31,V3(2),V3(3),VB(1),VB(2),VB(3))/(RR*BL)
      CT1L = VD21 / RL
      CT1R = VD31 / RR
      QBD = (CT1B - CT2B)/(FOURPI * RBD)
      QTLD = (CT1L + 1.0)/(FOURPI * RTL)
      QTRD = (CT1R + 1.0)/(FOURPI * RTR)
C     FORM THE PROPER DOWNWASH OR SIDEWASH INFLUENCE COEFFICIENT
  100 IF(I .GT. NMAXH) GO TO 120
      IF(J .GT. NMAXH)GO TO 110
C     HORIZONTAL - ON - HORIZONTAL
      WSUM   = V2(1)*QB/RB + V2(2)*QTL/RTL - V3(2)*QTR/RTR
      WDSUM=VD21*QBD/RBD+V2(2)*QTLD/RTL-V3(2)*QTRD/RTR
      GO TO 140
C     VERTICAL - ON - HORIZONTAL
  110 WSUM   =(V2(2)*QTL/RTL - V3(2)*QTR/RTR)*A
      WDSUM=(V2(2)*QTLD/RTL-V3(2)*QTRD/RTR)*A
      GO TO 140
  120 IF(J .GT. NMAXH)GO TO 130
C     HORIZONTAL - ON - VERTICAL
      WSUM   =(V2(3)*QTL/RTL - V3(3)*QTR/RTR)*A
      WDSUM=(V2(3)*QTLD/RTL-V3(3)*QTRD/RTR)*A
      GO TO 140
C     VERTICAL - ON - VERTICAL
  130 WSUM   = V2(1)*QB/RB + V2(3)*QTL/RTL - V3(3)*QTR/RTR
      WDSUM=VD21*QBD/RBD+V2(3)*QTLD/RTL-V3(3)*QTRD/RTR
  140 W(J) = W(J) + WSUM
      IF(STAB) WS(J) = WS(J) + A * WSUM
      WD(J) = WD(J) + WDSUM
      IF(J .GT. NMAXH) GO TO 150
      IF(LIKE .GT. 0) GO TO 170
  150 IF(A .LT. 0.0)GO TO 160
      IF(J .LE. NMAXH) ET = -ET
      IF(J .GT. NMAXH) ZE = -ZE
      A = -1.0
      GO TO 70
  160 A = 1.0
      IF(J .LE. NMAXH) ET = -ET
      IF(J .GT. NMAXH) ZE = -ZE
  170 CONTINUE
      CALL WRITMS(8,W,ITOTAL,IWRITE)
      IWR1 = IWRITE + 1
      IF(STAB) CALL WRITMS(8,WS,ITOTAL,IWR1)
      IWR1 = IWR1 + 1
      CALL WRITMS(8,WD,ITOTAL,IWR1)
      IF(PRINT) GO TO 230
      ILINES = ILINES + LINES
      IF(ILINES .LT. 61) GO TO 190
      WRITE(6, 180 )
  180 FORMAT(1H1)
      ILINES = LINES + 1
  190 WRITE(6, 200 )I
  200 FORMAT(1H0,54X,16HTAIL MATRIX ROW ,I3)
      WRITE(6, 210 )(W(JJ),JJ=1,NTOTAL)
  210 FORMAT(2X,10E13.5)
      IF(STAB) WRITE(6, 220 )I
  220 FORMAT(1H0,50X,26HTAIL STABILITY MATRIX ROW ,I3)
      IF(STAB) WRITE(6, 210 )(WS(JJ),JJ = 1,NTOTAL)
  230 CONTINUE
      RETURN
      END


*DECK RHS2
      SUBROUTINE RHS2(N)
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1             IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),
     2             XTR(40),XTL(40),CHR(40),CHL(40)
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,
```

```
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,   RHS20090
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             RHS20100
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),   RHS20110
     1      IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),      RHS20120
     2              CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),       RHS20130
     3              CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),        RHS20140
     4      SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)RHS20150
      COMMON/TAIL4/B(100,5)                                             RHS20160
      LOGICAL HOR,VERT                                                  RHS20170
      HOR = NROWH .GT. 0                                                RHS20180
      VERT = NROWV .GT. 0                                               RHS20190
   20 DEGREE = 0.01745329                                               RHS20200
      J = NMAXH + 1                                                     RHS20210
      IF(ISENSE .EQ. 5) GO TO 170                                       RHS20220
C  SET UP THE RIGHT HAND SIDE FOR THE TAIL ALPHA CASE                   RHS20230
      IF(.NOT. HOR) GO TO 50                                            RHS20240
      DO 30 I = 1,NMAXH                                                 RHS20250
   30 B(I,N) = DEGREE                                                   RHS20260
      DO 40 I = J,NTOTAL                                                RHS20270
   40 B(I,N) = 0.0                                                      RHS20280
      N = N + 1                                                         RHS20290
C  SET UP THE RIGHT HAND SIDE FOR THE TAIL BETA CASE                    RHS20300
   50 IF(.NOT. VERT) GO TO 80                                           RHS20310
      DO 60 I = 1,NMAXH                                                 RHS20320
   60 B(I,N) = 0.0                                                      RHS20330
      DO 70 I = J,NTOTAL                                                RHS20340
   70 B(I,N) = DEGREE                                                   RHS20350
      N = N + 1                                                         RHS20360
C  SET UP THE RIGHT HAND SIDE FOR THE TAIL CAMBER CASE                  RHS20370
   80 IF((ICAMH .EQ. 0) .AND. (ICAMV .EQ. 0))GO TO 110                  RHS20380
      DO 90 I = 1,NMAXH                                                 RHS20390
   90 B(I,N) = CAMH(I)                                                  RHS20400
      II = 0                                                            RHS20410
      DO 100 I = J,NTOTAL                                               RHS20420
      II = II + 1                                                       RHS20430
  100 B(I,N) = CAMV(II)                                                 RHS20440
      N = N + 1                                                         RHS20450
C  SET UP THE RIGHT HAND SIDE FOR THE TAIL PITCHING CASE                RHS20460
  110 IF(ISTAB .NE. 1) RETURN                                           RHS20470
      IF(.NOT. HOR) GO TO 140                                           RHS20480
      DO 120 I = 1,NMAXH                                                RHS20490
  120 B(I,N) = 2.0* (SQUIG(I) - XCG) / CREF                             RHS20500
      DO 130 I = J,NTOTAL                                               RHS20510
  130 B(I,N) = 0.0                                                      RHS20520
      N = N + 1                                                         RHS20530
C  SET UP THE RIGHT HAND SIDE FOR THE TAIL YAWING CASE                  RHS20540
  140 IF(.NOT. VERT) RETURN                                             RHS20550
      DO 150 I = 1,NMAXH                                                RHS20560
  150 B(I,N) = 0.0                                                      RHS20570
      DO 160 I = J,NTOTAL                                               RHS20580
  160 B(I,N) = -(SQUIG(I) - XCG)                                        RHS20590
      N = N + 1                                                         RHS20600
      RETURN                                                            RHS20610
C  SET UP THE RIGHT HAND SIDE FOR THE TAIL ROLLING CASE                 RHS20620
  170 IF(.NOT. HOR) GO TO 190                                           RHS20630
      DO 180 I = 1,NMAXH                                                RHS20640
      K = IROWH(I)                                                      RHS20650
  180 B(I,1) = YH(K)                                                    RHS20660
  190 IF(.NOT. VERT) RETURN                                             RHS20670
      II = 0                                                            RHS20680
```

```
      DO 200 I = J,NTOTAL                                               RHS20690
      II = II + 1                                                       RHS20700
      K = IROWV(II)                                                     RHS20710
  200 B(I,1) = ZV(K)                                                    RHS20720
      RETURN                                                            RHS20730
      END                                                               RHS20740
```

```
*DECK TAILL                                                             TALL0010
      SUBROUTINE TAILL(ITYPE,DUMMY,ITOTAL)                              TALL0020
C  THIS SUBROUTINE INTEGRATES THE LOADING ON THE TAIL SURFACES FOR      TALL0030
C  AERODYNAMIC COEFFICIENTS AND STABILITY AND CONTROL DERIVATIVES       TALL0040
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                TALL0050
      COMMON/MARK/NROWS,NROWS1,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) TALL0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            TALL0070
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     TALL0080
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     TALL0090
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,   TALL0100
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             TALL0110
      COMMON/TAIL2/YH(20),ZV(10),DELTAH(20),DELTAV(10),NH(20),NV(10),   TALL0120
     1      IHX(20),IVX(10),IHC(20),IVC(10),NX(5),NC(5),XBT(10,5),      TALL0130
     2              CAMBER(10,5),CHH(20),CHV(10),XGH(80),XGV(20),       TALL0140
     3              CAMH(80),CAMV(20),DXH(80),DXV(20),DOWN(200),        TALL0150
     4      SIDE(200),XMH,XMV,CREFH,CREFV,AREAH,AREAV,IROWH(80),IROWV(20)TALL0160
      COMMON/TAIL3/XH(80),XV(20),XCH(80),XCV(20),XLH(20),XTH(20),       TALL0170
     1              XLV(10),XTV(10),YACH,ZACV                           TALL0180
      COMMON/TAIL4/CP(100,5)                                            TALL0190
      COMMON/FFLO3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),         TALL0200
     1              IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),              TALL0210
     2              XTR(40),XTL(40),CHR(40),CHL(40)                     TALL0220
      COMMON/KEEPEM/CLW(5),CMW(5),CLLW(5),CDW(5),CNW(5),CYW(5),CLOW,    TALL0230
     1          CMOW,CLLPW,CLLRW,CNPW,CNRW,CYPW,CYRW,CLH(5),CMH(5),CRH (5),TALL0240
     2          CDH(5),CLOH,CMOH,CLLPH,CYV(5),CYBV,CNV(5),CNBV,CRV (5),   TALL0250
     3          CRBV ,CDV(5),CLLPV,CLLRV,CNPV,CNRV,CYPV,CYRV,CMF(5),CNF(5),TALL0260
     4          CMOF,CNRF                                               TALL0270
      DIMENSION CLLH(20,5),CLMH(20,5),CLDH(20,5)                        TALL0280
      DIMENSION CLYV(10,5),CLNV(10,5),CLDV(10,5)                        TALL0290
      DIMENSION WD(100,5),DUMMY(ITOTAL)                                 TALL0300
      LOGICAL HOR,VERT,STAB,PRINT                                       TALL0310
      HOR = NROWH .GT. 0                                                TALL0320
      VERT = NROWV .GT. 0                                               TALL0330
      STAB = ISTAB .EQ. 1                                               TALL0340
      PRINT = IPRINT .GT. 0                                             TALL0350
      LIKE = ISYMM                                                      TALL0360
      IF(HOR .AND. VERT) LIKE = 1                                       TALL0370
C  DEFINE THE INDUCED DOWNWASH NEEDED TO CALCULATE TAIL INDUCED DRAG    TALL0380
      IREAD = 2* (NMAX+NJT)                                             TALL0390
      DO 30  I = 1,NTOTAL                                               TALL0400
      DO 20  N = 1,NSIDE                                                TALL0410
   20 WD(I,N) = 0.0                                                     TALL0420
   30 CONTINUE                                                          TALL0430
      IF(ITYPE .EQ. 3) GO TO 70                                         TALL0440
      DO 60  I = 1,NTOTAL                                               TALL0450
      IREAD = IREAD + 3                                                 TALL0460
      CALL READMS(8,DUMMY,ITOTAL,IREAD)                                 TALL0470
      DO 50  J = 1,NTOTAL                                               TALL0480
      DO 40  N = 1,NSIDE                                                TALL0490
   40 WD(I,N) = WD(I,N) + DUMMY(J) * CP(J,N)                            TALL0500
```

```
   50 CONTINUE                                                          TALL0510
   60 CONTINUE                                                          TALL0520
C  CALCULATE THE CHORDWISE LOADING                                      TALL0530
   70 DO 110 N = 1,NSIDE                                                TALL0540
      IF(.NOT. HOR)GO TO 90                                             TALL0550
      DO 80 I = 1,NMAXH                                                 TALL0560
      K = IROWH(I)                                                      TALL0570
   80 CP(I,N) = 2.0 * CP(I,N) / (DXH(I) * CHH(K))                       TALL0580
   90 IF(.NOT. VERT)GO TO 110                                           TALL0590
      DO 100 I = 1,NMAXV                                                TALL0600
      II = I + NMAXH                                                    TALL0610
      K = IROWV (I)                                                     TALL0620
  100 CP(II,N) = 2.0 * CP(II,N) / (DXV(I) * CHV(K))                     TALL0630
  110 CONTINUE                                                          TALL0640
C  CALCULATE SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES              TALL0650
      DO 190 N = 1,NSIDE                                                TALL0660
      GO TO (750,760,750),ITYPE                                         TALL0661
  750 SA = 0.0                                                          TALL0662
      CA = 1.0                                                          TALL0663
      GO TO 770                                                         TALL0664
  760 SA = SIN(AL(N) / 57.295779)                                       TALL0665
      CA = COS(AL(N) / 57.295779)                                       TALL0666
C  HORIZONTAL TAIL                                                      TALL0670
  770 IF(.NOT. HOR) GO TO 140                                           TALL0680
      I = 0                                                             TALL0690
      CLH(N) = 0.0                                                      TALL0700
      CMH(N) = 0.0                                                      TALL0710
      CDH(N) = 0.0                                                      TALL0720
      CRH(N) = 0.0                                                      TALL0730
      DO 130 K = 1,NROWH                                                TALL0740
      CLLH(K,N) = 0.0                                                   TALL0750
      CLMH(K,N) = 0.0                                                   TALL0760
      CLDH(K,N) = 0.0                                                   TALL0770
      NWK = NH(K)                                                       TALL0780
      DO 120 J = 1,NWK                                                  TALL0790
      I = I + 1                                                         TALL0800
      CLLH(K,N) = CLLH(K,N) + CP(I,N) * DXH(I)                          TALL0810
      CLMH(K,N) = CLMH(K,N) - CP(I,N) * DXH(I) * (XGH(I) + 0.25*DXH(I)) TALL0820
  120 CLDH(K,N) = CLDH(K,N) + CP(I,N) * WD(I,N)                         TALL0830
      CDEL = CHH(K) * DELTAH(K)                                         TALL0840
      CLH(N) = CLH(N) + CLLH(K,N) * CDEL                                TALL0850
      CMH(N) = CMH(N) + CDEL * CHH(K) * (CLMH(K,N) - CLLH(K,N) *        TALL0860
     1     ((HL + XLH(K) - XMC) * CA - HH * SA) / CHH(K))               TALL0870
      CDH(N) = CDH(N) + CLDH(K,N) * CDEL                                TALL0880
  130 CRH(N) = CRH(N) - CLLH(K,N) * YH(K) * CDEL                        TALL0890
      FACTOR = 2.0 / AREA                                               TALL0900
      IF(LIKE .LE. 0) FACTOR = 4.0 / AREA                               TALL0910
      CLH(N) = CLH(N) * FACTOR                                          TALL0920
      CMH(N) = CMH(N) * FACTOR / CREF                                   TALL0930
      CDH(N) = CDH(N) * FACTOR                                          TALL0940
      CRH(N) = CRH(N) * FACTOR                                          TALL0950
      IF(LIKE .EQ. 0) CRH(N) = 0.0                                      TALL0960
C  VERTICAL TAIL                                                        TALL0970
  140 IF(.NOT. VERT) GO TO 170                                          TALL0980
      I = 0                                                             TALL0990
      II = NMAXH                                                        TALL1000
      CYV(N) = 0.0                                                      TALL1010
      CNV(N) = 0.0                                                      TALL1020
      CDV(N) = 0.0                                                      TALL1030
      CRV(N) = 0.0                                                      TALL1040
```

```
      DO 160 K = 1,NROWV                                                TALL1050
      CLYV(K,N) = 0.0                                                   TALL1060
      CLNV(K,N) = 0.0                                                   TALL1070
      CLDV(K,N) = 0.0                                                   TALL1080
      NWK = NV(K)                                                       TALL1090
      DO 150 J = 1,NWK                                                  TALL1100
      I = I + 1                                                         TALL1110
      II = II + 1                                                       TALL1120
      CLYV(K,N) = CLYV(K,N) + CP(II,N) * DXV(I)                         TALL1130
      CLNV(K,N) = CLNV(K,N) - CP(II,N) * DXV(I) * (XGV(I) + 0.25*DXV(I))TALL1140
  150 CLDV(K,N) = CLDV(K,N) + CP(II,N) * WD(II,N)                       TALL1150
      CDEL = CHV(K) * DELTAV(K)                                         TALL1160
      CYV(N) = CYV(N) + CLYV(K,N) * CDEL                                TALL1170
      CNV(N) = CNV(N) + CDEL * CHV(K) * (CLNV(K,N) - CLYV(K,N) *        TALL1180
     1             (VL + XLV(K) - XMC) / CHV(K))                        TALL1190
      CDV(N) = CDV(N) + CLDV(K,N) * CDEL                                TALL1200
  160 CRV(N) = CRV(N) + CLYV(K,N) * ZV(K) * CDEL                        TALL1210
      CYV(N) = CYV(N) /AREA                                             TALL1220
      CNV(N) = CNV(N) / (AREA * CREF)                                   TALL1230
      CDV(N) = CDV(N) / AREA                                            TALL1240
      CRV(N) = CRV(N) / AREA                                            TALL1250
  170 IF((N .LT. NSIDE) .OR. (.NOT. STAB)) GO TO 190                    TALL1260
      IF(ITYPE .EQ. 3) GO TO 180                                        TALL1270
      CLLRV = -CRV(N)                                                   TALL1280
      CNRV = -CNV(N)                                                    TALL1290
      CYRV = -CYV(N)                                                    TALL1300
      GO TO 190                                                         TALL1310
  180 CLLPH = CRH(N)                                                    TALL1320
      CLLPV = -CRV(N)                                                   TALL1330
      CNPV = -CNV(N)                                                    TALL1340
      CYPV = -CYV(N)                                                    TALL1350
  190 CONTINUE                                                          TALL1360
C  PRINT OUT THE CHORDWISE, SPANWISE, AND TOTAL TAIL LOADINGS           TALL1370
      IF(ITYPE .EQ. 3)GO TO 730                                         TALL1380
      IF(PRINT) GO TO 500                                               TALL1390
      NTIMES = NSIDE                                                    TALL1400
      IF(STAB .AND. HOR) NTIMES = NTIMES - 1                            TALL1410
      IF(STAB .AND. VERT) NTIMES = NTIMES - 1                           TALL1420
      CALL HEADER(ILINES)                                               TALL1430
      GO TO ( 200 , 220 ),ITYPE                                         TALL1440
  200 WRITE(6, 210 )                                                    TALL1450
  210 FORMAT(1H0,43X,8(5H*****),2H**/                                   TALL1460
     1          44X,1H*, 9X,22HISOLATED TAIL ANALYSIS, 9X,1H*/          TALL1470
     2          44X,42H*  CHORDWISE LOADING FOR ALL TAIL CASES   */     TALL1480
     3          44X,8(5H*****),2H**/)                                   TALL1490
      GO TO 240                                                         TALL1500
  220 WRITE(6, 230 )                                                    TALL1510
  230 FORMAT(1H0,43X,8(5H*****),2H**/                                   TALL1520
     1          44X,1H*,7X,26HINTERFERENCE TAIL ANALYSIS,7X,1H*/        TALL1530
     2          44X,42H*  CHORDWISE LOADING FOR ALL TAIL CASES   */     TALL1540
     3          44X,8(5H*****),2H**/)                                   TALL1550
  240 DO 380 N = 1,NTIMES                                               TALL1560
      GO TO ( 250 , 310 ),ITYPE                                         TALL1570
  250 GO TO ( 260 , 280 , 300 ),N                                       TALL1580
  260 IF(.NOT. HOR) GO TO 270                                           TALL1590
      WRITE(6, 320 )                                                    TALL1600
      GO TO 380                                                         TALL1610
  270 WRITE(6, 330 )                                                    TALL1620
      GO TO 380                                                         TALL1630
  280 IF(.NOT. VERT) GO TO 290                                          TALL1640
```

```
      WRITE(6, 340 )
      GO TO 380
  290 WRITE(6, 350 )
      GO TO 380
  300 WRITE(6, 360 )
      GO TO 380
  310 WRITE(6, 370 )N,AL(N)
  320 FORMAT(52X,25HCASE 1 IS THE ALPHA  CASE)
  330 FORMAT(52X,25HCASE 1 IS THE  BETA  CASE)
  340 FORMAT(52X,25HCASE 2 IS THE  BETA  CASE)
  350 FORMAT(52X,25HCASE 2 IS THE CAMBER CASE)
  360 FORMAT(52X,25HCASE 3 IS THE CAMBER CASE)
  370 FORMAT(51X,5HCASE ,I1,16H IS FOR ALPHA = ,F5.2)
  380 CONTINUE
  390 IF(.NOT. HOR) GO TO 450
      WRITE(6, 400 )
  400 FORMAT(//,55X,4H****,3(5H*****)/55X,19H* HORIZONTAL TAIL */
     1          55X,4H****,3(5H*****))
      ILINES = ILINES + NTIMES + 11
      II = 0
      DO 440 K = 1,NROWH
      WRITE(6, 410 )K,YH(K),CHH(K),(N,N=1,5)
  410 FORMAT(1H0,39X,7HSECTION,I3,6X,4HY = ,F10.6,5X,8HCHORD = ,F10.6/,
     1          29X,7HELEMENT,6X,3HX/C,6X,5(3X,5HCASE ,I1,3X))
      NHK = NH(K)
      DO 420 I = 1,NHK
      II = II + 1
  420 WRITE(6, 430 )II,XGH(II),(CP(II,N),N = 1,NTIMES)
  430 FORMAT(32X,I2,6X,F10.6,5F12.6)
      ILINES = ILINES + NHK + 3
      IF(ILINES .LE. 55) GO TO 440
      CALL HEADER(ILINES)
      WRITE(6, 400 )
  440 CONTINUE
  450 IF(.NOT. VERT) GO TO 500
      WRITE(6, 460 )
  460 FORMAT(//,56X,2H**,3(5H*****)/56X,17H* VERTICAL TAIL */
     1          56X,2H**,3(5H*****))
      ILINES = ILINES + 5
      II = NMAXH
      III = 0
      DO 490 K = 1,NROWV
      WRITE(6, 470 )K,ZV(K),CHV(K),(N,N=1,5)
  470 FORMAT(1H0,39X,7HSECTION,I3,6X,4HZ = ,F10.6,5X,8HCHORD = ,F10.6/,
     1          29X,7HELEMENT,6X,3HX/C,6X,5(3X,5HCASE ,I1,3X))
      NVK = NV(K)
      DO 480 I = 1,NVK
      II = II + 1
      III = III + 1
  480 WRITE(6, 430 )III,XGV(III),(CP(II,N),N=1,NTIMES)
      ILINES = ILINES + NVK + 3
      IF(ILINES .LE. 55) GO TO 490
      CALL HEADER(ILINES)
      WRITE(6, 460 )
  490 CONTINUE
C  PRINT OUT THE SPANWISE LOADINGS
  500 CALL HEADER(ILINES)
      ILINES = ILINES + 4
      GO TO ( 510 , 530 ),ITYPE
  510 WRITE(6, 520 )
```

```
  520 FORMAT(1H0,38X,10(5H*****),1H*/
     1          39X,1H*,13X,22HISOLATED TAIL ANALYSIS,14X,1H*/
     2          39X,51H*  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES  */
     3          39X,10(5H*****),1H*)
      GO TO 550
  530 WRITE(6, 540 )
  540 FORMAT(1H0,38X,10(5H*****),1H*/
     1          39X,1H*,11X,26HINTERFERENCE TAIL ANALYSIS,12X,1H*/
     2          39X,51H*  SPANWISE AND TOTAL LOADING FOR ALL TAIL CASES  */
     3          39X,10(5H*****),1H*)
  550 IF(.NOT. HOR) GO TO 610
      WRITE(6, 400 )
      WRITE(6, 560 )(N,N=1,5)
  560 FORMAT(1H0,3X,5(9X,5HCASE ,I1,10X)/2X,2HK ,
     1          5(2X,3HCLH,5X,3HCMH,5X,4HCDIH,3X))
      DO 570 K = 1,NROWH
  570 WRITE(6, 580 )K,(CLLH(K,N),CLMH(K,N),CLOH(K,N),N=1,NTIMES)
  580 FORMAT(1X,I2,1X,5(3F8.4,1X))
      WRITE(6, 590 )(CLH(N),CMH(N),CDH(N),N=1,NTIMES)
  590 FORMAT(1H ,2H--,26(5H-----)/4X,5(3F8.4,1X))
      IF(LINE .NE. 0) WRITE(6, 600 )(CRH(N),N=1,NTIMES)
  600 FORMAT(1H0,3X,5(4X,7HCLLH = ,F10.6,4X))
      ILINES = ILINES + NROWH + 12
  610 IF(.NOT. VERT) GO TO 650
      ILINES = ILINES + NROWV + 12
      IF(ILINES .GT. 55) CALL HEADER(ILINES)
      WRITE(6, 460 )
      WRITE(6, 620 )(N,N=1,5)
  620 FORMAT(1H0,3X,5(9X,5HCASE ,I1,10X)/2X,2HK
     1          5(2X,3HCYV,5X,3HCNV,5X,4HCDIV,3X))
      DO 630 K = 1,NROWV
  630 WRITE(6, 580 )K,(CLYV(K,N),CLNV(K,N),CLDV(K,N),N=1,NTIMES)
      WRITE(6, 590 )(CYV(N),CNV(N),CDV(N),N=1,NTIMES)
      WRITE(6, 640 )(CRV(N),N=1,NTIMES)
  640 FORMAT(1H0,3X,5(4X,7HCLLV = ,F10.6,4X))
      IF(ITYPE .NE. 1) GO TO 650
      CYBV = CYV(2)
      CNBV = CNV(2)
      CRBV = CRV(2)
  650 IF(.NOT. STAB) GO TO 710
      ILINES = ILINES + 12
      IF(ILINES .GT. 60) CALL HEADER(ILINES)
      IF(.NOT. HOR) GO TO 680
      WRITE(6, 660 )CLH(NTIMES+1),CMH(NTIMES+1)
  660 FORMAT(1H0,39X,39HHORIZONTAL TAIL CONTRIBUTION TO CLQ  = ,F12.7/
     1          39X,40HHORIZONTAL TAIL CONTRIBUTION TO CMQMC = ,F12.7)
      WRITE(6, 670 )CLLPH
  670 FORMAT(1H0,39X,39HHORIZONTAL TAIL CONTRIBUTION TO CLLP = ,F12.7)
      CLQH = CLH(NTIMES+1)
      CMQH = CMH(NTIMES+1)
  680 IF(.NOT. VERT)GO TO 710
      WRITE(6, 690 )CLLPV,CNPV,CYPV
  690 FORMAT(1H0,41X,37HVERTICAL TAIL CONTRIBUTION TO CLLP = ,F12.7/
     1          42X,37HVERTICAL TAIL CONTRIBUTION TO CNP  = ,F12.7/
     2          42X,37HVERTICAL TAIL CONTRIBUTION TO CYP  = ,F12.7)
      WRITE(6, 700 )CLLRV,CNRV,CYRV
  700 FORMAT(1H0,41X,37HVERTICAL TAIL CONTRIBUTION TO CLLR = ,F12.7/
     1          43X,36HVERTICAL TAIL CONTRIBUTION TO CNR = ,F12.7/
     2          43X,36HVERTICAL TAIL CONTRIBUTION TO CYR = ,F12.7)
  710 WRITE(6, 720 )
```

```
  720 FORMAT(1H0,10X,47H*** NOTE *** ALL TAIL COEFFICIENTS AND STABILIT,
     1          50HY DERIVATIVES ARE NON-DIMENSIONALIZED BY WING AREA,
     2          9H AND CREF)
  730 RETURN
      END
```

```
*DECK STAGE7
      OVERLAY(OVLY,7,0)
      PROGRAM STAGE7
C
C     THIS PROGRAM CONTROLS CALCULATION OF INDUCED VELOCITIES DUE TO
C     CIRCULAR JETS
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMK, NW(40),NJ(40),IW(40),IJ(40)
      COMMON/GEFECT/GHITE(10),IGAND,IGND,NHITE
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XGC(400),XI(400),DEL(400),
     1              D(40),KK(400),ITYPE(400)
      COMMON /JET3/ E1,E2,E3,CD,M
      COMMON /JET4/ HS,NMAX
      COMMON /JET5/ UJ0,U,D0,THETA,H,ZMAX1,TETAMX,XOCL,YOCL,ZOCL,DIHED,
     1              JPRINT,JNUMB,JETREP,ZGND,NSIGMA
      COMMON /JET6/ MB(200),MU(200),PD(200),QD(200),RD(200),SD(200),
     1 ZD(200),SMINOR(200),DS(200),INJET(100),SQUI,ET,ZET,INJETP,NINJET,
     2 SIGMA(200),HSPRME
      COMMON /JET7/ VXT(600),VYT(600),VZT(600)
      COMMON/JET8/P(200),PP(200),PDP(200),MP(200),MPP(200),CP(200),
     1                   Q(200),QP(200),PDQ(200),MQ(200),MQP(200),CQ(200),
     2                   R(200),RP(200),PDR(200),MR(200),MRP(200),CR(200),
     3                   S(200),SP(200),PDS(200),MS(200),MSP(200),CS(200),
     4                   ZS(200),TETA(200),CDSAVE(200),E2SAVE(200)
      COMMON /JET10/ CIRCUM(200)
      COMMON/JET11/NJETS,NCOLOD,NFUSZ,NFUSY,NTOTAL,ISYM,FMACH,
     1              XXX(600),YYY(600),ZZZ(600)
      COMMON/JET12/UJ(10),DDO(10),THTA(10),HH(10),ZMX(10),XO(10),YO(10),
     1              ZO(10),DIHE(10),ISY(10),JETRE(10),NJET
      COMMON/JET13/CJ(10),THO(10)
      COMMON/FFLD3/SQUIG(200),ETA(200),ZETA(200),NPOINT,AL(10),
     1              IFFLD,TCR,AINF(40,10),XLR(40),XLL(40),
     2              XTR(40),XTL(40),CHR(40),CHL(40)
      COMMON /FFLD4/ VXJET(600),VYJET(600),VZJET(600)
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HHH,VL,VH,
     1              NMAXH,NMAXV,NTOT,   NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON/FUSLGI/PI,NSEG,NSEGI,NF(40),ISECT,ICAMF,XB(41),ZCL(41),
     1               RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2               V(40,7),W(40,7)
      REAL M,MB,MU,MP,MQ,MR,MS,MPP,MQP,MRP,MSP
      LOGICAL PRINT
C
C  INITIALIZE THE CONTROL FLAGS AND PARAMETERS
      NTOTAL = NEWMAX + NTOT + NSEG + NPOINT
      IWRITE = 2*(NMX+NJT) + 3*NTOT
      INJETP = 1
      JPRINT = IPRINT
      PRINT = (JPRINT .GE. 0)
```

```
      NSIGMA = 0
      NJETS = NJET
      NALPHA = 5
      DO 20 NA = 1,NALPHA
   20 AL(NA) = 5.0*(NA-1)
      U = 1.0
      FMACH = 0.0
      TETAMX = 89.0
      ZGND = 0.0
      IF(IGND .GT. 0) ZGND = GHITE(NHITE)
C  CYCLE THE ANGLE OF ATTACK
      DO 240 NA = 1,NALPHA
      ALPHA = AL(NA)
      SA = SIN(ALPHA/57.295779)
      CA = COS(ALPHA/57.295779)
      DO 30  K = 1,NTOTAL
      VXJET(K) = 0.0
      VYJET(K) = 0.0
      VZJET(K) = 0.0
   30 CONTINUE
C  DEFINE THE FLOW FIELD COORDINATES
C  WING DOWNWASH CONTROL POINTS
   40 DO 50  I = 1,NEWMAX
      K = KK(I)
      XXX(I) = XI(I) + 0.50*DEL(I)*CHORD(K)
      YYY(I) = Y(K)
      ZZZ(I) = 0.0
   50 CONTINUE
C  HORIZONTAL AND VERTICAL TAIL AND OFF-BODY POINTS
      IF(NTOTAL .EQ. NEWMAX) GO TO 70
      J = 0
      ISTART = NEWMAX + 1
      DO 60  I = ISTART,NTOTAL
      J = J + 1
      XXX(I) = SQUIG(J)*CA + ZETA(J)*SA
      YYY(I) = ETA(J)
      ZZZ(I) = ZETA(J)*CA - SQUIG(J)*SA
   60 CONTINUE
C  CYCLE ALL THE JETS
   70 DO 170 J = 1,NJETS
      JNUMB = J
      ISYM = ISY(J)
      UJ0 = UJ(J)
      D0 = DDO(J)
      THETA = THTA(J) - ALPHA
      THET = THO(J) + ALPHA
      H = HH(J)
      ZMAX1 = ZMX(J)
      DIHED = DIHE(J)
      JETREP = JETRE(J)
      XOCL = XO(J)*CA + ZO(J)*SA
      YOCL = YO(J)
      ZOCL = ZO(J)*CA - XO(J)*SA
      IF((YO(J) .EQ. 0.0) .AND. (DIHED .EQ. 0.0)) ISYM = 1
C  CALCULATE JET CENTERLINE
   80 IF(JETREP .EQ. 0) CALL OVERLAY(4HOVLY,7,1)
C  PRINT THE JET GEOMETRY
      IF(PRINT) GO TO 150
      CALL HEADER(ILINES)
      WRITE(6, 90 ) JNUMB,ALPHA,CJ(J),D0,THET ,DIHED,ISYM,
```

```
     1                   XOCL,YOCL,ZOCL,H
   90 FORMAT(1H0, 37X,11(4H****)         / 38X, 15H*  CIRCULAR JET,        STG71120
     1        29H GEOMETRIC CHARACTERISTICS   * / 38X,1H*,2X,               STG71130
     2        10HJET NUMBER,I3,5X,7HALPHA =,F5.1,8H DEGREES, 2X,1H* /       STG71140
     3        38X,11(4H****) //19X,2HCJ ,9X,6HD0/B/2,5X,6HTHETA0,5X,        STG71150
     4        5HDIHED,7X,4HISYM,8X,6HX0/B/2,5X,6HY0/B/2,                    STG71160
     5       5X,6HZ0/B/2,5X,4HDELZ/15X,4F11.6,6X,I1,6X,4F11.6//3X,1HN,4X,STG71170
     6        5HX/B/2, 6X,5HY/B/2, 6X,5HZ/B/2, 6X, 4HUJ/U,    7X,5HTHETA,  STG71180
     7  14X,1HN,4X,5HX/B/2,6X,5HY/B/2,6X,5HZ/B/2,6X,4HUJ/U,   7X,5HTHETA)STG71190
      ILINES = ILINES + 10                                                  STG71200
      SINDIH = SIN(DIHED/57.295779)                                         STG71210
      COSDIH = COS(DIHED/57.295779)                                         STG71220
      DO 140 I = 1,NMAX,2                                                   STG71230
      I2 = I + 1                                                            STG71240
      ILINES = ILINES + 1                                                   STG71250
      XCL1 = XOCL + SD(I)                                                   STG71260
      YCL1 = YOCL + ZD(I)*SINDIH                                            STG71270
      ZCL1 = ZOCL - ZD(I)*COSDIH                                            STG71280
      THE1 = 90.0 - TETA(I)                                                 STG71290
      IF(I .EQ. NMAX)  GO TO 100                                            STG71300
      XCL2 = XOCL + SD(I2)                                                  STG71310
      YCL2 = YOCL + ZD(I2)*SINDIH                                           STG71320
      ZCL2 = ZOCL - ZD(I2)*COSDIH                                           STG71330
      THE2 = 90.0 - TETA(I2)                                                STG71340
  100 IF(ILINES .LT. 61) GO TO 120                                          STG71350
      WRITE(6, 110 )                                                        STG71360
  110 FORMAT(1H1, 1X,1HN,5X,5HX/B/2,6X,5HY/B/2,                             STG71370
     1        6X,5HZ/B/2,6X,6HUJ/UJ0,5X,5HTHETA, 14X,1HN,4X,5HX/B/2,        STG71380
     2        6X,5HY/B/2,6X,5HZ/B/2,6X,6HUJ/UJ0,5X,5HTHETA)                 STG71390
      ILINES = 2                                                            STG71400
  120 WRITE(6, 130 ) I,XCL1,YCL1,ZCL1,PD(I),THE1,                           STG71410
     1                I2,XCL2,YCL2,ZCL2,PD(I2),THE2                         STG71420
  130 FORMAT(1H ,I3,5F11.6,10X,I3,5F11.6)                                   STG71430
  140 CONTINUE                                                              STG71440
C  CALCULATE INDUCED VELOCITIES                                             STG71450
  150 CALL OVERLAY(4HOVLY,7,2)                                              STG71460
      IF(IA .EQ. 2) GO TO 250                                               STG71470
      DO 160 I = 1,NTOTAL                                                   STG71480
      VXJET(I) = VXJET(I) + VXT(I)                                          STG71490
      VYJET(I) = VYJET(I) + VYT(I)                                          STG71500
      VZJET(I) = VZJET(I) + VZT(I)                                          STG71510
  160 CONTINUE                                                              STG71520
  170 CONTINUE                                                              STG71530
C  PRINT THE INDUCED VELOCITIES                                             STG71540
      IF(PRINT) GO TO 230                                                   STG71550
      CALL HEADER(ILINES)                                                   STG71560
      WRITE(6, 180 ) ALPHA                                                  STG71570
  180 FORMAT(1H0,34X,12(4H****),         /35X,24H*  FLOW FIELD VELOCITIES, STG71580
     1        24H DUE TO CIRCULAR JETS   * /                                STG71590
     2        35X,1H*, 13X,  7HALPHA =, F5.1,  8H DEGREES , 13X, 1H* /      STG71600
     3        35X, 12(4H****),        // 3X,1HN,4X,5HX/B/2,5X,5HY/B/2,5X,   STG71610
     4        5HZ/B/2,5X,2HVX,8X,2HVY,8X,2HVZ,10X,1HN,4X,5HX/B/2,           STG71620
     5        5X,5HY/B/2,5X,5HZ/B/2,5X,2HVX,8X,2HVY,8X,2HVZ)                STG71630
      ILINES = ILINES + 7                                                   STG71640
      DO 220 N=1,NTOTAL,2                                                   STG71650
      N2 = N + 1                                                            STG71660
      ILINES = ILINES + 1                                                   STG71670
      IF(ILINES .LT. 61) GO TO 200                                          STG71680
      WRITE(6, 190 )                                                        STG71690
      ILINES = 2                                                            STG71700
                                                                            STG71710
```

```
  190 FORMAT(1H1, 1X,1HN, 5X, 5HX/B/2, 5X, 5HY/B/2, 5X, 5HZ/B/2,            STG71720
     1        5X, 2HVX,  8X, 2HVY,  8X, 2HVZ, 9X,1HN,5X,5HX/B/2,5X,         STG71730
     2        5HY/B/2,5X,5HZ/B/2 5X,2HVX,8X,2HVY,8X,2HVZ)                   STG71740
  200 WRITE(6, 210 ) N,XXX(N),YYY(N),ZZZ(N),VXJET(N),VYJET(N),VZJET(N),     STG71750
     1        N2,XXX(N2),YYY(N2),ZZZ(N2),VXJET(N2),VYJET(N2),VZJET(N2)      STG71760
  210 FORMAT(1H ,I3,6F10.5,4X,I3,6F10.5)                                    STG71770
  220 CONTINUE                                                              STG71780
C  STORE THE FLOW FIELD VELOCITIES FOR THIS ALPHA ON UNIT 8                 STG71790
  230 IWRITE = IWRITE + 1                                                   STG71800
      CALL WRITMS(8,VXJET,600,IWRITE)                                       STG71810
      IWRITE = IWRITE + 1                                                   STG71820
      CALL WRITMS(8,VYJET,600,IWRITE)                                       STG71830
      IWRITE = IWRITE + 1                                                   STG71840
      CALL WRITMS(8,VZJET,600,IWRITE)                                       STG71850
  240 CONTINUE                                                              STG71860
  250 CONTINUE                                                              STG71870
      END                                                                   STG71880
```

```
*DECK CNTRLN                                                                CNTR0010
      OVERLAY(OVLY,7,1)                                                     CNTR0020
      PROGRAM CNTRLN                                                        CNTR0030
C                                                                           CNTR0040
C      SUBROUTINE CENTERLINE COMPUTES THE JET CENTERLINE COORDINATES ANDCNTR0050
C      THE SINK AND DOUBLET STRENGTHS ASSOCIATED WITH THE EXHAUSTING JETCNTR0060
C                                                                           CNTR0070
      COMMON /JET1/ A0,  A1,  A2,BM1,  B0,  B1,  B2,E5                      CNTR0080
      COMMON /JET2/ AA0,AA1,AA2,BB0,BB1,BB2,BB3,EB5                         CNTR0090
      COMMON /JET3/ E1,E2,E3,CD,M                                           CNTR0100
      COMMON /JET4/ HS,NMAX                                                 CNTR0110
      COMMON /JET5/ UJ0,U,D0,THETA,H,ZMAX1,TETAMX,XOCL,YOCL,ZOCL,DIHED,     CNTR0120
     1             JPRINT,JNUMB,JETREP,ZGND,NSIGMA                          CNTR0130
      COMMON /JET6/ MB(200),MU(200),PD(200),QD(200),RD(200),SD(200),        CNTR0140
     1 ZD(200),SMINOR(200),DS(200),INJET(100),XI,ETA,ZETA,INJETP,NINJET,CNTR0150
     2 SIGMA(200),HSPRME                                                    CNTR0160
      COMMON/JET8/P(200),PP(200),PDP(200),MP(200),MPP(200),DP(200),         CNTR0170
     1                 Q(200),QP(200),PDQ(200),MQ(200),MQP(200),CQ(200),CNTR0180
     2                 R(200),RP(200),PDR(200),MR(200),MRP(200),CR(200),CNTR0190
     3                 S(200),SP(200),PDS(200),MS(200),MSP(200),CS(200),CNTR0200
     4                 ZS(200),TETA(200),CDSAVE(200),E2SAVE(200)            CNTR0210
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                                    CNTR0220
      COMMON /JET10/ CIRCUM(200)                                            CNTR0230
      COMMON/JET11/NJETS,NCOLOQ,NFUSZ,NFUSY,NTOTAL,ISYM,FMACH,              CNTR0240
     1           XXX(600),YYY(600),ZZZ(600)                                 CNTR0250
       REAL M,MB,MU,MP,MQ,MR,MS,MPP,MQP,MRP,MSP                             CNTR0260
C                                                                           CNTR0270
       PI = 3.14159                                                         CNTR0280
       M = UJ0/U                                                            CNTR0290
       COSTO = COS( THETA *PI/180.  )                                       CNTR0300
       HSPRME = 0.3 *M/COSTO                                                CNTR0310
       IF ( THETA  .LT.  0.0 )  HSPRME = 0.3 *M*COSTO                       CNTR0320
       ZREG1 = HSPRME *D0                                                   CNTR0330
C      INITIALIZE C/L STARTING POINT (X0,Y0,Z0) AT (0.,0.,0.) IN C/L COOCNTR0340
       X0 = 0.                                                              CNTR0350
       Y0 = 0.                                                              CNTR0360
       Z0 = 0.                                                              CNTR0370
       DXDZ0 = TAN ( 3.1416*THETA/180.0  )                                  CNTR0380
C      INITIALIZE STABILITY PARAMETERS A1,A2,AA1,AA2 FOR THE ADAMS          CNTR0390
```

```
C      PREDICTOR-CORRECTOR ''2/3 - 1/3'' RULE.                              CNTR0400
C      (REF. HAMMING, DAC.LIB. QA 246 H18, P.205,15.8,15.9)                 CNTR0410
       A1  = 2./3.                                                          CNTR0420
       A2  = 1./3.                                                          CNTR0430
       AA1 = A1                                                             CNTR0440
       AA2 = A2                                                             CNTR0450
C      INITIALIZE  ENTRAINMENT   COEFFICIENTS E1,E2,E3                      CNTR0460
C      INITIALIZE CROSSFLOW DRAG COEFFICIENT  CD                            CNTR0470
       E1 = 0.45                                                            CNTR0480
       E2 = 0.08                                                            CNTR0490
       E3 = 30.0                                                            CNTR0500
       CD = 1.80                                                            CNTR0510
C      INITIALIZE JET VELOCITY P(1) = UJ/UJ0 = 1.0                          CNTR0520
C      INITIALIZE JET DIAMETER  Q(1) = D/D0 = 1.0                           CNTR0530
       P(1) = 1.0                                                           CNTR0540
       Q(1) = 1.0                                                           CNTR0550
       R(1) = DXDZ0                                                         CNTR0560
       S(1) = X0/D0                                                         CNTR0570
       ZS(1)= Z0/D0                                                         CNTR0580
       TETA(1) = THETA                                                      CNTR0590
       A0  = 1.-A1 -A2                                                      CNTR0600
       BM1 = (9. -A1) /24.                                                  CNTR0610
       B0  = (19. +13. *A1 +8. *A2) /24.                                    CNTR0620
       B1  = (-5. +13. *A1 +32. *A2) /24.                                   CNTR0630
       B2  = (1. -A1 +8. *A2) /24.                                          CNTR0640
       E5  = (-19. +11 *A1 -8. *A2) /6.                                     CNTR0650
       AA0 = A0                                                             CNTR0660
       BB0 = (55. +9.*AA1 + 8. *AA2) /24.                                   CNTR0670
       BB1 = (-59. +19. *AA1 +32. *AA2) /24.                                CNTR0680
       BB2 = (37. -5. *AA1 +8. *AA2) /24.                                   CNTR0690
       BB3 = (-9. +AA1) /24.                                                CNTR0700
       EB5 = (251. -19. *AA1 -8. *AA2) /6.                                  CNTR0710
       HS = H /D0                                                           CNTR0720
       NMAX = (ZMAX1-Z0)/H  + 1.005                                         CNTR0730
       IF ( NMAX  .GT.  199 )  NMAX = 199                                   CNTR0740
C      CALL SUBROUTINE 'RKUTTA' TO GET STARTING VALUES                      CNTR0750
C      FOR ADAMS PREDICTOR-CORRECTOR METHOD OF INTEGRATION                  CNTR0760
   20  CALL RKUTTA                                                          CNTR0770
C   THE ADAMS PREDICTOR-CORRECTOR TAKES CONTROL FROM RUNGE-KUTTA AT Y(N*CNTR0780
C   THE FIRST PREDICTOR-CORRECTOR BACKWARDS VALUES, FOR Y(4), ARE INITIACNTR0790
C      (REF.  HAMMING  P.206 SECTION 15.10)                                 CNTR0800
   30  PDP(4) = 0.                                                          CNTR0810
       PDQ(4) = 0.                                                          CNTR0820
       PDR(4) = 0.                                                          CNTR0830
       PDS(4) = 0.                                                          CNTR0840
       CP(4)  = 0.                                                          CNTR0850
       CQ(4)  = 0.                                                          CNTR0860
       CR(4)  = 0.                                                          CNTR0870
       CS(4)  = 0.                                                          CNTR0880
   40  DO 50  J = 1,3                                                       CNTR0890
       N = J - 1                                                            CNTR0900
       CALL MBMU                                                            CNTR0910
       N = J                                                                CNTR0920
   50 CONTINUE                                                              CNTR0930
       CALL MBMU                                                            CNTR0940
       N = 4                                                                CNTR0950
   60 CONTINUE                                                              CNTR0960
       DO 80    N =   4,NMAX                                                CNTR0970
       MFLAG = 1                                                            CNTR0980
       NPM = N + MFLAG                                                      CNTR0990
```

```
       ZS(NPM) = ZS(N) + HS                                                 CNTR1000
C      SOLVE FOR  P(N+1) Q(N+1) R(N+1) S(N+1)                               CNTR1010
       L = N                                                                CNTR1020
       K = N + 1                                                            CNTR1030
       NFUNCT = 1                                                           CNTR1040
       CALL ADAMS                                                           CNTR1050
       CALL MOTION                                                          CNTR1060
       CALL ADAMS1                                                          CNTR1070
       NFUNCT = 2                                                           CNTR1080
       CALL ADAMS                                                           CNTR1090
       CALL MOTION                                                          CNTR1100
       CALL ADAMS1                                                          CNTR1110
       NFUNCT = 3                                                           CNTR1120
       CALL ADAMS                                                           CNTR1130
      MR(N) = MR(N+1)                                                       CNTR1140
       CALL MOTION                                                          CNTR1150
       CALL ADAMS1                                                          CNTR1160
       NFUNCT = 4                                                           CNTR1170
       CALL ADAMS                                                           CNTR1180
       CALL MOTION                                                          CNTR1190
       CALL ADAMS1                                                          CNTR1200
       TETA (NPM) = 180./3.1416 * ATAN( R(NPM) )                            CNTR1210
C      SUBROUTINE 'MOTION' SOLVES THE EQUATIONS OF MOTION FOR REGIONS 1 CNTR1220
      MFLAG = 0                                                             CNTR1230
       NFUNCT = 5                                                           CNTR1240
       CALL MOTION                                                          CNTR1250
       I = N+1                                                              CNTR1260
       CALL MBMU                                                            CNTR1270
   70  IMAX = I                                                             CNTR1280
       IF (TETAMX  .GT.  0.0  .AND.  TETA(NPM)   GE.  TETAMX)  GO TO 90 CNTR1290
   80  CONTINUE                                                             CNTR1300
       NMAX = IMAX                                                          CNTR1310
       GO TO 100                                                            CNTR1320
   90  CONTINUE                                                             CNTR1330
       NMAX = IMAX                                                          CNTR1340
  100  CONTINUE                                                             CNTR1350
C      ADJUST MB (THE LINE SINK STRENGTH/UNIT JET DIA) BY                   CNTR1360
C      SIGMA (THE POINT SOURCE STRENGTHPUNIT JET DIA)                       CNTR1370
C      NOTE--- THE VARIABLE MB IS REALLY A MB NET FROM THIS POINT ON IN CNTR1380
       DO 110 I = 1,NMAX                                                    CNTR1390
  110  MB(I) = MB(I) - SIGMA(I)                                             CNTR1400
      CONTINUE                                                              CNTR1410
      END                                                                   CNTR1420
```

```
*DECK MBMU                                                                  MBMU0010
       SUBROUTINE MBMU                                                      MBMU0020
C                                                                           MBMU0030
C      SUBROUTINE MBMU COMPUTES C/L SOURCE AND DOUBLET STRENGTHS            MBMU0040
C                                                                           MBMU0050
      COMMON /JET3/ E1,E2,E3,CD,M                                           MBMU0060
      COMMON /JET4/ HS,NMAX                                                 MBMU0070
      COMMON /JET5/ UJ0,U,D0,THETA,H,ZMAX1,TETAMX,XOCL,YOCL,ZOCL,DIHED,     MBMU0080
     1              JPRINT,JNUMB,JETREP,ZGND,NSIGMA                         MBMU0090
      COMMON /JET6/ MB(200),MU(200),PD(200),QD(200),RD(200),SD(200),        MBMU0100
     1 ZD(200),SMINOR(200),DS(200),INJET(100),XI,ETA,ZETA,INJETP,NINJET,MBMU0110
     2 SIGMA(200),HSPRME                                                    MBMU0120
      COMMON/JET8/P(200),PP(200),PDP(200),MP(200),MPP(200),CP(200),         MBMU0130
```

```
     1                  Q(200),QP(200),PDQ(200),MQ(200),MQP(200),CQ(200),MBMU0140
     2                  R(200),RP(200),PDR(200),MR(200),MRP(200),CR(200),MBMU0150
     3                  S(200),SP(200),PDS(200),MS(200),MSP(200),CS(200),MBMU0160
     4                  ZS(200),TETA(200),COSAVE(200),E2SAVE(200)        MBMU0170
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                                MBMU0180
      COMMON /JET10/ CIRCUM(200)                                        MBMU0190
C                                                                       MBMU0200
       REAL M,MB,MU,MP,MQ,MR,MS,MPP,MQP,MRP,MSP                         MBMU0210
       PI = 3.1416                                                      MBMU0220
      I = N + 1                                                         MBMU0230
       SIGMA(I) = 0.                                                    MBMU0240
C      DIMENSIONALIZE VARIABLES P,Q,R,S,ZS                              MBMU0250
       PD(I) = P(I)*UJO                                                 MBMU0260
       QD(I) = Q(I)*DO                                                  MBMU0270
       RD(I) = R(I)                                                     MBMU0280
       SD(I) = S(I)*DO                                                  MBMU0290
       ZD(I) = ZS(I)*DO                                                 MBMU0300
       ZDI = ZD(I)                                                      MBMU0310
       IF ( ZGND  .EQ.  0.0 )  GO TO 20                                 MBMU0320
       IF (ZDI .GT. ZGND) ZD(I) = ZGND                                  MBMU0330
   20  SINA = RD(I) /SQRT(1.+RD(I)**2 )                                 MBMU0340
       COSA =  1.   /SQRT(1.+RD(I)**2 )                                 MBMU0350
       DELTAS = HS*DO/COSA                                              MBMU0360
       DS(I) = DELTAS                                                   MBMU0370
C      BRANCH AND COMPUTE C/L SINK STRENGTHS (MB), SOURCE STRENGTHS (SIGMBMU0380
C      DOUBLET STRENGTHS (MU)                                           MBMU0390
       IF ( ZS(I)  .GT.  HSPRME )  GO TO 30                             MBMU0400
C      *****  REGION  I  *****************                              MBMU0410
       VARB = 1. -0.375 *( ZS(I) /HSPRME )                              MBMU0420
       MU(I) = (PI/4. )*U*QD(I)**2*COSA*DELTAS                          MBMU0430
       MU(I) = MU(I) *VARB                                              MBMU0440
       SMINOR(I) = QD(I) *(1. -2.5*ZS(I)*(U/UJO) ) /2.0                 MBMU0450
       CIRCUM(I) = PI*QD(I)*SQRT((1.+(1.+0.75 *ZS(I)/HSPRME)**2 ) /2.0 ) MBMU0460
       GO TO 40                                                         MBMU0470
C      *****  REGION  II *****************                              MBMU0480
   30  CONTINUE                                                         MBMU0490
       VARB = 0.625                                                     MBMU0500
       MU(I) =( PI*U*QD(I)**2*COSA /4.0  ) * DELTAS                     MBMU0510
       MU(I) = MU(I) *VARB                                              MBMU0520
       CIRCUM(I) = 2.24 *QD(I)                                          MBMU0530
C      SMINOR(I) AND DS(I) ARE ARRAYS GOING TO ROUTINE PTEST.           MBMU0540
       SMINOR(I) = QD(I) /8.0                                           MBMU0550
   40  IF ( NSIGMA  .GT.  0 )  GO TO 50                                 MBMU0560
C      NSIGMA=0,COMPUTE C/L SIGMA DISTRIBUTION AND ADJUST THE LINE SINK MBMU0570
C      DISTRIBUTION (MB) BY SIGMA IN SUBROUT CNTRLN.   SIGMA IS A FUNCTIMBMU0580
C      THE C/L RADIUS OF CURVATURE (RC).                                MBMU0590
       RC = ( (1. + R(I)**2 )**1.5 /RP(I) )                             MBMU0600
       CURVA = 1. /RC                                                   MBMU0610
       SIGMA(I) = 3. *H *CURVA / COSA                                   MBMU0620
   50  E = E2SAVE(I)*(PD(I)-U*SINA) /(1.+E3*U*COSA/PD(I) )              MBMU0630
       MB(I) = (E1*U*QD(I)*COSA + E*CIRCUM(I) ) *DELTAS/QD(I)           MBMU0640
       RETURN                                                           MBMU0650
      END                                                               MBMU0660


*DECK RKUTTA                                                            RKUT0010
       SUBROUTINE RKUTTA                                                RKUT0020
C                                                                       RKUT0030
```

```
C      SUBROUTINE 'RKUTTA' COMPUTES THE STARTING VALUES FOR THE         RKUT0040
C      ADAMS PREDICTOR-CORRECTOR METHOD IF INTEGRATION                  RKUT0050
C      ( REF. HAMMING  P.212 SECTION 16.2 )                             RKUT0060
C                                                                       RKUT0070
      COMMON /JET4/ HS,NMAX                                             RKUT0080
      COMMON /JET5/ UJO,U,DO,THETA,H,ZMAX1,TETAMX,XOCL,YOCL,ZOCL,DIHED, RKUT0090
     1              JPRINT,JNUMB,JETREP,ZGND,NSIGMA                     RKUT0100
      COMMON/JET8/P(200),PP(200),PDP(200),MP(200),MPP(200),CP(200),     RKUT0110
     1                 Q(200),QP(200),PDQ(200),MQ(200),MQP(200),CQ(200),RKUT0120
     2                 R(200),RP(200),PDR(200),MR(200),MRP(200),CR(200),RKUT0130
     3                 S(200),SP(200),PDS(200),MS(200),MSP(200),CS(200),RKUT0140
     4                 ZS(200),TETA(200),COSAVE(200),E2SAVE(200)        RKUT0150
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                                RKUT0160
       DIMENSION PT(4),PPT(4),Y(4),                                     RKUT0170
     1           QT(4),QPT(4),F(4),                                     RKUT0180
     2           RT(4),RPT(4),G(16),                                    RKUT0190
     3           ST(4),SPT(4),DY(4),ZST(4)                              RKUT0200
       REAL  MP,MQ,MR,MS,MPP,MQP,MRP,MSP                                RKUT0210
C                                                                       RKUT0220
      CONST = 180./3.141592                                             RKUT0230
       DX = HS                                                          RKUT0240
       NK = 4                                                           RKUT0250
       NKUTTA = 3                                                       RKUT0260
C      CALL SUBROUTINE MOTION TO GET INITIAL DERIVATIVE VALUES          RKUT0270
       NFUNCT = 5                                                       RKUT0280
       MFLAG = 0                                                        RKUT0290
       N = 1                                                            RKUT0300
       K = 1                                                            RKUT0310
       L = 0                                                            RKUT0320
       CALL MOTION                                                      RKUT0330
       PT(1) = P(1)                                                     RKUT0340
       QT(1) = Q(1)                                                     RKUT0350
       RT(1) = R(1)                                                     RKUT0360
       ST(1) = S(1)                                                     RKUT0370
       ZST(1) = ZS(1)                                                   RKUT0380
      PPT(1) = PP(1)                                                    RKUT0390
      QPT(1) = QP(1)                                                    RKUT0400
      RPT(1) = RP(1)                                                    RKUT0410
      SPT(1) = SP(1)                                                    RKUT0420
       Y(1) = P(1)                                                      RKUT0430
       Y(2) = Q(1)                                                      RKUT0440
       Y(3) = R(1)                                                      RKUT0450
       Y(4) = S(1)                                                      RKUT0460
       X = ZS(1)                                                        RKUT0470
C      BEGIN LOOP THAT INCREMENTS ZS AND ASIGNS THE FINAL OUTPUT VALUES RKUT0480
       DO 30  L = 1,NKUTTA                                              RKUT0490
C      BEGIN THE LOOP THAT ACCESSES THE ACTUAL RUNGE-KUTTA ROUTINE 'DRK1RKUT0500
C      SOLVE FOR THE NEXT POINT.                                        RKUT0510
C      THE INTERNAL COUNTER IN 'DRK1' IS INITIALIZED AT 0               RKUT0520
       IS = 0                                                           RKUT0530
       DO 20 N = 1,NK                                                   RKUT0540
       K = N                                                            RKUT0550
       P(N) = Y(1)                                                      RKUT0560
       Q(N) = Y(2)                                                      RKUT0570
       R(N) = Y(3)                                                      RKUT0580
       S(N) = Y(4)                                                      RKUT0590
       ZS(N) = X                                                        RKUT0600
       MFLAG = 0                                                        RKUT0610
       NFUNCT = 5                                                       RKUT0620
       CALL MOTION                                                      RKUT0630
```

```
       DY(1) = PP(N)                                                    RKUT0640
       DY(2) = QP(N)                                                    RKUT0650
       DY(3) = RP(N)                                                    RKUT0660
       DY(4) = SP(N)                                                    RKUT0670
   20  CALL DRK1 ( X,Y,DY,DX,NK,F,G,IS )                                RKUT0680
       PT(L+1) = Y(1)                                                   RKUT0690
       QT(L+1) = Y(2)                                                   RKUT0700
       RT(L+1) = Y(3)                                                   RKUT0710
       ST(L+1) = Y(4)                                                   RKUT0720
       ZST(L+1) = X                                                     RKUT0730
      PPT(L+1) = PP(NK)                                                 RKUT0740
      QPT(L+1) = QP(NK)                                                 RKUT0750
      RPT(L+1) = RP(NK)                                                 RKUT0760
      SPT(L+1) = SP(NK)                                                 RKUT0770
   30  CONTINUE                                                         RKUT0780
      NKP1 = NKUTTA + 1                                                 RKUT0790
       DO 40  L = 1,NKP1                                                RKUT0800
       P(L) = PT(L)                                                     RKUT0810
       Q(L) = QT(L)                                                     RKUT0820
       R(L) = RT(L)                                                     RKUT0830
       S(L) = ST(L)                                                     RKUT0840
       ZS(L) = ZST(L)                                                   RKUT0850
      PP(L) = PPT(L)                                                    RKUT0860
      QP(L) = QPT(L)                                                    RKUT0870
      RP(L) = RPT(L)                                                    RKUT0880
      SP(L) = SPT(L)                                                    RKUT0890
   40 TETA(L) = CONST*ATAN(R(L))                                        RKUT0900
       RETURN                                                           RKUT0910
      END                                                               RKUT0920


*DECK PTEST                                                             PTST0010
       SUBROUTINE PTEST                                                 PTST0020
C                                                                       PTST0030
C      ROUTINE TESTS FOR VELOCITY TEST POINTS FALLING INSIDE THE JET ENVPTST0040
C      K  = VELOCITY TEST POINT COUNTER                                 PTST0050
C      N  = JET CENTERLINE ELEMENT COUNTER                              PTST0060
C      A  = ELLIPTICAL JET CROSS-SECTION MAJOR AXIS                     PTST0070
C      B  = ELLIPTICAL JET CROSS-SECTION MINOR AXIS                     PTST0080
C                                                                       PTST0090
      COMMON /JET6/ MB(200),MU(200),PD(200),QD(200),RD(200),SD(200),    PTST0100
     1 ZD(200),SMINOR(200),DS(200),INJET(100),XI,ETA,ZETA,INJETP,NINJET,PTST0110
     2 SIGMA(200),HSPRME                                                PTST0120
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                                PTST0130
      REAL MB,MU                                                        PTST0140
C                                                                       PTST0150
       A = QD(N) /2.0                                                   PTST0160
       B = SMINOR(N)                                                    PTST0170
       IF ((ABS(XI)-DS(N)/2.) .LE. 0.0)  GO TO 20                       PTST0180
       RETURN                                                           PTST0190
   20  RPTSQ =  ETA**2 + ZETA**2                                        PTST0200
       SINSQ = ZETA**2 /RPTSQ                                           PTST0210
       COSSQ =  ETA**2 /RPTSQ                                           PTST0220
       RELLSQ = (A**2 *B**2 ) /(A**2 *SINSQ + B**2 *COSSQ)              PTST0230
       IF (RPTSQ  .LT.  RELLSQ)  GO TO 30                               PTST0240
       RETURN                                                           PTST0250
   30  NINJET = NINJET + 1                                              PTST0260
       IF ( NINJET  .GT.  100 )  RETURN                                 PTST0270
```

```
        INJET(NINJET) = K                                               PTST0280
        RETURN                                                          PTST0290
       END                                                              PTST0300


*DECK ADAMS                                                             ADMS0010
       SUBROUTINE ADAMS                                                 ADMS0020
C                                                                       ADMS0030
C      SUBROUTINE 'ADAMS'  SOLVES THE ADAMS PREDICTOR-CORRECTOR EQUATIONADMS0040
C      FOR REFERENCE TO THE ADAMS PREDICTOR-CORRECTOR METHOD SEE---     ADMS0050
C      NUMERICAL METHODS FOR SCIENTISTS AND ENGINEERS                   ADMS0060
C      R.W.HAMMING  (DAC. LIB. QA 246 H18  C-35578 NO.2 )               ADMS0070
C                                                                       ADMS0080
      COMMON/SPIRIT/ NEWMAX,NEWCMD,NOALFA,LOGIC,IR,ISTAB                ADMS0090
      COMMON /JET1/ A0, A1, A2,BM1, B0, B1, B2,E5                       ADMS0100
      COMMON /JET2/ AA0,AA1,AA2,BB0,BB1,BB2,BB3,EB5                     ADMS0110
      COMMON /JET4/ HS,NMAX                                             ADMS0120
      COMMON/JET8/P(200),PP(200),PDP(200),MP(200),MPP(200),CP(200),     ADMS0130
     1                 Q(200),QP(200),PDQ(200),MQ(200),MQP(200),CQ(200),ADMS0140
     2                 R(200),RP(200),PDR(200),MR(200),MRP(200),CR(200),ADMS0150
     3                 S(200),SP(200),PDS(200),MS(200),MSP(200),CS(200),ADMS0160
     4                 ZS(200),TETA(200),COSAVE(200),E2SAVE(200)        ADMS0170
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                                ADMS0180
       REAL MP,MQ,MR,MS,MPP,MQP,MRP,MSP                                 ADMS0190
C                                                                       ADMS0200
      IR = 1                                                            ADMS0210
      GO TO ( 20 , 30 , 40 , 50 , 100 ),NFUNCT                          ADMS0220
   20 PDP(N+1) = AA0* P(N) + AA1* P(N-1) + AA2* P(N-2) +                ADMS0230
     1       HS*( BB0*PP(N) + BB1*PP(N-1) + BB2*PP(N-2) + BB3*PP(N-3) ) ADMS0240
      MP(N+1) = PDP(N+1) - (EB5/(EB5-E5))*(PDP(N) - CP(N))              ADMS0250
      RETURN                                                            ADMS0260
   30 PDQ(N+1) = AA0* Q(N) + AA1* Q(N-1) + AA2* Q(N-2) +                ADMS0270
     1       HS*( BB0*QP(N) + BB1*QP(N-1) + BB2*QP(N-2) + BB3*QP(N-3) ) ADMS0280
      MQ(N+1) = PDQ(N+1) - (EB5/(EB5-E5))*(PDQ(N) - CQ(N))              ADMS0290
      RETURN                                                            ADMS0300
   40 PDR(N+1) = AA0* R(N) + AA1* R(N-1) + AA2* R(N-2) +                ADMS0310
     1       HS*( BB0*RP(N) + BB1*RP(N-1) + BB2*RP(N-2) + BB3*RP(N-3) ) ADMS0320
      MR(N+1) = PDR(N+1) - (EB5/(EB5-E5))*(PDR(N) - CR(N))              ADMS0330
      RETURN                                                            ADMS0340
   50 PDS(N+1) = AA0* S(N) + AA1* S(N-1) + AA2* S(N-2) +                ADMS0350
     1       HS*( BB0*SP(N) + BB1*SP(N-1) + BB2*SP(N-2) + BB3*SP(N-3) ) ADMS0360
      MS(N+1) = PDS(N+1) - (EB5/(EB5-E5))*(PDS(N) - CS(N))              ADMS0370
      RETURN                                                            ADMS0380
C      EXECUTE A TEMPORARY RETURN TO MAIN TO PICK UP SOME REQUIRED      ADMS0390
C      VARIABLES AND COMPUTE MZP(N+1)  FROM EQUATIONS OF MOTION SUBROUTIADMS0400
C                                                                       ADMS0410
      ENTRY ADAMS1                                                      ADMS0420


C                                                                       ADMS0430
      GO TO ( 60 , 70 , 80 , 90 , 120 ),NFUNCT                          ADMS0440
   60 CP(N+1) = A0* P(N) + A1* P(N-1) + A2* P(N-2) +                    ADMS0450
     1     HS*( B0*PP(N) + B1*PP(N-1) + B2*PP(N-2) + BM1*MPP(N+1) )     ADMS0460
      PDPMC = PDP(N+1) - CP(N+1)                                        ADMS0470
      P(N+1) = CP(N+1) - (E5/(EB5-E5))*PDPMC                            ADMS0480
      RETURN                                                            ADMS0490
```

```
 70 CQ(N+1) = A0* Q(N) + A1* Q(N-1) + A2* Q(N-2) +                    ADMS0500
  1       HS*( B0*QP(N) + B1*QP(N-1) + B2*QP(N-2) + BM1*MQP(N+1) )    ADMS0510
    PDQMC = PDQ(N+1) - CQ(N+1)                                         ADMS0520
    Q(N+1) = CQ(N+1) - (E5/(EB5-E5))*PDQMC                             ADMS0530
    RETURN                                                             ADMS0540
 80 CR(N+1) = A0* R(N) + A1* R(N-1) + A2* R(N-2) +                    ADMS0550
  1       HS*( B0*PP(N) + B1*RP(N-1) + B2*RP(N-2) + BM1*MRP(N+1) )    ADMS0560
    PDRMC = PDR(N+1) - CR(N+1)                                         ADMS0570
    R(N+1) = CR(N+1) - (E5/(EB5-E5))*PDRMC                             ADMS0580
     RETURN                                                            ADMS0590
 90 CS(N+1) = A0* S(N) + A1* S(N-1) + A2* S(N-2) +                    ADMS0600
  1       HS*( B0*SP(N) + B1*SP(N-1) + B2*SP(N-2) + BM1*MSP(N+1) )    ADMS0610
    PDSMC = PDS(N+1) - CS(N+1)                                         ADMS0620
    S(N+1) = CS(N+1) - (E5/(EB5-E5))*PDSMC                             ADMS0630
     RETURN                                                            ADMS0640
100 WRITE (6, 110 )                                                    ADMS0650
110 FORMAT(42H0***** ERROR IN CALL TO ADAMS. NFUNCT = 5.              ADMS0660
  1         14H PROGRAM HALT.)                                         ADMS0665
    IR = 2                                                             ADMS0670
    RETURN                                                             ADMS0680
120 WRITE (6, 130 )                                                    ADMS0690
130 FORMAT(43H0***** ERROR IN CALL TO ADAMS1. NFUNCT = 5.             ADMS0700
  1         14H PROGRAM HALT.)                                         ADMS0705
    IR = 2                                                             ADMS0710
    RETURN                                                             ADMS0720
    END                                                                ADMS0730
```

```
*DECK JETVEL                                                           JETV0010
      OVERLAY(OVLY,7,2)                                                JETV0020
      PROGRAM JETVEL                                                   JETV0030
C                                                                      JETV0040
C      SUBROUTINE JETVELOCITY COMPUTES THE INDUCED VELOCITY AT EACH POINJETV0050
C      INPUT IN THE UNIFORM FLOW FIELD OF VELOCITY  U  SURROUNDING THE  JETV0060
C      EXHAUSTING JET OF INITIAL VELOCITY UJO                          JETV0070
C                                                                      JETV0080
      COMMON /JET3/ E1,E2,E3,CD,M                                      JETV0090
      COMMON /JET4/ HS,NMAX                                            JETV0100
      COMMON /JET5/ UJO,U,DO,THETA,H,ZMAXI,TETAMX,XOCL,YOCL,ZOCL,DIHED, JETV0110
     1              JPRINT,JNUMB,JETREP,ZGND,NSIGMA                    JETV0120
      COMMON /JET6/ MB(200),MU(200),PD(200),QD(200),RD(200),SD(200),   JETV0130
     1 ZD(200),SMINOR(200),DS(200),INJET(100),XI,ETA,ZETA,INJETP,NINJET,JETV0140
     2 SIGMA(200),HSPRME                                               JETV0150
      COMMON /JET7/ VXT(600),VYT(600),VZT(600)                         JETV0160
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                               JETV0170
      COMMON/JET11/NJETS,NCOLOQ,NFUSZ,NFUSY,NTOTAL,ISYM,FMACH,         JETV0180
     1              XP(600),YP(600),ZP(600)                            JETV0190
      DIMENSION  XCL(600),YCL(600),ZCL(600)                            JETV0200
       LOGICAL  IMAGE,NODIH,INJETF                                     JETV0210
       REAL   M,MB,MU                                                  JETV0220
C                                                                      JETV0230
      PI = 3.141592                                                    JETV0240
       INJETF = .FALSE.                                                JETV0250
       IF ( INJETP  .GE.  1 )  INJETF = .TRUE.                         JETV0260
       NINJET = 0                                                      JETV0270
      RADIAN = DIHED*PI/180.                                           JETV0280
      SINDIH = SIN(RADIAN)                                             JETV0290
      COSDIH = COS(RADIAN)                                             JETV0300
```

```
      NODIH = .TRUE.                                                   JETV0310
      IF (ABS(COSDIH-1.) .GT. 1.0E-10) NODIH=.FALSE.                   JETV0320
      BETA = SQRT(1. - FMACH**2)                                       JETV0330
      IF (FMACH .LT. 1.0E-10) GO TO 30                                 JETV0340
      XOCL = XOCL/BETA                                                 JETV0350
      DO 20 I=1,NMAX                                                   JETV0360
   20 SD(I) = SD(I)/BETA                                               JETV0370
   30 XO = XOCL                                                        JETV0380
      YO = YOCL                                                        JETV0390
      ZO = ZOCL                                                        JETV0400
      IMAGE = .FALSE.                                                  JETV0410
      IF (ISYM .EQ. 0) IMAGE = .TRUE.                                  JETV0420
      DO 40 K=1,NTOTAL                                                 JETV0430
      VXT(K) = 0.0                                                     JETV0440
      VYT(K) = 0.0                                                     JETV0450
   40 VZT(K) = 0.0                                                     JETV0460
C      TRANSFORM POINTS AT WHICH INDUCED VELOCITIES ARE TO BE COMPUTED  JETV0470
C      IN THE PRINCIPAL REF. SYSTEM (XP,YP,ZP) INTO JET CENTERLINE     JETV0480
C      REF. SYSTEM (XCL,YCL,ZCL)                                       JETV0490
   50 DO 60 K=1,NTOTAL                                                 JETV0500
      XCL(K) = XP(K) - XO                                              JETV0510
      YCL(K) = -YP(K) + YO                                             JETV0520
   60 ZCL(K) = -ZP(K) + ZO                                             JETV0530
C ACCOUNT FOR DIHEDRAL EFFECTS ...                                     JETV0540
      IF (NODIH) GO TO 80                                              JETV0550
      DO 70 K=1,NTOTAL                                                 JETV0560
      YTEMP = YCL(K)                                                   JETV0570
      YCL(K) = ZCL(K)*SINDIH + YCL(K)*COSDIH                           JETV0580
   70 ZCL(K) = ZCL(K)*COSDIH - YTEMP *SINDIH                           JETV0590
C      THE POINT (XP,YP,ZP) IS NOW IN TERMS OF JET C/L REF.SYS. (XCL,YCLJETV0600
   80  CONTINUE                                                        JETV0610
C      BEGIN LOOP THAT SELECTS NEXT POINT   FOR WHICH VELOCITIES ARE    JETV0620
C      TO BE COMPUTED  AND THAT ASSIGNS THE FINAL SINK AND DOUBLET     JETV0630
C      CONTRIBUTIONS TO THAT POINT.                                    JETV0640
   90 DO 290 K=1,NTOTAL                                                JETV0650
       VXSINK = 0.                                                     JETV0660
       VYSINK = 0.                                                     JETV0670
       VZSINK = 0.                                                     JETV0680
       VXDBLT = 0.                                                     JETV0690
       VYDBLT = 0.                                                     JETV0700
       VZDBLT = 0.                                                     JETV0710
C      FOR THE NEXT POINT (XCL,YCL,ZCL)  COMPUTE AND SUM ALL SINK AND DOJETV0720
C      VELOCITY CONTRIBUTIONS FROM  NMAX SINK/DOUBLET DISTRIBUTION ALONGJETV0730
       DO 260   N=1,NMAX                                               JETV0740
C      COMPUTE THE TOTAL SINK VELOCITY CONTRIBUTION AT POINT (XCL,YCL,ZCJETV0750
C      SUMMING ALL VELOCITY CONTRIBUTIONS DUE TO THE  C/L SINK DISTRIBUTJETV0760
  100  D1=(ZD(N)-ZCL(K))**2 + (SD(N)-XCL(K))**2                        JETV0770
  110  D2=SQRT((ZD(N)-ZCL(K))**2+(SD(N)-XCL(K))**2+(YCL(K)-QD(N)/2.)**2)JETV0780
  120  D3=SQRT((ZD(N)-ZCL(K))**2+(SD(N)-XCL(K))**2+(YCL(K)+QD(N)/2.)**2)JETV0790
C                                                                      JETV0800
C     (NOTE) USINK,VSINK,WSINK ARE SINK CONTRIBUTIONS IN THE X,Y,Z DIRECJETV0810
C     IF D1=0.,THERE IS NO X OR Z CONTRIBTUION AND USINK AND WSINK ARE SJETV0820
       USINK = 0.0                                                     JETV0830
       WSINK = 0.0                                                     JETV0840
C                                                                      JETV0850
  130  VSINK =          - ( MB(N)/(4.*PI))*( 1./D2 - 1./D3 )           JETV0860
       IF ( D1 .EQ. 0.0 )  GO TO 160                                   JETV0870
  140  USINK =          - ( MB(N)*(SD(N)-XCL(K)) /(4.*PI*D1) )  *      JETV0880
     1           ( (YCL(K)-QD(N)/2.) /D2 - ( YCL(K)+QD(N)/2.) /D3   )  JETV0890
  150  WSINK =          - ( MB(N)*(ZD(N)-ZCL(K)) /(4.*PI*D1) )  *      JETV0900
```

```
     1           ( (YCL(K)-QD(N)/2.) /D2 -   (YCL(K)+QD(N)/2.) /D3   ) JETV0910
  160 CONTINUE                                                         JETV0920
       VXSINK = VXSINK + USINK                                         JETV0930
       VYSINK = VYSINK + VSINK                                         JETV0940
       VZSINK = VZSINK + WSINK                                         JETV0950
C      COMPUTE THE TOTAL DOUBLET VELOCITY CONTRIBUTION AT POINT (XCL,YCLJETV0960
C      SUMMING ALL VELOCITY CONTRIBUTIONS DUE TO THE C/L DOUBLET DISTRIBJETV0970
C      XI,ETA,ZETA  ARE DOUBLET COORDINATES AND HAVE THE FOLLOWING     JETV0980
C      RELATIONSHIP WITH XCL,YCL,ZCL .                                 JETV0990
C      IF LAST POINT N, USE PREVIOUSLY COMPUTED SINA AND COSA.         JETV1000
       IF ( N  .EQ.  NMAX )  GO TO 200                                 JETV1010
  170  DENOM = SQRT ((ZD(N+1)-ZD(N))**2 + (SD(N+1)-SD(N))**2)          JETV1020
  180  SINA = (SD(N+1)-SD(N)) /DENOM                                   JETV1030
  190  COSA = (ZD(N+1)-ZD(N)) /DENOM                                   JETV1040
  200  XI  =    (XCL(K)-SD(N))*SINA - (ZD(N)-ZCL(K))*COSA              JETV1050
       ETA = -YCL(K)                                                   JETV1060
  210  ZETA=  (XCL(K)-SD(N))*COSA + (ZD(N)-ZCL(K))*SINA                JETV1070
C      IF INJETP FLAG IS ON (INJETF = TRUE), TEST FOR POINTS INSIDE THE JETV1080
       IF (INJETF)  CALL PTEST                                         JETV1090
C      UI,VI,WI  ARE VELOCITY COMPONENTS IN THE DOUBLET COORDINATE SYSTEJETV1100
C      HAVE THE FOLLOWING RELATIONSHIP WITH THE C/L REF. SYSTEM (XCL,YCLJETV1110
  220  D4 = ( XI**2 + ETA**2 + ZETA**2 )                               JETV1120
  230  UI = ( -6.*XI *ZETA*MU(N))/(4.*PI*     D4**(2.5)    )           JETV1130
  240  VI = ( -6.*ETA*ZETA*MU(N))/(4.*PI*     D4**(2.5)    )           JETV1140
  250  WI = (2.*MU(N)) /(4.*PI*D4**1.5) *( 1. -(3.*ZETA**2)/D4  )      JETV1150
       UBLOCK =            (UI*SINA + WI*COSA)                         JETV1160
       VBLOCK =          - VI                                          JETV1170
       WBLOCK =            (UI*COSA - WI*SINA)                         JETV1180
       VXDBLT = VXDBLT + UBLOCK                                        JETV1190
       VYDBLT = VYDBLT + VBLOCK                                        JETV1200
       VZDBLT = VZDBLT + WBLOCK                                        JETV1210
  260  CONTINUE                                                        JETV1220
C      ASSIGN THE COMPONENT SINK AND DOUBLET VELOCITIES AT POINT (XCL,YCJETV1230
C      DUE TO ALL SINK AND DOUBLET SOURCE DISTRIBUTIONS ALONG JET C/L ANJETV1240
C      TRANSFORM CENTERLINE REF. VELOCITIES BACK INTO THE PRINCIPAL REF.JETV1250
       VXS = VXSINK                                                    JETV1260
       VYS = -VYSINK                                                   JETV1270
       VZS = -VZSINK                                                   JETV1280
C FIRST ACCOUNT FOR DIHEDRAL JET ANGLE EFFECTS ...                     JETV1290
C (NOTE ... TO TRANSFORM BACK, SIMPLY ROTATE BY -DIHED)                JETV1300
      IF (NODIH) GO TO 270                                             JETV1310
      VYTEMP = VYS                                                     JETV1320
      VYS = VZS*(-SINDIH) + VYS*(COSDIH)                               JETV1330
      VZS = VZS*(COSDIH) - VYTEMP*(-SINDIH)                            JETV1340
  270 CONTINUE                                                         JETV1350
       VXD = VXDBLT                                                    JETV1360
       VYD = -VYDBLT                                                   JETV1370
       VZD = -VZDBLT                                                   JETV1380
      IF (NODIH) GO TO 280                                             JETV1390
      VYTEMP = VYD                                                     JETV1400
      VYD = VZD*(-SINDIH) + VYD*(COSDIH)                               JETV1410
      VZD = VZD*(COSDIH) - VYTEMP*(-SINDIH)                            JETV1420
  280 CONTINUE                                                         JETV1430
C      RUNNING SUM OF COMPONENT VELOCITIES (INCLUDING IMAGE EFFECTS)   JETV1440
      VXT(K) = VXS + VXD + VXT(K)                                      JETV1450
      VYT(K) = VYS + VYD + VYT(K)                                      JETV1460
      VZT(K) = VZS + VZD + VZT(K)                                      JETV1470
C      SUM OF COMPONENT VELOCITIES FOR THIS JET                        JETV1480
       VXTJ = VXS + VXD                                                JETV1490
       VYTJ = VYS + VYD                                                JETV1500
```

```
       VZTJ = VZS + VZD
  290  CONTINUE                                                        JETV1510
       NINMAX = NINJET                                                 JETV1520
       IF (NINJET .GT. 100)  NINMAX = 100                              JETV1530
      IF(NINJET .GT. 0) WRITE(6, 300 ) NINJET,(INJET(I),I=1,NINMAX)    JETV1540
  300 FORMAT(1H1/// 17X,35H***** WARNING *****   THE FOLLOWING,I4,     JETV1550
     1        47H FLOW FIELD POINTS FELL INSIDE THE JET ENVELOPE //    JETV1560
     2        39X,51HTHE INDUCED VELOCITIES AT THESE POINTS ARE IN ERROR JETV1570
     3        // 10(/10X,10I10))                                       JETV1580
      IF(.NOT. IMAGE) GO TO 310                                        JETV1590
      YO = -YO                                                         JETV1600
      SINDIH = -SINDIH                                                 JETV1610
      IMAGE = .FALSE.                                                  JETV1620
       INJETF = .FALSE.                                                JETV1630
C NOW CALCULATE THE IMAGE EFFECTS ...                                  JETV1640
      GO TO 50                                                         JETV1650
  310 CONTINUE                                                         JETV1660
      END                                                              JETV1670
                                                                       JETV1680
```

```
*DECK MOTION
       SUBROUTINE MOTION                                               MOTN0010
C                                                                      MOTN0020
C      SUBROUTINE 'MOTION' SOLVES THE EQUATIONS OF MOTION FOR REGIONS 1 MOTN0030
C                                                                      MOTN0040
      COMMON /JET3/ E1,E2,E3,CD,M                                      MOTN0050
      COMMON /JET6/ MB(200),MU(200),PD(200),QD(200),RD(200),SD(200),   MOTN0060
     1 ZD(200),SMINOR(200),DS(200),INJET(100),XI,ETA,ZETA,INJETP,NINJET,MOTN0070
     2 SIGMA(200),HSPRME                                               MOTN0080
      COMMON/JET8/P(200),PP(200),PDP(200),MP(200),MPP(200),CP(200),    MOTN0090
     1                 Q(200),QP(200),PDQ(200),MQ(200),MQP(200),CQ(200),MOTN0100
     2                 R(200),RP(200),PDR(200),MR(200),MRP(200),CR(200),MOTN0110
     3                 S(200),SP(200),PDS(200),MS(200),MSP(200),CS(200),MOTN0120
     4                 ZS(200),TETA(200),COSAVE(200),E2SAVE(200)       MOTN0130
      COMMON/JET9/NFUNCT,MFLAG,N,NP1,K,L                               MOTN0140
       REAL   M                                                        MOTN0150
       REAL MP,MQ,MR,MS,MPP,MQP,MRP,MSP                                MOTN0160
      DATA PI/3.1415926/                                               MOTN0170
      IPTNUM = K - MFLAG                                               MOTN0180
C (IPTNUM CAN ASSUME THE VALUE OF THE PRESENT (I.E. KNOWN) POINT OR THE MOTN0190
C (JUST FOUND) POINT.  WHEN CALLED BY RKUTTA, MFLAG=0 IMPLIES THE PRESENMOTN0200
C POINT.  WHEN CALLED BY ADAMS OR ADAMS1 MFLAG=1 IMPLIES THE PRESENT POIMOTN0210
      ZSTAR = ZS(IPTNUM)                                               MOTN0220
      UJSTAR = P(IPTNUM)                                               MOTN0230
      OJSTAR = Q(IPTNUM)                                               MOTN0240
      XI = R(IPTNUM)                                                   MOTN0250
C FIND THE DISTANCE INTO REGION 1 ...                                  MOTN0260
      FRAC1 = ZSTAR/HSPRME                                             MOTN0270
C IF FRAC .GT. 1  THEN WE ARE IN REGION 2 ...                          MOTN0280
      IF (FRAC1.GT. 1) GO TO 50                                        MOTN0290
C REGION 1 ...                                                         MOTN0300
      D = 1.0 - 0.75*FRAC1                                             MOTN0310
      DPRIME = -0.75/HSPRME                                            MOTN0320
      CD = (-1.0/D**2 + 6.6/D + 0.4) / 6.0                             MOTN0330
      E2 = 0.8                                                         MOTN0340
      IF (ZSTAR .GT. 10.0) GO TO 40                                    MOTN0350
      IF (FRAC1 .GT. 0.8) GO TO 30                                     MOTN0360
      IF (FRAC1 .GT. 0.6) GO TO 20                                     MOTN0370
                                                                       MOTN0380
```

```
      E2 = (10./32.) * 0.08                                             MOTN0390
      GO TO 40                                                          MOTN0400
   20 E2 = (12./32.) * 0.08                                             MOTN0410
      GO TO 40                                                          MOTN0420
   30 E2 = (21./32.) * 0.08                                             MOTN0430
   40 CONTINUE                                                          MOTN0440
      GO TO 60                                                          MOTN0450
C REGION 2 ...                                                          MOTN0460
   50 D = 0.25                                                          MOTN0470
      DPRIME = 0.0                                                      MOTN0480
      CD = 1.8                                                          MOTN0490
      E2 = 0.08                                                         MOTN0500
   60 CJSTAR = PI*DJSTAR*SQRT( (1.0 + D**2)/2.0 )                       MOTN0510
      AJSTAR = PI*(DJSTAR**2)*D / 4.0                                   MOTN0520
      COST = 1.0 / SQRT( 1.0 + XI**2 )                                  MOTN0530
      SINT = XI * COST                                                  MOTN0540
C ESTAR = E / (RHO*UJO*DJO)                                             MOTN0550
      ESTAR = (E1*DJSTAR*COST/M) + (E2*UJSTAR*CJSTAR*(M*UJSTAR-SINT))/  MOTN0560
     1                                   (UJSTAR*M + E3*COST)           MOTN0570
      CDSAVE(L+1) = CD                                                  MOTN0580
      E2SAVE(L+1) = E2                                                  MOTN0590
      GO TO ( 70 , 80 , 90 , 100 , 70 ), NFUNCT                         MOTN0600
   70 PPRIME = (ESTAR / (AJSTAR*UJSTAR*M))* ( XI - UJSTAR*M/COST )      MOTN0610
      PP(K) = PPRIME                                                    MOTN0620
      MPP(K) = PPRIME                                                   MOTN0630
      IF (NFUNCT .EQ. 1) RETURN                                         MOTN0640
   80 QPRIME = ( 1.0/(2.0*D*UJSTAR) ) * ( 4.0*ESTAR/(PI*DJSTAR*COST) -  MOTN0650
     1                         DJSTAR*( D*PP(K) + UJSTAR*DPRIME) )      MOTN0660
      QP(K) = QPRIME                                                    MOTN0670
      MQP(K) = QPRIME                                                   MOTN0680
      IF (NFUNCT .EQ. 2) RETURN                                         MOTN0690
   90 RPRIME = ( (1.0 + XI**2 )**1.5 / (AJSTAR*UJSTAR**2) ) *           MOTN0700
     1          ( ESTAR*COST/M + 0.5*CD*DJSTAR*(COST/M)**2 )            MOTN0710
      RP(K) = RPRIME                                                    MOTN0720
      MRP(K) = RPRIME                                                   MOTN0730
      IF (NFUNCT .EQ. 3) RETURN                                         MOTN0740
  100 SP(K) = R(K)                                                      MOTN0750
      MSP(K) = R(K)                                                     MOTN0760
      RETURN                                                            MOTN0770
      END                                                               MOTN0780
```

```
*DECK DRK1                                                              DRK10010
      SUBROUTINE DRK1 ( A, B, C, DX, N, F, G, IS )                      DRK10020
C                                                                       DRK10030
C          A  = INPUT  = INITIAL VALUE OF INDEPENDENT VARIABLE           DRK10040
C             = OUTPUT = CURRENT VALUE OF INDEPENDENT VARIABLE           DRK10050
C                                                                       DRK10060
C          B  = INPUT  = INITIAL VALUES OF DEPENDENT VARIABLES          DRK10070
C             = OUTPUT = CURRENT VALUES OF DEPENDENT VARIABLES          DRK10080
C                      = AN ARRAY OF LENGTH N                           DRK10090
C                                                                       DRK10100
C          C  = INPUT  = AN ARRAY (OF LENGTH N) CONTAINING THE CURRENT  DRK10110
C                        VALUES OF THE DERIVATIVES.  THESE VALUES MUST  DRK10120
C                        BE COMPUTED BY THE MAIN PROGRAM.               DRK10130
C                                                                       DRK10140
C          DX = INPUT  = THE INCREMENT IN THE INDEPENDENT VARIABLE      DRK10150
C                                                                       DRK10160
C          N  = INPUT  = THE NUMBER (INTEGER) OF SIMULTANEOUS EQUATIONS DRK10170
C                        TO BE INTEGRATED                               DRK10180
C                                                                       DRK10190
C          F  = OUTPUT = AN ARRAY (OF LENGTH N) USED TO STORE VALUES OF DRK10200
C                        THE ARRAY B                                    DRK10210
C                                                                       DRK10220
C          G  = OUTPUT = AN ARRAY (OF LENGHT 4*N) WHICH CONTAINS        DRK10230
C                        INTERMEDIATE VALUES COMPUTED BY THE            DRK10240
C                        SUBROUTINE.  FOUR ENTRIES OF G ARE USED TO     DRK10250
C                        COMPUTE ONE ENTRY OF B.                        DRK10260
C                                                                       DRK10270
C          IS = INPUT  = INITIALLY MUST BE SET TO ZERO (INTEGER)        DRK10280
C             = OUTPUT = CONTROL FLAG (INTEGER) MODIFIED BY THE         DRK10290
C                        SUBROUTINE                                     DRK10300
C                                                                       DRK10310
C     NOTE THAT THERE IS NO CHECK ON THE VALIDITY OF THE VARIABLE IS.   DRK10320
C                                                                       DRK10330
      DIMENSION  B( 1 ), C( 1 ), F( 1 ), G( 1 )                         DRK10340
C                                                                       DRK10350
      IS = IS + 1                                                       DRK10360
      GO TO ( 20 , 40 , 70 , 90 ), IS                                   DRK10370
C                                                                       DRK10380
C     FIRST ENTRY                                                       DRK10390
C                                                                       DRK10400
   20 E = A                                                             DRK10410
      DO 30 I = 1, N                                                    DRK10420
      F( I ) = B( I )                                                   DRK10430
      G( 4 * I - 3 ) = C( I ) * DX                                      DRK10440
   30 B( I ) = F( I ) + G( 4 * I - 3 ) / 2.0                            DRK10450
      GO TO 60                                                          DRK10460
C                                                                       DRK10470
C     SECOND ENTRY                                                      DRK10480
C                                                                       DRK10490
   40 DO 50 I = 1, N                                                    DRK10500
      G( 4 * I - 2 ) = C( I ) * DX                                      DRK10510
   50 B( I ) = F( I ) + G( 4 * I - 2 ) / 2.0                            DRK10520
   60 A = E + DX / 2.0                                                  DRK10530
      GO TO 110                                                         DRK10540
C                                                                       DRK10550
C     THIRD ENTRY                                                       DRK10560
C                                                                       DRK10570
   70 DO 80 I = 1, N                                                    DRK10580
      G( 4 * I - 1 ) = C( I ) * DX                                      DRK10590
   80 B( I ) = F( I ) + G( 4 * I - 1 )                                  DRK10600
      A = E + DX                                                        DRK10610
      GO TO 110                                                         DRK10620
C                                                                       DRK10630
C     FOURTH ENTRY                                                      DRK10640
C                                                                       DRK10650
   90 DO 100 I = 1, N                                                   DRK10660
      G( 4 * I ) = C( I ) * DX                                          DRK10670
      B( I ) = G( 4 * I - 3 ) + 2.0 * ( G( 4 * I - 2 ) + G( 4 * I - 1 ))DRK10680
  100 B( I ) = ( B( I ) + G( 4 * I ) ) / 6.0 + F( I )                   DRK10690
      IS = 0                                                            DRK10700
  110 RETURN                                                            DRK10710
      END                                                               DRK10720
```

```
*DECK STAGE8                                                            STG80010
      OVERLAY(OVLY,10,0)                                                STG80020
      PROGRAM STAGE8                                                    STG80030
C                                                                       STG80040
C  THIS SUBROUTINE CONTROLS THE MATRIX QUASI-INVERSION FOR WING OR TAIL STG80050
C                                                                       STG80060
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                STG80070
      COMMON /SOLV1/ WKAREA(10000)                                      STG80080
C                                                                       STG80090
C  JET-WING MATRIX                                                      STG80100
      IF(LOGIC .EQ. 1) CALL OVERLAY(4HOVLY,8,1)                         STG80110
C  TAIL MATRIX                                                          STG80120
      IF(LOGIC .EQ. 2) CALL OVERLAY(4HOVLY,8,2)                         STG80130
      CONTINUE                                                          STG80140
      END                                                               STG80150
```

```
*DECK MATRIX                                                            MATR0010
      SUBROUTINE MATRIX(A,RHS,NO,MRHS,KD,NMOST,MMOST,                   MATR0020
     1         NI,MM,NO,NAT,LTAPE,IR)                                   MATR0030
C                                                                       MATR0040
C               DIRECT MATRIX SOLUTION AND QUASI-INVERSION              MATR0050
C*** ***LTAPE IS THE TAPE THE L(I,J) MATRIX WILL BE PUT ON              MATR0060
C*** ***NATAPE IS A SCRATCH TAPE                                        MATR0070
C*** ***THE TRIANGULAR MATRIX EXCEPT FOR THE LAST K ROWS WILL BE KEPT ONMATR0080
C*** ***TAPE MM                                                         MATR0090
C*** ***THE LAST K ROWS OF THE TRIANGULAR MATRIX WILL BE PUT ON THE     MATR0100
C*** ***LTAPE BEHIND THE RHS MATRIX                                     MATR0110
C                                                                       MATR0120
      DIMENSION A(KD),RHS(NMOST,MMOST)                                  MATR0130
      LOGICAL JPASS1,LASTRS,LAST                                        MATR0140
C                                                                       MATR0150
      IR = 2                                                            MATR0160
      REWIND LTAPE                                                      MATR0170
      NATAPE = NAT                                                      MATR0180
      REWIND NATAPE                                                     MATR0190
      N = NO                                                            MATR0200
      KORE = KD                                                         MATR0210
   20 M = KORE / N  -  1                                                MATR0220
      MMAX = MIN0(MRHS,M)                                               MATR0230
      NPM = N + MMAX                                                    MATR0240
      IF( (3*NPM) .GT. KORE ) RETURN                                    MATR0250
      M = 0                                                             MATR0260
      NPM = N                                                           MATR0270
      MT = MM                                                           MATR0280
      REWIND MT                                                         MATR0290
      NIN = NI                                                          MATR0300
      REWIND NIN                                                        MATR0310
      NOUT = NO                                                         MATR0320
      REWIND NOUT                                                       MATR0330
      MP1 = M + 1                                                       MATR0340
      NN = N                                                            MATR0350
      NEL = NPM                                                         MATR0360
      NLCNT = 0                                                         MATR0370
C - - CALCULATE THE MAXIMUM NO. OF ROWS, 'K'                            MATR0380
   30 K = (KORE - NEL) / NEL                                            MATR0390
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE            MATR0400
      LAST = K .GE. NN                                                  MATR0410
      IF( .NOT. LAST ) GO TO 40                                         MATR0420
      K = NN                                                            MATR0430
      B = 3 + MMAX*2                                                    MATR0440
      C = 2 * (1 + MMAX - KORE )                                        MATR0450
      KTEMP = ( -B + SQRT(B**2  -  4 * C )  )  / 2                      MATR0460
      IF(KTEMP .GE. K)GO TO 40                                          MATR0470
C***   *  WE MUST REDUCE THE FINAL K                                    MATR0480
      K = KTEMP                                                         MATR0490
      LAST = .FALSE.                                                    MATR0500
C  ** IF THERE WOULD BE ONLY 1 ROW IN THE NEXT PASS,                    MATR0510
C  ** REDUCE THE NUMBER OF ROWS IN THIS PASS                            MATR0520
   40 IF((NEL-K) .EQ. 1) K = K - 1                                      MATR0530
C - - READ 'K' ROWS OF THE AUGMENTED 'A' MATRIX                         MATR0540
      NT = 0                                                            MATR0550
      DO 50 IB = 1, K                                                   MATR0560
      NS = NT + 1                                                       MATR0570
      NT = NT + NEL                                                     MATR0580
   50 CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                            MATR0590
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROWMATR0600
      IF(K .EQ. 1) GO TO 120                                            MATR0610
C - - 'K' IS GREATER THAN '1' SO WE CAN START THE TRIANGULARIZATION     MATR0620
      NELP1 = NEL + 1                                                   MATR0630
      NS = - NEL                                                        MATR0640
      NELP2 = NELP1 + 1                                                 MATR0650
C - - FORM THE 'TRAPEZOIDAL' ARRAY  (B)                                 MATR0660
      DO 60 IB = 2, K                                                   MATR0670
      NP = NELP2 - IB                                                   MATR0680
      NS = NS + NELP1                                                   MATR0690
      NT = NS                                                           MATR0700
      DO 60 IO = IB, K                                                  MATR0710
      NT = NT + NEL                                                     MATR0720
      MN = NT                                                           MATR0730
      NB = NS                                                           MATR0740
      A(NT) = A(NT) / A(NS)                                             MATR0750
      DO 60 NF = 2, NP                                                  MATR0760
      MN = MN + 1                                                       MATR0770
      NB = NB + 1                                                       MATR0780
   60 A(MN) = A(MN) - A(NT) * A(NB)                                     MATR0790
C*** ***WRITE PART OF THE LMATRIX ON LTAPE   (TRIANGULAR PART)          MATR0800
      WRITE (LTAPE)K                                                    MATR0810
      NLCNT = NLCNT + 1                                                 MATR0820
      LBEG =NELP1                                                       MATR0830
      KM1 = K - 1                                                       MATR0840
      DO 70 IB = 1,KM1                                                  MATR0850
      LEND = LBEG + IB- 1                                               MATR0860
      CALL SAVE(LTAPE, 1, IB, IB, A(LBEG), 1, AA2)                      MATR0870
   70 LBEG = LBEG + NN                                                  MATR0880
C - - WRITE THE 'TRAPEZOIDAL' MATRIX ON TAPE                            MATR0890
      NT = 0                                                            MATR0900
      NP = NEL                                                          MATR0910
      NS = - NEL                                                        MATR0920
      DO 80 IO = 1, K                                                   MATR0930
      NS = NS + NELP1                                                   MATR0940
      NT = NT + NEL                                                     MATR0950
      CALL  SAVE(MT, 2, NP, NP, A(NS), 1, AA2)                          MATR0960
   80 NP = NP - 1                                                       MATR0970
      IF (LAST) GO TO 120                                               MATR0980
      NP = NP - M                                                       MATR0990
      NS = KORE - NEL + 1                                               MATR1000
C - - READ ANOTHER ROW                                                  MATR1010
      DO 110 IO = 1, NP                                                 MATR1020
```

```
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)
C - - MODIFY THIS ROW BY THE 'TRAPEZOIDAL' ARRAY
      NT = 1
      MN = NS
      DO 100 IB = 1, K
      NB = NT
      NF = MN + 1
      A(MN) = A(MN) / A(NT)
      DO 90  NN = NF, KORE
      NB = NB + 1
   90 A(NN) = A(NN) - A(MN) * A(NB)
      MN = NF
  100 NT = NT + NELP1
C - - WRITE THE MODIFIED ROW ON TAPE
C*** ***WRITE REST OF LMATRIX ON LTAPE
      MNM1 = MN - 1
      NN1 = MNM1 - NS + 1
      CALL  SAVE(LTAPE, 1, NN1, NN1, A(NS), 1, AA2)
      NN1 = KORE - MN + 1
  110 CALL  SAVE(NOUT, 1, NN1, NN1, A(MN), 1, AA2)
      REWIND NOUT
      REWIND NIN
C - - SWITCH THE TAPES
      NT = NIN
      NIN = NOUT
      NOUT = NT
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK
      NEL = NEL - K
      NN = NEL - M
      GO TO 30
C - - REWIND ALL TAPES
  120 REWIND NIN
      REWIND NOUT
  130 NI = KORE - K * M + 1
      REWIND LTAPE
      REWIND MT
C*** ***CALCULATE THE NUMBER OF COLUMNS TO BRING OFF OF THE RHS TAPE
      MTOTAL = 0
      M = MMAX
      IF(M .EQ. 0)GO TO 500
C*** ***MTOTALIS THE TOTAL NUMBER OF RHS COLUMNS ALREADY BROUGHT IN
  140 MTOTAL = MTOTAL + M
      LASTRS = MTOTAL.GE. MRHS
      MTOTAL = MTOTAL - M
      IF (LASTRS)M = MRHS - MTOTAL
      MTOTAL = MTOTAL + M
C*** ***BRING IN M COLUMNS OF RHS
      KINIT = KORE - (M*N)
      IINIT = KINIT
      NBEG = KINIT
      DO 150 J =1,M
      DO 150 I = 1,N
      NBEG = NBEG + 1
  150 A(NBEG) = RHS(I,J)
C*** ***BRING IN L(I,J) MATRIX AND APPLY IT TO RHS
      NBEG = 1 + KINIT
      NEND = 1 + (M-1) * N + KINIT
      KSUM = 0
C*** ***DO TRIANGULAR SECTION OF LMATRIX
  160 READ (LTAPE)K
C*** ***KSUM IS THE TOTAL NUMBER OF L ROWS THAT WILL
C*** ***BE READ AFTER THIS TRIANGULAR SECTION IS FINISHED
      KSUM = KSUM + K
      KM1 = K - 1
C***   ***NOTE THAT KM1 CAN'T BE 0 SINCE K CAN'T BE 1 AND STILL HAVE SOM
C***   ***ON THE LTAPE
      DO 190  I = 1,KM1
      NBEG = NBEG + 1
      NEND = NEND + 1
C*** ***READ 1 ROW OF L(I,J) FROM LTAPE---K-1 TIMES---EACH TIME
C*** ***STARTING WITH L(1)
      CALL  GETT(LTAPE, 1, I, A, 1, AA2)
      JCNT = -1
C*** ***REDUCE THE RHS BY GOING ACROSS A SOLUTION ROW (WHICH
C*** ***ARE NOT IN CONSECUTIVE ORDER, BUT A(1), A(N+1), A(2N+1) ETC.)
      DO 180 NPP = NBEG,NEND,N
      JCNT = JCNT + 1
      SUM = 0.0
      NROW = KINIT + ( JCNT * N )
      DO 170 NN=1,I
      NROW = NROW + 1
  170 SUM = SUM + ( A(NN)*A(NROW))
  180 A(NPP) = A(NPP) - SUM
  190 CONTINUE
      IF (KSUM .EQ. N)GO TO 230
C*** ***KSUM = N IF YOU HAVE READ ENTIRE LMATRIX AND
C*** ***THERE IS NO CONSTANT SECTION LEFT
      NTBEG = NBEG
      NTEND = NEND
      KSUMP1 = KSUM + 1
C*** ***READ REST OF LROWS 1 ROW AT A TIME FOR CONSTANT SECTION
      DO 220 I=KSUMP1,N
      NTBEG = NTBEG + 1
      NTEND = NTEND + 1
      CALL  GETT(LTAPE, 1, K, A, 1, AA2)
      JCNT = -1
C*** ***PARTIALLY  REDUCE A RHS ACROSS A RHS ROW BY APPLYING K NUMBER
C*** ***OF L(I,J) S
      DO 210 NPP = NTBEG,NTEND,N
      JCNT = JCNT + 1
      SUM = 0.0
      NROW = KINIT + ( JCNT * N )
      DO 200 NN = 1,K
      NROW = NROW + 1
  200 SUM = SUM + ( A(NN) * A(NROW) )
  210 A(NPP) = A(NPP) - SUM
  220 CONTINUE
      NBEG = NBEG + 1
      NEND = NEND + 1
C*** ***KINIT IS NOW FAR DOWN A COLUMN OF RHS TO START MULTIPLYING BY
C*** ***L(I,J) AT EACH PASS THROUGH
      KINIT = KINIT + K
C*** ***IF KSUMP1 = N THERE ARE NO MORE L(I,J)'S LEFT
      IF(KSUMP1 .LT. N)GO TO 160
C*** ***WRITE OUT ALL BUT LAST K ROWS OF RHS IN ROW ORDER ON NATAPE
  230 B = 4*M + 3
      C = -2 * KORE
      K = (  -B  +  SQRT( B**2  -  4*C )  )/2
      IF(K .GT. ND) K = ND
      KF = K
      KM1 = K - 1
      KLEFT = N - KF + IINIT
      INITP1 = IINIT + 1
      NEND = (M-1)*N + IINIT
      DO 240 NPP = INITP1,KLEFT
      NEND = NEND + 1
  240 WRITE(NATAPE) ( A(J),J=NPP,NEND,N)
      REWIND NATAPE
C*** ***JPASS1 IS TRUE ON 1ST PASS THRU BACK SOLUTION
      JPASS1 = .TRUE.
C***   ***PUT REMAINING RHS IN CONTIGUOUS LOCATIONS BY COLUMNS
C***  FROM KORE - (M * KF) + 1  TO  KORE
      NNEW = KORE - KF + 1
      MM1 = M - 1
C***   ***IF M = 1, THE ELTS OF THE 1 RHS COLUMN ARE ALREADY IN CONTIGUO
C***   ***LOCATIONS
      IF( M .EQ. 1 )GO TO 260
      DO 250 I = 1,MM1
      NOLD = KORE - (I*N) + 1
      DO 250 J = 1,KF
      NNEW = NNEW - 1
      NOLD = NOLD - 1
      A(NNEW) = A(NOLD)
  250 CONTINUE
  260 CONTINUE
C***   ***NOW NNEW = KORE - (M*KF) + 1
C***   ***NOW NOLD = KORE - (M - 1) * N + 1 - KF
C   ***   SKIP 1ST PART OF TRAPEZOIDAL MATRIX & READ LAST K ROWS
C*** ***ATTATCH RHS TO IT SO THAT EVERYTHING IS IN CONSECUTIVE ORDER
      NREMAN = ND - K
      IF(NREMAN .EQ. 0)GO TO 280
      DO 270 I = 1,NREMAN
  270 READ(MT) IDUMMY
  280 NEND = 0
      KCNT = K
      NNEW = NNEW - 1
C***   ***NOTE THAT K = KF WHICH IS ALREADY KNOWN IN CORE
      DO 290 JCNT = 1,K
      NBEG = NEND + 1
      CALL  GETT(MT, 4, KCNT, A(NBEG), 1, AA2)
      KCNT = KCNT - 1
      NEND = NBEG + KCNT
      NNEW = NNEW + 1
      KEND = (MM1 * KF) + NNEW
      DO 290 NPP=NNEW,KEND,KF
      NEND = NEND + 1
  290 A(NEND) = A(NPP)
      REWIND LTAPE
      REWIND MT
C - - THERE, NOW WE CAN START THE BACK-SOLUTION
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(NI)
C***   ***NL IS THE LAST SUBSCRIPT + 1 OF THE TRAPEZOIDAL A MATRIX THAT
C***   ***CORE
      NL = NEND + 1
      NREM = N
      NPM = N + M
      NEL = NPM
      MP1 = M + 1
      LAST = K .EQ. N
      NPASS = 0
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS
  300 KM1 = K - 1
      KP1 = K + 1
      NS = NL - MP1
      NPASS = NPASS + 1
      DO 330 MN = 1, M
      NF = NS + MN
      A(NF) = A(NF) / A(NS)
      NT = NS
      IF (KM1 .EQ. 0) GO  TO 330
      DO 320 IB = 1, KM1
      NF = NF - IB - M
      NT = NT - MP1 - IB
      SUM = 0.0
      NP = NF
      N2 = MP1 + IB
      DO 310 IO = 1, IB
      NN = NT + IO
      NP = NP + N2 - IO
  310 SUM = SUM + A(NN) * A(NP)
  320 A(NF) = (A(NF) - SUM) / A(NT)
  330 CONTINUE
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(NI)
      NI = KORE + 1
      DO 350 NN = 1, K
      DO 340 MN = 1, M
      NL = NL - 1
      NI = NI - 1
  340 A(NI) = A(NL)
  350 NL = NL - NN
C - - WRITE THE SOLUTIONS ON TAPE
C
      WRITE (NIN) K
      NS = NI - 1
      DO 360 MN = 1, M
      NT = NS + MN
  360 WRITE (NIN ) (A(IO), IO = NT, KORE, M)
C - - TEST IF THIS IS THE LAST PASS
      IF (LAST) GO TO 470
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF
C     THE SOLUTIONS OBTAINED SO FAR  (EQ 21)
C * * NOTE..LOCATIONS A(1) TO A(NI-1) ARE NOW FREE TO USE
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'
      NELOLD = NEL
      KOLD = K
      NEL = NEL - K
      NREM = NREM - K
C
      NROW = NREM - K + 1
      IF (K .LT. NREM) GO TO 370
      LAST = .TRUE.
      NROW = 1
      K = NREM
  370 NS = 1
      NT = NELOLD + 1
C - - READ IN THE ROWS TO BE MODIFIED
      DO 450 IB = 1, NREM
      NT = NT - 1
      IF (IB .LE. NROW) GO TO 380
      NS = NS + NN
```

```
      NT = NT + NN                                                      MATR3430
  380 IF (.NOT. JPASS1 )GO TO 390                                       MATR3440
      NBEG = NT - M + 1                                                 MATR3450
C*** ***READ RHS FROM NATAPE                                            MATR3460
      CALL  GETT(NATAPE, 1, M, A(NBEG), 1, AA2 )                        MATR3470
      NT = NT - M                                                       MATR3480
  390 CALL  GETT(MT, 2, NN, A(NS), 1, AA2 )                             MATR3490
      IF( .NOT. JPASS1 ) GO TO 400                                      MATR3500
      NT = NT + M                                                       MATR3510
      NN = NN + M                                                       MATR3520
  400 NP = N1 - 1                                                       MATR3530
      NF = NT - M - KM1                                                 MATR3540
      NN = NN - KOLD                                                    MATR3550
      DO 420 MN = 1, M                                                  MATR3560
      N2 = NF                                                           MATR3570
      NA = NP + MN                                                      MATR3580
      NB = NA                                                           MATR3590
      SUM = 0.0                                                         MATR3600
      DO 410 ID = 1, KOLD                                               MATR3610
      SUM = SUM + A(N2) * A(NA)                                         MATR3620
      N2 = N2 + 1                                                       MATR3630
  410 NA = NA + M                                                       MATR3640
      N2 = N2 + MN - 1                                                  MATR3650
  420 A(N2) = A(N2) - SUM                                               MATR3660
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW                MATR3670
      NL = NT - M + 1                                                   MATR3680
      IF (IB .GE. NROW) GO TO 430                                       MATR3690
      NF = NL - KP1                                                     MATR3700
      NN1 = NF - NS + 1                                                 MATR3710
      NN2 = NT - NL + 1                                                 MATR3720
      CALL  SAVE(NOUT, 4, NN, NN1, A(NS), NN2, A(NL))                   MATR3730
      GO TO 450                                                         MATR3740
  430 NF = NL - KOLD                                                    MATR3750
      DO 440 MN = NL, NT                                                MATR3760
      A(NF) = A(MN)                                                     MATR3770
  440 NF = NF + 1                                                       MATR3780
  450 CONTINUE                                                          MATR3790
C*** ***IF 1ST TIME THRU BACK SOLN,SWITCH TAPES SO THAT MT WHICH HAS THEMATR3800
C*** ***ORIGINAL TRAPEZOIDAL MATRIX ON IT BECOMES NATAPE AND IS NOT TO  MATR3810
C*** ***TAKE PART IN ALTERNATING SHRINKING MATRICES.  NATAPE BECOMES MT MATR3820
C***, ***AND THIS NOW DOES THE ALTERNATING WITH NOUT.                   MATR3830
      IF( .NOT. JPASS1 ) GO TO 460                                      MATR3840
      NTEMP = MT                                                        MATR3850
      MT = NATAPE                                                       MATR3860
      NATAPE = NTEMP                                                    MATR3870
      JPASS1= .FALSE.                                                   MATR3880
      REWIND NATAPE                                                     MATR3890
  460 REWIND MT                                                         MATR3900
      REWIND NOUT                                                       MATR3910
C - - SWITCH THE TAPES                                                  MATR3920
      NT = MT                                                           MATR3930
      MT = NOUT                                                         MATR3940
      NOUT = NT                                                         MATR3950
C - - LOOP BACK THRU THE SOLUTION                                       MATR3960
      NL = NF                                                           MATR3970
      GO TO 300                                                         MATR3980
C - - START TO WRAP IT UP                                               MATR3990
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE     MATR4000
C - - READ IN THE SOLUTIONS AND STORE THEM IN THEIR FINAL CORE LOCATIONSMATR4010
  470 REWIND NIN                                                        MATR4020
```

```
      N1 = N + 1                                                        MATR4030
      DO 480 IB = 1, NPASS                                              MATR4040
      READ (NIN) K                                                      MATR4050
      N1 = N1 - K                                                       MATR4060
      DO 480 ID = 1, M                                                  MATR4070
  480 CALL GETT(NIN,1,K,RHS(N1,ID),1,AA2)                               MATR4080
C ---  REWIND ALL INPUT TAPES                                           MATR4090
      REWIND NIN                                                        MATR4100
      REWIND MT                                                         MATR4110
      REWIND NOUT                                                       MATR4120
C   ***   IF TAPE WAS NEVER SWITCHED IT WOULD BE FOOLISH TO SWITCH BACK MATR4130
      IF(JPASS1)GO TO 490                                               MATR4140
C***   ***SWITCH TAPES                                                  MATR4150
C*** ***BACK SO THAT  MT WILL CONAIN THE TRAPEZOIDAL MATRIX             MATR4160
C*** ***NATAPE WILL HAVE NOTHING USEFUL ON IT.                          MATR4170
      NTEMP = NATAPE                                                    MATR4180
      NATAPE = MT                                                       MATR4190
      MT = NTEMP                                                        MATR4200
      REWIND NATAPE                                                     MATR4210
  490 IF( .NOT. LASTRS )GO TO 140                                       MATR4220
  500 REWIND LTAPE                                                      MATR4230
      REWIND MT                                                         MATR4240
      KRED = 0                                                          MATR4250
      DO 510 J=1,NLCNT                                                  MATR4260
      READ(LTAPE)KREAD                                                  MATR4270
      KRED = KRED + KREAD                                               MATR4280
      KREAD = KREAD + ( N - KRED - 1)                                   MATR4290
      DO 510 LREAD=1,KREAD                                              MATR4300
  510 READ(LTAPE)                                                       MATR4310
      DO 520 NROW = 1,NO                                                MATR4320
      CALL  GETT(MT, 2, ICNT, A(1), 1, AA2)                             MATR4330
  520 CALL  SAVE(LTAPE, 2, ICNT, ICNT, A, 1, AA2)                       MATR4340
C   ***   REWIND ALL FILES                                              MATR4350
      REWIND LTAPE                                                      MATR4360
      REWIND  N1                                                        MATR4370
      REWIND  MM                                                        MATR4380
      REWIND  NO                                                        MATR4390
      REWIND  NAT                                                       MATR4400
      IR = 1                                                            MATR4410
      RETURN                                                            MATR4420
      END                                                               MATR4430
```

```
*DECK BAKSUB                                                            BAKS0010
      SUBROUTINE BAKSUB(TRANS,SUMMER,NEWMAX,IO,ISTAB,IGND)              BAKS0020
C                                                                       BAKS0030
C  THIS SUBROUTINE BACK SUBSTITUTES THE COEFFICIENT MATRIX AND THE      BAKS0040
C  GAMMA SOLUTION TO OBTAIN THE RIGHT SIDE MATRIX FOR THE PURPOSE OF    BAKS0050
C  CHECKING THE MATRIX SOLUTION.                                        BAKS0060
C                                                                       BAKS0070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            BAKS0080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BAKS0090
      COMMON/SOLVI/GAMMA(400,10)                                        BAKS0100
      DIMENSION TRANS(NEWMAX),SUMMER(NEWMAX)                            BAKS0110
C                                                                       BAKS0120
      WRITE(6, 20 )                                                     BAKS0130
   20 FORMAT(1H1,47X,26HBACK SUBSTITUTION SOLUTION)                     BAKS0140
      NWT1 = NWT + 1                                                    BAKS0150
```

```
      IR1 = -1                                                          BAKS0160
      IR2 = 2 * NMAX - 1                                                BAKS0170
      IF(ISTAB .EQ. 0) GO TO 30                                         BAKS0180
      IR1 = 0                                                           BAKS0190
      IR2 = IR2 + 1                                                     BAKS0200
C  CYCLE THE RIGHT HAND SIDES                                           BAKS0210
   30 DO 80  N = 1,NCASES                                               BAKS0220
      IREAD = IR1                                                       BAKS0230
C  CYCLE THE MATRIX ROWS CORRESPONDING TO ELEMENTS ON THE WING          BAKS0240
      DO 60 I = 1,NEWMAX                                                BAKS0250
C  READ THE COEFFICIENT MATRIX ROW                                      BAKS0260
      IF(I .EQ. NWT1) IREAD = IR2                                       BAKS0270
      IREAD = IREAD + 2                                                 BAKS0280
   40 CALL READMS(IO,TRANS,NEWMAX,IREAD)                                BAKS0290
C  SUM UP THE TERMS FOR THIS ROW AND RIGHT SIDE                         BAKS0300
      SUMMER(I) = 0.00                                                  BAKS0310
      DO 50 J = 1,NEWMAX                                                BAKS0320
      SUMMER(I) = SUMMER(I) + TRANS(J) * GAMMA(J,N)                     BAKS0330
   50 CONTINUE                                                          BAKS0340
   60 CONTINUE                                                          BAKS0350
C  PRINT THE NTH RIGHT SIDE COLUMN                                      BAKS0360
      WRITE(6, 70 ) N,SUMMER                                            BAKS0370
   70 FORMAT(1H0,50X,17HRIGHT SIDE COLUMN,I4,60(/1X,10E13.5))           BAKS0380
   80 CONTINUE                                                          BAKS0390
      RETURN                                                            BAKS0400
      END                                                               BAKS0410
```

```
*DECK STG8W                                                             ST8W0010
      OVERLAY(OVLY,10,1)                                                ST8W0020
      PROGRAM STG8W                                                     ST8W0030
C                                                                       ST8W0040
C  THIS SUBROUTINE CONTROLS THE QUASI-INVERSION OF THE JET-WING MATRIX  ST8W0050
C                                                                       ST8W0060
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                ST8W0070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            ST8W0080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) ST8W0090
      COMMON/SOLV1/GAMMA(400,10)                                        ST8W0100
      COMMON /SOLVI/ WKAREA(10000)                                      ST8W0110
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          ST8W0120
      DIMENSION TRANS(400),SUMMER(400)                                  ST8W0130
      EQUIVALENCE (WKAREA(1),TRANS(1)),(WKAREA(401),SUMMER(1))          ST8W0140
C                                                                       ST8W0150
C  DECIDE WHICH MATRIX PERMANENT STORAGE UNIT IS TO BE USED             ST8W0160
      IO = 1                                                            ST8W0170
      IF(IGND .NE. 0) IO = 8                                            ST8W0180
C  PREPARE THE MATRIX INPUT TAPE UNIT 2                                 ST8W0190
      ISIZE = NEWMAX                                                    ST8W0200
      CALL PREPW(TRANS,ISIZE,NEWMAX,IO,ISTAB)                           ST8W0210
C  DEFINE THE REQUIRED PARAMETERS                                       ST8W0220
      IF(IPRINT .LT. 0) WRITE(6, 20 )                                   ST8W0230
   20 FORMAT(1H1,48X,23HJET-WING GAMMA SOLUTION)                        ST8W0240
      NMOST = 400                                                       ST8W0250
      MMOST = 10                                                        ST8W0260
      KORE = 10000                                                      ST8W0270
      N = NEWMAX                                                        ST8W0280
      MRHS = NCASES                                                     ST8W0290
      N1 = 2                                                            ST8W0300
```

```
      NO = 3                                                            ST8W0310
      NAT = 4                                                           ST8W0320
      MM = 10                                                           ST8W0330
      LTAPE = 11                                                        ST8W0340
C  INVERT THE MATRIX AND SOLVE FOR THE FUNDAMENTAL CASE RIGHT SIDES     ST8W0350
C  THE GAMMA ARRAY HAS BEEN DEFINED ELSEWHERE AND CONTAINS THE RHS      ST8W0360
      CALL MATRIX(WKAREA,GAMMA,N,MRHS,KORE,NMOST,MMOST,                 ST8W0370
     1              N1,MM,NO,NAT,LTAPE,IR)                              ST8W0380
      IF(IR .EQ. 2) GO TO 50                                            ST8W0390
C  THE SOLUTIONS HAVE BEEN STORED IN THE GAMMA ARRAY                    ST8W0400
C  PRINT THE SOLUTIONS AND CHECK BY BACK SUBSTITUTION                   ST8W0410
      IF(IPRINT .GE. 0) GO TO 70                                        ST8W0420
      DO 40  N = 1,NCASES                                               ST8W0430
      WRITE(6, 30 ) N,(GAMMA(J,N),J=1,NEWMAX)                           ST8W0440
   30 FORMAT(1H0,50X,16HFUNDAMENTAL CASE,I4,60(/1X,10E13.5))            ST8W0450
   40 CONTINUE                                                          ST8W0460
      CALL BAKSUB(TRANS,SUMMER,ISIZE,IO,ISTAB,IGND)                     ST8W0470
      GO TO 70                                                          ST8W0480
C  AN ERROR HAS OCCURED                                                 ST8W0490
   50 WRITE(6, 60 )                                                     ST8W0500
   60 FORMAT(1H0/// 40X,39HNOT ENOUGH ROOM WAS RESERVED FOR MATRIX)     ST8W0510
   70 CONTINUE                                                          ST8W0520
      END                                                               ST8W0530
```

```
*DECK PREPW                                                             PRPW0010
      SUBROUTINE PREPW(TRANS,ISIZE,NEWMAX,IO,ISTAB)                     PRPW0020
C                                                                       PRPW0030
C  THIS SUBROUTINE PREPARES THE JET-WING MATRIX FOR QUASI-INVERSION     PRPW0040
C                                                                       PRPW0050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            PRPW0060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) PRPW0070
      COMMON/SOLV1/B(400,10)                                            PRPW0080
      DIMENSION TRANS(ISIZE)                                            PRPW0090
C                                                                       PRPW0100
      IREAD = 1                                                         PRPW0110
      IF(ISTAB .NE. 0) IREAD = IREAD + 1                                PRPW0120
      REWIND 2                                                          PRPW0130
      NWT1 = NWT + 1                                                    PRPW0140
      IREAD = IREAD - 2                                                 PRPW0150
      IR2 = 2 * NMAX - 1                                                PRPW0160
      IF(ISTAB .NE. 0) IR2 = IR2 + 1                                    PRPW0170
C  READ THE MATRIX COEFFICIENTS                                         PRPW0180
      DO 30  I = 1,NEWMAX                                               PRPW0190
      IF(I .EQ. NWT1) IREAD = IR2                                       PRPW0200
      IREAD = IREAD + 2                                                 PRPW0210
      CALL READMS(IO,TRANS,ISIZE,IREAD)                                 PRPW0220
C  THE MATRIX ROW HAS NOW BEEN ASSEMBLED AND FILLS THE TRANS ARRAY.     PRPW0230
C  WRITE THE ROW ON UNIT 2 FOR INPUT TO MATRIX                          PRPW0240
   20 WRITE(2) TRANS                                                    PRPW0250
   30 CONTINUE                                                          PRPW0260
C  THE SYSTEM OF LINEAR EQUATIONS IS NOW READY FOR SOLUTION             PRPW0270
      RETURN                                                            PRPW0280
      END                                                               PRPW0290
```

```
*DECK STG8T                                                             ST8T0010
```

```
      OVERLAY(OVLY,10,2)
      PROGRAM STG8T
C
C  THIS SUBROUTINE CONTROLS THE QUASI-INVERSION OF THE TAIL MATRIX
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      COMMON /SOLVI/ WKAREA(10000)
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES
      COMMON /TAIL4/ GAMMA(100,5)
      DIMENSION TRANS(400)
      EQUIVALENCE (WKAREA(1),TRANS(1))
C
C  PREPARE THE MATRIX INPUT TAPE UNIT 2
      ISIZE = NTOTAL
      IO = 8
      CALL PREPT(TRANS,ISIZE,NTOTAL,IO,ISTAB)
      IF(IPRINT .LT. 0) WRITE(6, 20 )
   20 FORMAT(1H1,50X,19HTAIL GAMMA SOLUTION)
C  DEFINE THE REQUIRED PARAMETERS
      NMOST = 100
      MMOST = 5
      KORE = 10000
      N = NTOTAL
      MRHS = NSIDE
      NI = 2
      NO = 3
      NAT = 4
      MM = 12
      LTAPE = 13
C  INVERT THE MATRIX AND SOLVE FOR THE LINEAR CASE RIGHT SIDES
C  THE GAMMA ARRAY HAS BEEN DEFINED ELSEWHERE AND CONTAINS THE RHS
      CALL MATRIX(WKAREA,GAMMA,N,MRHS,KORE,NMOST,MMOST,
     1              NI,MM,NO,NAT,LTAPE,IR)
C  THE SOLUTIONS HAVE BEEN STORED IN THE GAMMA ARRAY. PRINT IF REQUIRED
      IF(IPRINT .GE. 0) GO TO 50
      DO 40  N = 1,NSIDE
      WRITE(6, 30 ) N,(GAMMA(J,N),J=1,NTOTAL)
   30 FORMAT(1H0,50X,16HTAIL LINEAR CASE, I4, 60(/1X,10E13.5))
   40 CONTINUE
   50 CONTINUE
      END
```

```
*DECK PREPT
      SUBROUTINE PREPT(TRANS,ISIZE,NTOTAL,IO,ISTAB)
C
C  THIS SUBROUTINE PREPARES THE TAIL MATRIX FOR QUASI-INVERSION
C
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)
      DIMENSION TRANS(ISIZE)
C
C  FIND THE RECORD TO BEGIN READING
      IREAD = 2 * (NMAX + NJT) - 2
      IF(ISTAB .EQ. 1) IREAD = IREAD + 1
```

```
C  READ THE MATRIX COEFFICIENTS AND WRITE THEM ON SCRATCH UNIT 2
      REWIND 2
      DO 20 I = 1,NTOTAL
      IREAD = IREAD + 3
      CALL READMS(IO,TRANS,ISIZE,IREAD)
      WRITE(2) TRANS
   20 CONTINUE
      RETURN
      END
```

```
*DECK STAGE9
      OVERLAY(OVLY,11,0)
      PROGRAM STAGE9
C
C  THIS SUBROUTINE CONTROLS THE MATRIX BACK-SOLUTION FOR NEW RIGHT
C  SIDES FOR THE JET-WING OR TAIL
C
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON /SOLV8/ WKAREA(10000)
C
C  JET-WING SOLUTIONS
      IF(LOGIC .EQ. 1) CALL OVERLAY(4HOVLY,9,1)
C  TAIL SOLUTIONS
      IF(LOGIC .EQ. 2) CALL OVERLAY(4HOVLY,9,2)
      CONTINUE
      END
```

```
*DECK NEWSOL
      SUBROUTINE NEWSOL(A,RHS,ND,MRHS,KD,NMOST,MMOST,
     1              NI,MM,NO,NAT,LTAPE,IR)
C
C                    DIRECT MATRIX BACK-SOLUTION
C*** ***LTAPE IS THE TAPE THE L(I,J) MATRIX WILL BE PUT ON
C*** ***NATAPE IS A SCRATCH TAPE
C*** ***THE TRIANGULAR MATRIX EXCEPT FOR THE LAST K ROWS WILL BE KEPT ON
C*** ***TAPE MM
C*** ***THE LAST K ROWS OF THE TRIANGULAR MATRIX WILL BE PUT ON THE
C*** ***LTAPE BEHIND THE RHS MATRIX
C
      DIMENSION A(KD),RHS(NMOST,MMOST)
      LOGICAL JPASS1,LASTRS,LAST
C
      IR = 2
      REWIND LTAPE
      NATAPE = NAT
      REWIND NATAPE
      N = ND
      KORE = KD
      IF(MRHS .NE. 0) GO TO 30
      WRITE(6, 20 )
   20 FORMAT(1H0//16X,41H***  NO NEW RIGHT SIDES WERE SUPPLIED FOR,
     1      47H SOLUTION BY EXISTING QUASI-INVERSE MATRIX  ***)
      IR = 1
      RETURN
```

```
   30 M = KORE/N -1
      MMAX = MINO(MRHS,M)
      NPM = N + MMAX
      IF( (3*NPM) .GT. KORE ) RETURN
      M = 0
      NPM = N
      MT = MM
      REWIND MT
      NIN = NI
      REWIND NIN
      NOUT = NO
      REWIND NOUT
      NN = N
      NEL = NPM
C - - CALCULATE THE MAXIMUM NO. OF ROWS, 'K'
C*** ***CALCULATE THE NUMBER OF COLUMNS TO BRING OFF OF THE RHS TAPE
      MTOTAL = 0
      M = MMAX
C*** ***MTOTALIS THE TOTAL NUMBER OF RHS COLUMNS ALREADY BROUGHT IN
   40 MTOTAL = MTOTAL + M
      LASTRS = MTOTAL.GE. MRHS
      MTOTAL = MTOTAL - M
      IF (LASTRS)M = MRHS - MTOTAL
      MP1 = M + 1
      MM1 = M - 1
      MTOTAL = MTOTAL + M
C*** ***BRING IN M COLUMNS OF RHS
      KINIT = KORE - (M*N)
      IINIT = KINIT
      NBEG = KINIT
      DO 50  J =1,M
      DO 50  I = 1,N
      NBEG = NBEG + 1
   50 A(NBEG) = RHS(I,J)
C*** ***BRING IN L(I,J) MATRIX AND APPLY IT TO RHS
      NBEG = 1 + KINIT
      NEND = 1 + (M-1) * N + KINIT
      KSUM = 0
      NLCNT = 0
C*** ***DO TRIANGULAR SECTION OF LMATRIX
   60 READ (LTAPE)K
      NLCNT = NLCNT + 1
C*** ***KSUM IS THE TOTAL NUMBER OF L ROWS THAT WILL
C*** ***BE READ AFTER THIS TRIANGULAR SECTION IS FINISHED
      KSUM = KSUM + K
      KM1 = K - 1
C***    ***NOTE THAT KM1 CAN'T BE 0 SINCE K CAN'T BE 1 AND STILL HAVE SOM
C***    ***ON THE LTAPE
      DO 90   I = 1,KM1
      NBEG = NBEG + 1
      NEND = NEND + 1
C*** ***READ 1 ROW OF L(I,J) FROM LTAPE---K-1 TIMES---EACH TIME
C*** ***STARTING WITH L(I)
      CALL GETT(LTAPE,1,I,A,1,AA2)
      JCNT = -1
C*** ***REDUCE THE RHS BY GOING ACROSS A SOLUTION ROW (WHICH
C*** ***ARE NOT IN CONSECUTIVE ORDER, BUT A(I), A(N+I), A(2N+I) ETC.)
      DO 80  NPP = NBEG,NEND,N
      JCNT = JCNT + 1
      SUM = 0.0
```

```
      NROW = KINIT + ( JCNT * N )
      DO 70  NN=1,I
      NROW = NROW + 1
   70 SUM = SUM + ( A(NN)*A(NROW))
   80 A(NPP) = A(NPP) - SUM
   90 CONTINUE
      IF (KSUM .EQ. N)GO TO 130
C*** ***KSUM = N IF YOU HAVE READ ENTIRE LMATRIX AND
C*** ***THERE IS NO CONSTANT SECTION LEFT
      NTBEG = NBEG
      NTEND = NEND
      KSUMP1 = KSUM + 1
C***  ***READ REST OF LROWS 1 ROW AT A TIME FOR CONSTANT SECTION
      DO 120 I=KSUMP1,N
      NTBEG = NTBEG + 1
      NTEND = NTEND + 1
      CALL GETT(LTAPE,1,K,A,1,AA2)
      JCNT = -1
C*** ***PARTIALLY  REDUCE A RHS ACROSS A RHS ROW BY APPLYING K NUMBER
C*** ***OF L(I,J) S
      DO 110 NPP = NTBEG,NTEND,N
      JCNT = JCNT + 1
      SUM = 0.0
      NROW = KINIT + ( JCNT * N )
      DO 100 NN = 1,K
      NROW = NROW + 1
  100 SUM = SUM + ( A(NN) * A(NROW) )
  110 A(NPP) = A(NPP) - SUM
  120 CONTINUE
      NBEG = NBEG + 1
      NEND = NEND + 1
C*** ***KINIT IS HOW FAR DOWN A COLUMN OF RHS TO START MULTIPLYING BY
C*** ***L(I,J) AT EACH PASS THROUGH
      KINIT = KINIT + K
C*** ***IF KSUMP1 = N THERE ARE NO MORE L(I,J)'S LEFT
      IF(KSUMP1 .LT. N)GO TO 60
C*** ***WRITE OUT ALL BUT LAST K ROWS OF RHS IN ROW ORDER ON NATAPE
  130 B = 4*M + 3
      C = -2 * KORE
      K = (   -B  +  SQRT( B**2  -  4*C )  )/2
      IF(K .GT. ND) K = ND
      KF = K
      KM1 = K - 1
      KLEFT = N - KF + IINIT
      INITP1 = IINIT + 1
      NEND = (M-1)*N + IINIT
      DO 140 NPP = INITP1,KLEFT
      NEND = NEND + 1
  140 WRITE(NATAPE) ( A(J),J=NPP,NEND,N)
      REWIND NATAPE
C*** ***JPASS1 IS TRUE ON 1ST PASS THRU BACK SOLUTION
      JPASS1 = .TRUE.
C***   ***PUT REMAINING RHS IN CONTIGUOUS LOCATIONS BY COLUMNS
C***  FROM KORE - (M * KF) + 1  TO  KORE
      NNEW = KORE - KF + 1
C***   ***IF M = 1, THE ELTS OF THE 1 RHS COLUMN ARE ALREADY IN CONTIGUO
C***   ***LOCATIONS
      IF( M .EQ. 1 )GO TO 160
      DO 150 I = 1,MM1
      NOLD = KORE - (I*N) + 1
```

```
      DO 150 J = 1,KF                                                   NEWS1480
      NNEW = NNEW - 1                                                   NEWS1490
      NOLD = NOLD - 1                                                   NEWS1500
      A(NNEW) = A(NOLD)                                                 NEWS1510
  150 CONTINUE                                                          NEWS1520
  160 CONTINUE                                                          NEWS1530
C***   ***NOW NNEW = KORE - (M*KF) + 1                                  NEWS1540
C***   ***NOW NOLD = KORE - (M - 1) * N + 1 - KF                        NEWS1550
C  **   SKIP 1ST PART OF TRAPEZOIDAL MATRIX ON LTAPE                    NEWS1560
C  **   READ IN LAST K ROWS OF TRAPEZOIDAL MATRIX AND                   NEWS1570
C*** ***ATTATCH RHS TO IT SO THAT EVERYTHING IS IN CONSECUTIVE ORDER    NEWS1580
      NREMAN = NO - K                                                   NEWS1590
      IF( NREMAN .EQ. 0 )GO TO 180                                      NEWS1600
      DO 170 I=1,NREMAN                                                 NEWS1610
  170 READ(LTAPE) IDUMMY                                                NEWS1620
  180 NEND = 0                                                          NEWS1630
      KCNT = K                                                          NEWS1640
      NNEW = NNEW - 1                                                   NEWS1650
C***   ***NOTE THAT K = KF WHICH IS ALREADY KNOWN IN CORE               NEWS1660
      DO 190 JCNT = 1,K                                                 NEWS1670
      NBEG = NEND + 1                                                   NEWS1680
      CALL GETT(LTAPE,4,KCNT,A(NBEG),1,AA2)                             NEWS1690
      KCNT = KCNT - 1                                                   NEWS1700
      NEND = NBEG + KCNT                                                NEWS1710
      NNEW = NNEW + 1                                                   NEWS1720
      KEND = (MM1 * KF) + NNEW                                          NEWS1730
      DO 190 NPP=NNEW,KEND,KF                                           NEWS1740
      NEND = NEND + 1                                                   NEWS1750
  190 A(NEND) = A(NPP)                                                  NEWS1760
      REWIND LTAPE                                                      NEWS1770
C  **  SKIP OVER L MATRIX ON LTAPE TO GET TO TRAPEZOIDAL MATRIX         NEWS1780
      KRED = 0                                                          NEWS1790
      DO 200 I=1,NLCNT                                                  NEWS1800
      READ(LTAPE)KREAD                                                  NEWS1810
      KRED = KRED + KREAD                                               NEWS1820
      KREAD = KREAD + ( N - KRED - 1)                                   NEWS1830
      DO 200 LREAD = 1,KREAD                                            NEWS1840
  200 READ(LTAPE)                                                       NEWS1850
C - - THERE, NOW WE CAN START THE BACK-SOLUTION                         NEWS1860
C * * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(N1)     NEWS1870
C***   ***NL IS THE LAST SUBSCRIPT + 1 OF THE TRAPEZOIDAL A MATRIX THAT NEWS1880
C***   ***CORE                                                          NEWS1890
      NL = NEND + 1                                                     NEWS1900
      NREM = N                                                          NEWS1910
      NPM = N + M                                                       NEWS1920
      NEL = NPM                                                         NEWS1930
      MP1 = M + 1                                                       NEWS1940
      LAST = K .EQ. N                                                   NEWS1950
      NPASS = 0                                                         NEWS1960
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO 'K' ROWS                   NEWS1970
  210 KM1 = K - 1                                                       NEWS1980
      KP1 = K + 1                                                       NEWS1990
      NS = NL - MP1                                                     NEWS2000
      NPASS = NPASS + 1                                                 NEWS2010
      DO 240 MN = 1, M                                                  NEWS2020
      NF = NS + MN                                                      NEWS2030
      A(NF) = A(NF) / A(NS)                                             NEWS2040
      NT = NS                                                           NEWS2050
      IF (KM1 .EQ. 0) GO  TO 240                                        NEWS2060
      DO 230 IB = 1, KM1                                                NEWS2070
```

```
      NF = NF - IB - M                                                  NEWS2080
      N1 = NT - MP1 - IB                                                NEWS2090
      SUM = 0.0                                                         NEWS2100
      NP = NF                                                           NEWS2110
      N2 = MP1 + IB                                                     NEWS2120
      DO 220 IO = 1, IB                                                 NEWS2130
      NN = NT + IO                                                      NEWS2140
      NP = NP + N2 - IO                                                 NEWS2150
  220 SUM = SUM + A(NN) * A(NP)                                         NEWS2160
  230 A(NF) = (A(NF) - SUM) / A(NT)                                     NEWS2170
  240 CONTINUE                                                          NEWS2180
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(N1)      NEWS2190
      N1 = KORE + 1                                                     NEWS2200
      DO 260 NN = 1, K                                                  NEWS2210
      DO 250 MN = 1, M                                                  NEWS2220
      NL = NL - 1                                                       NEWS2230
      N1 = N1 - 1                                                       NEWS2240
  250 A(N1) = A(NL)                                                     NEWS2250
  260 NL = NL - NN                                                      NEWS2260
C - - WRITE THE SOLUTIONS ON TAPE                                       NEWS2270
      WRITE (NIN) K                                                     NEWS2280
      NS = N1 - 1                                                       NEWS2290
      DO 270 MN = 1, M                                                  NEWS2300
      NT = NS + MN                                                      NEWS2310
  270 WRITE ( NIN ) (A(IO), IO = NT, KORE, M)                           NEWS2320
C - - TEST IF THIS IS THE LAST PASS                                     NEWS2330
      IF (LAST) GO TO 380                                               NEWS2340
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF NEWS2350
C     THE SOLUTIONS OBTAINED SO FAR  (EQ 21)                            NEWS2360
C * * NOTE..LOCATIONS A(1) TO A(N1-1) ARE NOW FREE TO USE               NEWS2370
C - - CALCULATE THE NEXT VALUES OF 'NEL' AND 'NREM'                     NEWS2380
      NELOLD = NEL                                                      NEWS2390
      KOLD = K                                                          NEWS2400
      NEL = NEL - K                                                     NEWS2410
      NREM = NREM - K                                                   NEWS2420
      NROW = NREM - K + 1                                               NEWS2430
      IF (K .LT. NREM) GO TO 280                                        NEWS2440
      LAST = .TRUE.                                                     NEWS2450
      NROW = 1                                                          NEWS2460
      K = NREM                                                          NEWS2470
  280 NS = 1                                                            NEWS2480
      NT = NELOLD + 1                                                   NEWS2490
C - - READ IN THE ROWS TO BE MODIFIED                                   NEWS2500
      DO 360 IB = 1, NREM                                               NEWS2510
      NT = NT - 1                                                       NEWS2520
      IF (IB .LE. NROW) GO TO 290                                       NEWS2530
      NS = NS + NN                                                      NEWS2540
      NT = NT + NN                                                      NEWS2550
  290 IF (.NOT. JPASS1)GO TO 300                                        NEWS2560
      NBEG = NT - M + 1                                                 NEWS2570
C*** ***READ RHS FROM NATAPE                                            NEWS2580
      CALL GETT(NATAPE,1,M,A(NBEG),1,AA2)                               NEWS2590
      NT = NT - M                                                       NEWS2600
      CALL GETT(LTAPE,2,NN,A(NS),1,AA2)                                 NEWS2610
      NT = NT + M                                                       NEWS2620
      NN = NN + M                                                       NEWS2630
      GO TO 310                                                         NEWS2640
  300 CALL GETT(MT,2,NN,A(NS),1,AA2)                                    NEWS2650
  310 NP = N1 - 1                                                       NEWS2660
      NF = NT - M - KM1                                                 NEWS2670
```

```
      NN = NN - KOLD                                                    NEWS2680
      DO 330 MN = 1, M                                                  NEWS2690
      N2 = NF                                                           NEWS2700
      NA = NP + MN                                                      NEWS2710
      NB = NA                                                           NEWS2720
      SUM = 0.0                                                         NEWS2730
      DO 320 IO = 1, KOLD                                               NEWS2740
      SUM = SUM + A(N2) * A(NA)                                         NEWS2750
      N2 = N2 + 1                                                       NEWS2760
  320 NA = NA + M                                                       NEWS2770
      N2 = N2 + MN - 1                                                  NEWS2780
  330 A(N2) = A(N2) - SUM                                               NEWS2790
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW                NEWS2800
      NL = NT - M + 1                                                   NEWS2810
      IF (IB .GE. NROW) GO TO 340                                       NEWS2820
      NF = NL - KP1                                                     NEWS2830
      NN1 = NF - NS + 1                                                 NEWS2840
      NN2 = NT - NL + 1                                                 NEWS2850
      CALL SAVE(NOUT,4,NN,NN1,A(NS),NN2,A(NL))                          NEWS2860
      GO TO 360                                                         NEWS2870
  340 NF = NL - KOLD                                                    NEWS2880
      DO 350 MN = NL, NT                                                NEWS2890
      A(NF) = A(MN)                                                     NEWS2900
  350 NF = NF + 1                                                       NEWS2910
  360 CONTINUE                                                          NEWS2920
      IF( .NOT. JPASS1 ) GO TO 370                                      NEWS2930
      JPASS1= .FALSE.                                                   NEWS2940
      REWIND NATAPE                                                     NEWS2950
  370 REWIND MT                                                         NEWS2960
      REWIND NOUT                                                       NEWS2970
C - - SWITCH THE TAPES                                                  NEWS2980
      NT = MT                                                           NEWS2990
      MT = NOUT                                                         NEWS3000
      NOUT = NT                                                         NEWS3010
C - - LOOP BACK THRU THE SOLUTION                                       NEWS3020
      NL = NF                                                           NEWS3030
      GO TO 210                                                         NEWS3040
C - - START TO WRAP IT UP                                               NEWS3050
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KORE) ARE FREE     NEWS3060
C - - READ IN THE SOLUTIONS AND STORE THEM IN THEIR FINAL CORE LOCATIONSNEWS3070
  380 REWIND NIN                                                        NEWS3080
      N1 = N + 1                                                        NEWS3090
      DO 390 IB = 1, NPASS                                              NEWS3100
      READ (NIN) K                                                      NEWS3110
      N1 = N1 - K                                                       NEWS3120
      DO 390 IO = 1, M                                                  NEWS3130
  390 CALL GETT(NIN,1,K,RHS(N1,IO),1,AA2)                               NEWS3140
C  ---  REWIND ALL INPUT TAPES                                          NEWS3150
      REWIND NIN                                                        NEWS3160
      REWIND MT                                                         NEWS3170
      REWIND NOUT                                                       NEWS3180
      IF(LASTRS) GO TO 400                                              NEWS3190
C*** ***IF THERE ARE MORE RHS TO BE GOTTEN FROM RHS TAPE, SWITCH TAPES  NEWS3200
C*** ***BACK SO THAT  MT WILL CONAIN THE TRAPEZOIDAL MATRIX             NEWS3210
C*** ***NATAPE WILL HAVE NOTHING USEFUL ON IT.                          NEWS3220
      NTEMP = NATAPE                                                    NEWS3230
      NATAPE = MT                                                       NEWS3240
      MT = NTEMP                                                        NEWS3250
      REWIND NATAPE                                                     NEWS3260
      REWIND LTAPE                                                      NEWS3270
```

```
      GO TO 40                                                          NEWS3280
  400 REWIND LTAPE                                                      NEWS3290
C   ***   REWIND ALL FILES EXCEPT THE OUTPUT FILE   NW                  NEWS3300
      REWIND  NI                                                        NEWS3310
      REWIND  MM                                                        NEWS3320
      REWIND  NO                                                        NEWS3330
      REWIND  NAT                                                       NEWS3340
      IR = 1                                                            NEWS3350
      RETURN                                                            NEWS3360
      END                                                               NEWS3370
```

```
*DECK STG9W                                                             ST9W0010
      OVERLAY(OVLY,11,1)                                                ST9W0020
      PROGRAM STG9W                                                     ST9W0030
C                                                                       ST9W0040
C  THIS SUBROUTINE CONTROLS THE BACK-SOLUTION OF THE JET-WING MATRIX    ST9W0050
C  FOR NEW RIGHT SIDES                                                  ST9W0060
C                                                                       ST9W0070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                ST9W0080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            ST9W0090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) ST9W0100
      COMMON /SOLVB/ WKAREA(10000)                                      ST9W0110
      COMMON /FFLD5/ GAMJ(400,5)                                        ST9W0120
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     ST9W0130
     1              NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,   ST9W0140
     2              ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES             ST9W0150
      DIMENSION RHS1(400),RHS2(400),RHS3(400),RHS4(400),RHS5(400)       ST9W0160
      DIMENSION REMAIN(200)                                             ST9W0170
      EQUIVALENCE (GAMJ(1,1),RHS1(1)),(GAMJ(1,2),RHS2(1)),              ST9W0180
     1            (GAMJ(1,3),RHS3(1)),(GAMJ(1,4),RHS4(1)),              ST9W0190
     2            (GAMJ(1,5),RHS5(1))                                   ST9W0200
      EQUIVALENCE (WKAREA(1),REMAIN(1))                                 ST9W0210
C                                                                       ST9W0220
C  DEFINE THE REQUIRED PARAMETERS                                       ST9W0230
      IREAD = 2*(NMAX+NJT) + 3*NTOTAL + 3                               ST9W0240
      NMOST = 400                                                       ST9W0250
      MMOST = 5                                                         ST9W0260
      KORE = 10000                                                      ST9W0270
      N = NEWMAX                                                        ST9W0280
      MRHS = 5                                                          ST9W0290
      NI = 2                                                            ST9W0300
      NO = 3                                                            ST9W0310
      NAT = 4                                                           ST9W0320
      MM = 10                                                           ST9W0330
      LTAPE = 11                                                        ST9W0340
C  DEFINE THE NEW RIGHT SIDE COLUMNS FROM THE JET INDUCED VELOCITIES    ST9W0350
C  STORED ON UNIT 8                                                     ST9W0360
      CALL READMS(8,RHS1,400,IREAD)                                     ST9W0370
      IREAD = IREAD + 3                                                 ST9W0380
      CALL READMS(8,RHS2,400,IREAD)                                     ST9W0390
      IREAD = IREAD + 3                                                 ST9W0400
      CALL READMS(8,RHS3,400,IREAD)                                     ST9W0410
      IREAD = IREAD + 3                                                 ST9W0420
      CALL READMS(8,RHS4,400,IREAD)                                     ST9W0430
      IREAD = IREAD + 3                                                 ST9W0440
      CALL READMS(8,RHS5,400,IREAD)                                     ST9W0450
C  SOLVE FOR THE NEW RIGHT SIDES                                        ST9W0460
```

```
      CALL NEWSOL(WKAREA,GAMJ,N,MRHS,KORE,NMOST,MMOST,                   ST9W0470
     1            NI,MM,NO,NAT,LTAPE,IR)                                 ST9W0480
      IF(IR .EQ. 2) GO TO 50                                             ST9W0490
C  THE SOLUTIONS HAVE BEEN STORED IN THE GAMJ ARRAY. PRINT IF REQUIRED   ST9W0500
      IF(IPRINT .GE. 0) GO TO 70                                         ST9W0510
      WRITE(6, 20 )                                                      ST9W0520
   20 FORMAT(1H1,36X,46HWING GAMMA SOLUTION FOR CIRCULAR JET INFLUENCE)  ST9W0530
      DO 40  N = 1,5                                                     ST9W0540
      ALPHA = 5 * (N-1)                                                  ST9W0550
      WRITE(6, 30 ) ALPHA,(GAMJ(J,N),J=1,NEWMAX)                         ST9W0560
   30 FORMAT(1H0,43X,20HSOLUTION FOR ALPHA =,F5.1,8H DEGREES,            ST9W0570
     1          60(/1X,10E13.5))                                         ST9W0580
   40 CONTINUE                                                           ST9W0590
      GO TO 70                                                           ST9W0600
C  AN ERROR HAS OCCURED                                                  ST9W0610
   50 WRITE(6, 60 )                                                      ST9W0620
   60 FORMAT(1H0/// 40X,39HNOT ENOUGH ROOM WAS RESERVED FOR NEWSOL)      ST9W0630
   70 CONTINUE                                                           ST9W0640
      END                                                                ST9W0650


*DECK STG9T                                                              ST9T0010
      OVERLAY(OVLY,11,2)                                                 ST9T0020
      PROGRAM STG9T                                                      ST9T0030
C                                                                        ST9T0040
C  THIS SUBROUTINE CONTROLS THE BACK-SOLUTION OF THE TAIL MATRIX         ST9T0050
C  FOR NEW RIGHT SIDES                                                   ST9T0060
C                                                                        ST9T0070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                 ST9T0080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE             ST9T0090
      COMMON /SOLVB/ WKAREA(10000)                                       ST9T0100
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,      ST9T0110
     1             NMAXH,NMAXV,NTOTAL,NTPMP,NSIDE,NROWH,NROWV,ICAMH,     ST9T0120
     2             ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES               ST9T0130
      COMMON /TAIL4/ B(100,5)                                            ST9T0140
C                                                                        ST9T0150
C  DEFINE THE PARAMETERS                                                 ST9T0160
      NMOST = 100                                                        ST9T0170
      MMOST = 5                                                          ST9T0180
      KORE = 10000                                                       ST9T0190
      N = NMAXH + NMAXV                                                  ST9T0200
      MRHS = 5                                                           ST9T0210
      NI = 2                                                             ST9T0220
      NO = 3                                                             ST9T0230
      NAT = 4                                                            ST9T0240
      MM = 12                                                            ST9T0250
   20 LTAPE = 13                                                         ST9T0260
C  SOLVE THE NEW RIGHT SIDES                                             ST9T0270
      CALL NEWSOL(WKAREA,B,N,MRHS,KORE,NMOST,MMOST,                      ST9T0280
     1            NI,MM,NO,NAT,LTAPE,IR)                                 ST9T0290
C  THE SOLUTIONS HAVE BEEN STORED IN THE B ARRAY.  PRINT IF REQUIRED.    ST9T0300
      IF(IPRINT .GE. 0) GO TO 60                                         ST9T0310
      WRITE(6, 30 )                                                      ST9T0320
   30 FORMAT(1H1,37X,44HTAIL GAMMA SOLUTION IN NON-LINEAR FLOW FIELD)    ST9T0330
      DO 50  N = 1,5                                                     ST9T0340
      ALPHA = 5 * (N-1)                                                  ST9T0350
      WRITE(6, 40 ) ALPHA,(B(J,N),J=1,NTOTAL)                            ST9T0360
   40 FORMAT(1H0,42X,20HSOLUTION FOR ALPHA =,F5.1,8H DEGREES,            ST9T0370
     1          60(/1X,10E13.5))                                         ST9T0380
   50 CONTINUE                                                           ST9T0390
   60 CONTINUE                                                           ST9T0400
      END                                                                ST9T0410


*DECK STAG10                                                             ST100020
      OVERLAY(OVLY,12,0)                                                 ST100010
      PROGRAM STAG10                                                     ST100030
C  THIS SUBROUTINE CONTROLS EXECUTION OF THE APPROXIMATE SLENDER BODY    ST100040
C  FUSELAGE ANALYSIS.                                                    ST100050
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),     ST100060
     1             RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                 ST100070
     2             V(40,7),W(40,7)                                       ST100080
      CALL FUSELG                                                        ST100090
      CALL AXIAL                                                         ST100100
      DO 30 K = 1,NSEG                                                   ST100110
      N = NF(K)                                                          ST100120
      IF(ISECT .EQ. 2)GO TO 20                                           ST100130
      CALL ANGLES(K,N)                                                   ST100140
      CALL NEUMAN(K,N)                                                   ST100150
   20 CALL CPRES(K,N)                                                    ST100160
   30 CALL SUMER1(K,N)                                                   ST100170
      CALL SUMER2                                                        ST100180
      CONTINUE                                                           ST100190
      END                                                                ST100200


*DECK FUSELG                                                             FSLG0010
      SUBROUTINE FUSELG                                                  FSLG0020
C                                                                        FSLG0030
C  THIS SUBROUTINE CALCULATES THE FUSELAGE GEOMETRY FROM THE INPUT DATA  FSLG0040
C                                                                        FSLG0050
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),     FSLG0060
     1             RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                 FSLG0070
     2             V(40,7),W(40,7)                                       FSLG0080
      COMMON/FUSLG2/NCF,DCL(40),X(40),Z(40),AMEAN(40),                   FSLG0090
     1             AMEANK(40),RP(40),AP(40),AL(7),SA(7),CA(7),           FSLG0100
     2             GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),      FSLG0110
     3             SB(20),CB(20),DUMMY(21),D1(20),D2(20),                FSLG0120
     4             YFWD(21,40),YAFT(21,40),ZFWD(21,40),ZAFT(21,40)       FSLG0130
      COMMON/FUSLG3/URA(40),UKA(40),VA(20),WA(20),VS(20,7),WS(20,7),     FSLG0140
     1             US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),           FSLG0150
     2             CM(40,7),CN(40,7)                                     FSLG0160
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)  FSLG0170
      COMMON/SPIRIT/NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                  FSLG0180
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HX,HH,VL,VH,      FSLG0190
     1            NMAXH,NMAXV,NTOTAL,NTPMP,NSIDE,NROWH,NROWV,ICAMH,      FSLG0200
     2            ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES                FSLG0210
      COMMON/JOHN/  AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     FSLG0220
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE             FSLG0230
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI,IGRFSLG0240
      DIMENSION Y(21,40),Z(21,40)                                        FSLG0250
      DIMENSION YF(21),ZF(21),YA(21),ZA(21)                              FSLG0260
      DIMENSION A(84)                                                    FSLG0270
      EQUIVALENCE (A(1),YF(1)),(A(22),ZF(1)),(A(43),YA(1)),(A(64),ZA(1))FSLG0280
      EQUIVALENCE(Y(1,1),YAFT(1,1)),(Z(1,1),ZAFT(1,1))                   FSLG0290
      LOGICAL SECT,PASS                                                  FSLG0300
      DIST(A,B) = SQRT(A**2 + B**2)                                      FSLG0310
      SECT = ISECT .EQ. 1                                                FSLG0320
      PASS = (NEWCMU .LE. 1) .AND. (KCASE .LE. 1)                        FSLG0330
      IF(.NOT. PASS) LOGIC = 1                                           FSLG0340
      B2 = SPA / 2.0                                                     FSLG0540
C  CALCULATE CAMBERS                                                     FSLG0350
   20 DO 30  I = 1,NSEG                                                  FSLG0360
   30 DCL(I) =(ZCL(I+1) - ZCL(I)) / (XB(I+1) - XB(I))                    FSLG0370
      IF(.NOT. SECT) GO TO 60                                            FSLG0380
      IREAD = 2.0 * NMAX                                                 FSLG0390
      DO 50  I = 1,NSEG                                                  FSLG0400
      IREAD = IREAD + 1                                                  FSLG0410
      NI = NF(I) + 1                                                     FSLG0420
      CALL READMS(9,A,84,IREAD)                                          FSLG0430
      DO 40  J = 1,NI                                                    FSLG0440
      YFWD(J,I) = YF(J)                                                  FSLG0450
      ZFWD(J,I) = ZF(J)                                                  FSLG0460
      YAFT(J,I) = YA(J)                                                  FSLG0470
   40 ZAFT(J,I) = ZA(J)                                                  FSLG0480
   50 CONTINUE                                                           FSLG0490
C  NORMALIZE THE INPUTS                                                  FSLG0500
   60 IF(SECT) GO TO 70                                                  FSLG0510
      IF(.NOT. PASS) GO TO 160                                           FSLG0520
   70 IF(.NOT. PASS) GO TO 90                                            FSLG0530
      DO 80  I = 1,NSEG1                                                 FSLG0550
      XB(I) = XB(I) / B2                                                 FSLG0560
   80 ZCL(I) = ZCL(I) / B2                                               FSLG0570
      GO TO ( 90 , 120 , 140 ),ISECT                                     FSLG0580
   90 DO 110 K = 1,NSEG                                                  FSLG0590
      NI = NF(K) + 1                                                     FSLG0600
      DO 100 I = 1,NI                                                    FSLG0610
      YFWD(I,K) = YFWD(I,K) / B2                                         FSLG0620
      ZFWD(I,K) = ZFWD(I,K) / B2                                         FSLG0630
      YAFT(I,K) = YAFT(I,K) / B2                                         FSLG0640
  100 ZAFT(I,K) = ZAFT(I,K) / B2                                         FSLG0650
  110 CONTINUE                                                           FSLG0660
      GO TO 160                                                          FSLG0670
  120 DO 130 K = 1,NSEG1                                                 FSLG0680
      CSA(K) = CSA(K) / B2**2                                            FSLG0690
  130 RADIUS(K) = RADIUS(K) / B2                                         FSLG0700
      GO TO 160                                                          FSLG0710
  140 DO 150 K = 1,NSEG1                                                 FSLG0720
      AMAJ(K) = AMAJ(K) / B2                                             FSLG0730
  150 AMIN(K) = AMIN(K) / B2                                             FSLG0740
  160 NCF = 5                                                            FSLG0750
      ALPHA = -5.0                                                       FSLG0760
      DO 170 JA = 1,5                                                    FSLG0770
      ALPHA = ALPHA + 5.0                                                FSLG0780
      AL(JA) = ALPHA                                                     FSLG0790
      SA(JA) = SIN(ALPHA / 57.295779)                                    FSLG0800
  170 CA(JA) = COS(ALPHA / 57.295779)                                    FSLG0810
      IF(LOGIC .EQ. 1) GO TO 180                                         FSLG0820
      NCF = 7                                                            FSLG0830
      AL(6) = 0.0                                                        FSLG0840
      AL(7) = 0.0                                                        FSLG0850
      SA(6) = 0.0                                                        FSLG0860
      CA(6) = 1.0                                                        FSLG0870
      SA(7) = 0.0                                                        FSLG0880
      CA(7) = 1.0                                                        FSLG0890
  180 DO 300 I = 1,NSEG                                                  FSLG0900
      N = NF(I)                                                          FSLG0910
      NI = N + 1                                                         FSLG0920
      GO TO ( 260 , 190 , 210 ),ISECT                                    FSLG0930
C  CIRCULAR SECTIONS                                                     FSLG0940
  190 DT = PI / N                                                        FSLG0950
      IF(ISYMM .GT. 0)DT = 2.0 * PI / N                                  FSLG0960
      THETA = 0.0                                                        FSLG0970
      YFWD(1,I) = 0.0                                                    FSLG0980
      YAFT(1,I) = 0.0                                                    FSLG0990
      ZFWD(1,I) = -RADIUS(I)                                             FSLG1000
      ZAFT(1,I) = -RADIUS(I+1)                                           FSLG1010
      DO 200 J = 2,NI                                                    FSLG1020
      THETA = THETA + DT                                                 FSLG1030
      ST = SIN(THETA)                                                    FSLG1040
      CT = COS(THETA)                                                    FSLG1050
      YFWD(J,I) = RADIUS(I) * ST                                         FSLG1060
      ZFWD(J,I) = -RADIUS(I) * CT                                        FSLG1070
      YAFT(J,I) = RADIUS(I+1) * ST                                       FSLG1080
      ZAFT(J,I) = -RADIUS(I+1) * CT                                      FSLG1090
  200 CONTINUE                                                           FSLG1100
      GO TO 280                                                          FSLG1110
C  ELLIPTICAL SECTIONS                                                   FSLG1120
  210 DT = PI / N                                                        FSLG1130
      IF(ISYMM .GT. 0)DT = 2.0 * PI / N                                  FSLG1140
      THETA = 0.0                                                        FSLG1150
      YFWD(1,I) = 0.0                                                    FSLG1160
      YAFT(1,I) = 0.0                                                    FSLG1170
      ZFWD(1,I) = -AMAJ(I)                                               FSLG1180
      ZAFT(1,I) = -AMAJ(I+1)                                             FSLG1190
      DO 250 J = 2,NI                                                    FSLG1200
      THETA = THETA + DT                                                 FSLG1210
      CT = COS(THETA)                                                    FSLG1220
      IF(J .EQ. 1)GO TO 220                                              FSLG1230
      ZFWD(J,I) = -AMAJ(I) * CT                                          FSLG1240
      YFWD(J,I) = AMIN(I) * SQRT(1.0 - (ZFWD(J,I) / AMAJ(I))**2)         FSLG1250
      GO TO 230                                                          FSLG1260
  220 YFWD(J,I) = 0.0                                                    FSLG1270
      ZFWD(J,I) = 0.0                                                    FSLG1280
  230 IF(I .EQ. NSEG)GO TO 240                                           FSLG1290
      ZAFT(J,I) = -AMAJ(I+1) * CT                                        FSLG1300
      YAFT(J,I) = AMIN(I+1) * SQRT(1.0 - (ZAFT(J,I) / AMAJ(I+1))**2)     FSLG1310
      GO TO 250                                                          FSLG1320
  240 YAFT(J,I) = 0.0                                                    FSLG1330
      ZAFT(J,I) = 0.0                                                    FSLG1340
  250 CONTINUE                                                           FSLG1350
C  CALCULATE THE CROSS - SECTIONAL AREA AT THE FORWARD EDGE OF SECTION   FSLG1360
  260 CSA(I) = 0.0                                                       FSLG1370
      DO 270 J = 1,N                                                     FSLG1380
      CSA(I) = CSA(I) + 0.5 * ABS(YFWD(J,I) * ZFWD(J+1,I)                FSLG1390
     1             -YFWD(J+1,I) * ZFWD(J,I))                             FSLG1400
  270 CONTINUE                                                           FSLG1410
      IF(ISYMM .LE. 0)CSA(I) = 2.0 * CSA(I)                              FSLG1420
      RADIUS(I) = SQRT(CSA(I) / PI)                                      FSLG1430
C  CALCULATE THE SURFACE INCLINATION ANGLES AND THE ELEMENTAL AREAS      FSLG1440
  280 DO 290 J = 1,N                                                     FSLG1450
      DYF = YFWD(J+1,I) - YFWD(J,I)                                      FSLG1460
      DYA = YAFT(J+1,I) - YAFT(J,I)                                      FSLG1470
      DZF = ZFWD(J+1,I) - ZFWD(J,I)                                      FSLG1480
```

```
      DZA = ZAFT(J+1,I) - ZAFT(J,I)
      DS = 0.5 * (DIST(DYA,DZA) + DIST(DYF,DZF))
      DYF = 0.5 * (YFWD(J,I) + YFWD(J+1,I))
      DYA = 0.5 * (YAFT(J,I) + YAFT(J+1,I))
      DZF = 0.5 * (ZFWD(J,I) + ZFWD(J+1,I))
      DZA = 0.5 * (ZAFT(J,I) + ZAFT(J+1,I))
      RFP = DIST(DYF,DZF)
      RAP = DIST(DYA,DZA)
      DZF = DZF + ZCL(I)
      DZA = DZA + ZCL(I+1)
      RF = DIST(DYF,DZF)
      RA = DIST(DYA,DZA)
      DRP = RAP - RFP
      DR = RA - RF
      DX = XB(I+1) - XB(I)
      GAMMA(J,I) = ATAN(DR / DX)
      CHECK = 1.0
      IF(DR .NE. 0.0)CHECK = DRP / DR
      IF(CHECK .LT. 0.0)GAMMA(J,I) = -GAMMA(J,I)
      S(J,I) = DS + DIST(DR,DX)
  290 CONTINUE
C  CALCULATE THE COORDINATES AT THE PANEL CENTERS
      X(I) = 0.5 * (XB(I) + XB(I+1))
      ZC(I) = 0.5 * (ZCL(I) + ZCL(I+1))
  300 CONTINUE
      CSA(NSEG1) = 0.0
      RADIUS(NSEG1) = 0.0
      DO 320 I = 1,NSEG
      NI = NF(I) + 1
      DO 310 J = 1,NI
      Y(J,I) = 0.5 * (YAFT(J,I) + YFWD(J,I))
      Z(J,I) = 0.5 * (ZAFT(J,I) + ZFWD(J,I)) + ZC(I)
  310 CONTINUE
  320 CONTINUE
      RETURN
      END
```

```
*DECK AXIAL
      SUBROUTINE AXIAL
C
C  THIS SUBROUTINE CALCULATES THE VELOCITY POTENTIAL FOR THE AXIAL
C  FLOW ON THE EQUIVALENT BODY OF REVOLUTION
C
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2              V(40,7),W(40,7)
      COMMON/FUSLG2/NCF,DCL(40),X(40),ZC(40),RMEAN(40),
     1              AMEAN(40),RP(40),AP(40),AL(7),SA(7),CA(7),
     2              GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),
     3              SB(20),CB(20),THETA(21),ST(20),CT(20),
     4              YFWD(21,40),YAFT(21,40),ZFWD(21,40),ZAFT(21,40)
      COMMON/FUSLG3/URA(40),UXA(40),VA(20),WA(20),VS(20,7),WS(20,7),
     1              US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),
     2              CM(40,7),CN(40,7)
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      DIST(A,B) = SQRT(A**2 + B**2)
      DO 20 I = 1,NSEG
      RMEAN(I) = 0.5 * (RADIUS(I) + RADIUS(I+1))
      AMEAN(I) = PI * RMEAN(I)**2
C  CALCULATE THE AREA GRADIENTS
      RP(I) = (RADIUS(I+1) - RADIUS(I)) / (XB(I+1) - XB(I))
      AP(I) = (CSA(I+1) - CSA(I)) / (XB(I+1) - XB(I))
   20 CONTINUE
C  CALCULATE THE VELOCITY POTENTIAL AND ITS X- AND R- DERIVATIVES
C  I = STATION WHERE PHI IS CALCULATED
C  J = INFLUENCING STATION
      FPI = 0.25 / PI
      DO 40 I = 1,NSEG
      URA(I) = 0.0
      UXA(I) = 0.0
      DO 30 J = 1,NSEG
      DX1 = XB(J+1) - X(I)
      DX2 = XB(J) - X(I)
      SQRT1 = DIST(DX1,RMEAN(I))
      SQRT2 = DIST(DX2,RMEAN(I))
      URA(I) = URA(I) + FPI * AP(J) * (DX1 / SQRT1 - DX2 / SQRT2) /
     1    RMEAN(I)
      UXA(I) = UXA(I) + FPI * AP(J) * (1.0 / SQRT1 - 1.0 / SQRT2)
   30 CONTINUE
   40 CONTINUE
      IF(IPRINT .GE. 0)RETURN
      WRITE(6, 50 )
   50 FORMAT(1H1,48X,2H**,6(5H*****)/
     1       49X,32H* FUSELAGE AXIAL FLOW SOLUTION */49X,2H**,
     2       6(5H*****)/10X,3HSEG,9X,1HX,13X,1HR,13X,2HRP,13X,1HA,12X,
     3       2HAP,12X,2HUX,12X,2HUR/)
      WRITE(6, 60 )(I,X(I),RMEAN(I),RP(I),AMEAN(I),AP(I),UXA(I),URA(I),
     1                 I = 1,NSEG)
   60 FORMAT(10X,I3,7F14.6)
      RETURN
      END
```

```
*DECK ANGLES
      SUBROUTINE ANGLES(K,N)
C
C  THIS SUBROUTINE CALCULATES POLAR ANGLES FOR USE IN THE CROSS - FLOW
C  SOLUTION
C
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2              V(40,7),W(40,7)
      COMMON/FUSLG2/NCF,DCL(40),X(40),ZC(40),RMEAN(40),
     1              AMEAN(40),RP(40),AP(40),AL(7),SA(7),CA(7),
     2              GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),
     3              SB(20),CB(20),THETA(21),ST(20),CT(20),
     4              YFWD(21,40),Y(21,40),ZFWD(21,40),Z(21,40)
      COMMON/FUSLG3/URA(40),UXA(40),VA(20),WA(20),VS(20,7),WS(20,7),
     1              US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),
     2              CM(40,7),CN(40,7)
      DO 50 J=1,N
      ETA(J)=(Y(J,K)+Y(J+1,K))/2.0
      ZETA(J)=(Z(J,K)+Z(J+1,K))/2.0
      DY=Y(J+1,K)-Y(J,K)
      DZ=Z(J+1,K)-Z(J,K)
      ADY=ABS(DY)
      ADZ=ABS(DZ)
      IF(DY) 40 , 20 , 30
   20 IF(DZ.EQ.0.0)GO TO 60
      IF(DZ.GT.0.0)BETA(J)  =PI/2.0
      IF(DZ.LT.0.0)BETA(J)  =1.5*PI
      GO TO 50
   30 IF(DZ.EQ.0.0)BETA(J)  =0.0
      IF(DZ.GT.0.0)BETA(J)  =ATAN(ADZ/ADY)
      IF(DZ.LT.0.0)BETA(J)  =2.0*PI-ATAN(ADZ/ADY)
      GO TO 50
   40 IF(DZ.EQ.0.0)BETA(J)  =PI
      IF(DZ.GT.0.0)BETA(J)  =PI-ATAN(ADZ/ADY)
      IF(DZ.LT.0.0)BETA(J)  =PI+ATAN(ADZ/ADY)
   50 CONTINUE
      RETURN
   60 WRITE(6, 70 )
   70 FORMAT(1H0,10X,44H*** AN INPUT ERROR HAS CAUSED TERMINATION IN,
     1        22H SUBROUTINE ANGLES ***)
      STOP
      END
```

```
*DECK NEUMAN
      SUBROUTINE NEUMAN(K,N)
C
C  THIS SUBROUTINE SOLVES THE CROSS - FLOW PROBLEM FOR THE K-TH SEGMENT
C  USING A TWO DIMENSIONAL NEUMAN SOLUTION
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),
     2              V(40,7),W(40,7)
      COMMON/FUSLG2/NCF,DCL(40),X(40),ZC(40),RMEAN(40),
     1              AMEAN(40),RP(40),AP(40),AL(7),SA(7),CA(7),
     2              GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),
     3              SB(20),CB(20),THETA(21),ST(20),CT(20),
     4              YFWD(21,40),Y(21,40),ZFWD(21,40),Z(21,40)
      COMMON/FUSLG3/URA(40),UXA(40),VA(20),WA(20),VS(20,7),WS(20,7),
     1              US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),
     2              CM(40,7),CN(40,7)
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE
      COMMON/SPIRIT/NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,LBAR,XMC,XCG
      COMMON/SLIPIT/SSLIP
      DIMENSION T(20)
      DIMENSION B(20,7),SV(20,20),SW(20,20),A(20,20),AS(20,20),C(20)
      DIMENSION SVS(20,20),SWS(20,20)
      EQUIVALENCE (SV(1,1),YFWD(1,1)),(SW(1,1),ZFWD(1,1))
      REAL LRR
      LOGICAL DERIV
      DERIV = LOGIC .EQ. 2
      TPI=0.5/PI
      SSS = SIN(SSLIP / 57.295779)
      CSS = COS(SSLIP / 57.295779)
      DO 20 I=1,N
      T(I)=(Y(I+1,K)-Y(I,K))**2+(Z(I+1,K)-Z(I,K))**2
      SB(I)=SIN(BETA(I))
   20 CB(I)=COS(BETA(I))
C
C     K = SEGMENT NUMBER.
C     I = ELEMENT WHERE INDUCED VELOCITY IS TO BE CALCULATED.
C     J = ELEMENT INDUCING FLOW ON ELEMENT I.
C
      DO 80 I=1,N
      DO 70 J=1,N
      IF(J.EQ.I)GO TO 30
      R1=SQRT((Y(J,K)-ETA(I))**2+(Z(J,K)-ZETA(I))**2)
      R2=SQRT((Y(J+1,K)-ETA(I))**2+(Z(J+1,K)-ZETA(I))**2)
      ARG=(R1**2+R2**2-T(J))/(2.0*R1*R2)
      IF(ARG.GT.1.0)ARG=1.0
      IF(ARG.LT.-1.0)ARG=-1.0
      THETA3=ACOS(ARG)
      SQUIG=-(ETA(I)-Y(J,K))*SB(J)+(ZETA(I)-Z(J,K))*CB(J)
      IF(SQUIG.LT.0.0)THETA3=-THETA3
      LRR=ALOG(R1/R2)
      FACT1 = TPI * (CB(J) * LRR - SB(J) * THETA3)
      FACT2 = TPI * (SB(J) * LRR + CB(J) * THETA3)
      GO TO 40
   30 FACT1 = 0.5 * SB(J)
      FACT2 = -0.5 * CB(J)
   40 IF(ISYMM .GT. 0)GO TO 50
      R1=SQRT((-Y(J+1,K)-ETA(I))**2+(Z(J+1,K)-ZETA(I))**2)
      R2=SQRT((-Y(J,K)-ETA(I))**2+(Z(J,K)-ZETA(I))**2)
      ARG=(R1**2+R2**2-T(J))/(2.0*R1*R2)
      IF(ARG.GT.1.0)ARG=1.0
      IF(ARG.LT.-1.0)ARG=-1.0
      THETA3=ACOS(ARG)
      SQUIG=(ETA(I)+Y(J+1,K))*SB(J)+(ZETA(I)-Z(J+1,K))*CB(J)
      IF(SQUIG.LT.0.0)THETA3=-THETA3
      LRR=ALOG(R1/R2)
      FACT3 = TPI * (CB(J) * LRR + SB(J) * THETA3)
      FACT4 = TPI *(-SB(J) * LRR + CB(J) * THETA3)
      GO TO 60
   50 FACT3 = 0.0
      FACT4 = 0.0
   60 SV(I,J) = FACT1 + FACT3
      SW(I,J) = FACT2 + FACT4
      A(I,J) = SV(I,J) * SB(I) - SW(I,J) * CB(I)
      IF(.NOT. DERIV)GO TO 70
      SVS(I,J) = FACT1 - FACT3
      SWS(I,J) = FACT2 - FACT4
      AS(I,J) = SVS(I,J) * SB(I) - SWS(I,J) * CB(I)
   70 CONTINUE
   80 CONTINUE
C     THE LEFT SIDE MATRIX IS COMPLETE.  SET UP THE RIGHT SIDE MATRIX.
      DO 100 JA = 1,5
      DO 90 I = 1,N
      B(I,JA) = (SA(JA)*CSS+W(K,JA)-DCL(K))*CB(I) - (V(K,JA)-SSS)*SB(I)
   90 CONTINUE
  100 CONTINUE
C  SET UP THE ADDITIONAL RIGHT HAND SIDES FOR STABILITY
      IF(.NOT. DERIV) GO TO 120
      W(K,6) =  (2.0 / CREF) * (X(K) - XCG) - DCL(K)
      V(K,6) = 0.0
      W(K,7) = 0.0
      V(K,7) =  (2.0 / SPAN) *(X(K) - XCG)
      DO 110 I = 1,N
      B(I,6) = W(K,6) * CB(I)
      B(I,7) = -V(K,7) * SB(I)
```

```
      C(I) = B(I,7)                                                    NEUM0960
  110 CONTINUE                                                         NEUM0970
C     SOLVE THE MATRIX.                                                NEUM0980
  120 NSIZE=N                                                          NEUM0990
      NDD = 20                                                         NEUM1000
      M = 5                                                            NEUM1010
      IF(DERIV)M = 6                                                   NEUM1020
      D=1.0                                                            NEUM1030
      CALL MISI(A,NSIZE,NDD,B,M,NERR,D)                                NEUM1040
      IF(.NOT. DERIV)GO TO 140                                         NEUM1050
      M = 1                                                            NEUM1060
      CALL MISI(AS,NSIZE,NDD,C,M,NERR,D)                               NEUM1070
      DO 130 I = 1,N                                                   NEUM1080
  130 B(I,7) = C(I)                                                    NEUM1090
  140 CONTINUE                                                         NEUM1100
C     THE SOLUTION IS NOW IN ARRAY B(I,J).                             NEUM1110
C                                                                      NEUM1120
C     CALCULATE THE LOCAL SURFACE VELOCITIES FROM THE SOURCE STRENGTHS NEUM1130
C     AND INDUCED FLOW INFLUENCE COEFFICIENTS.                         NEUM1140
      M = 5                                                            NEUM1150
      IF(DERIV)M = 6                                                   NEUM1160
      DO 170 JA = 1,M                                                  NEUM1170
      DO 160 I=1,N                                                     NEUM1180
      VS(I,JA)=0.0                                                     NEUM1190
      WS(I,JA)=0.0                                                     NEUM1200
      DO 150 J=1,N                                                     NEUM1210
      VS(I,JA)=VS(I,JA)+B(J,JA)*SV(I,J)                                NEUM1220
  150 WS(I,JA)=WS(I,JA)+B(J,JA)*SW(I,J)                                NEUM1230
      VS(I,JA)=VS(I,JA)+V(K,JA) - SSS                                  NEUM1240
      WS(I,JA)=WS(I,JA)+SA(JA)*CSS+W(K,JA)-DCL(K)                      NEUM1250
  160 CONTINUE                                                         NEUM1260
  170 CONTINUE                                                         NEUM1270
      IF(.NOT. DERIV)RETURN                                            NEUM1280
      DO 190 I = 1,N                                                   NEUM1290
      VS(I,7) = 0.0                                                    NEUM1300
      WS(I,7) = 0.0                                                    NEUM1310
      DO 180 J = 1,N                                                   NEUM1320
      VS(I,7) = VS(I,7) + B(J,7) * SVS(I,J)                            NEUM1330
  180 WS(I,7) = WS(I,7) + B(J,7) * SWS(I,J)                            NEUM1340
      VS(I,7) = VS(I,7) + V(K,7)                                       NEUM1350
      WS(I,7) = WS(I,7) + W(K,7) -DCL(K)                               NEUM1360
  190 CONTINUE                                                         NEUM1370
      RETURN                                                           NEUM1380
      END                                                              NEUM1390


*DECK CPRES                                                            CPRS0010
      SUBROUTINE CPRES(K,N)                                            CPRS0020
C                                                                      CPRS0030
C  THIS SUBROUTINE COMBINES THE PERTURBATION VELOCITIES AND CALCULATES CPRS0040
C  THE PRESSURE DISTRIBUTION ON THE K-TH SEGMENT.                      CPRS0050
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),   CPRS0060
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),              CPRS0070
     2              V(40,7),W(40,7)                                    CPRS0080
      COMMON/FUSLG2/NCF,DCL(40),X(40),ZC(40),RMEAN(40),                CPRS0090
     1              AMEAN(40),RP(40),AP(40),AL(7),SA(7),CA(7),         CPRS0100
     2              GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),   CPRS0110
     3              SB(20),CB(20),THETA(21),ST(20),CT(20),             CPRS0120
     4              YFWD(21,40),Y(21,40),ZFWD(21,40),Z(21,40)          CPRS0130
      COMMON/FUSLG3/URA(40),UXA(40),VA(20),WA(20),VS(20,7),WS(20,7),   CPRS0140
     1              US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),        CPRS0150
     2              CM(40,7),CN(40,7)                                  CPRS0160
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           CPRS0170
      COMMON/SPIRIT/NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                CPRS0180
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG    CPRS0190
      COMMON/SLIPIT/SSLIP                                              CPRS0200
      LOGICAL DERIV,PRINT                                              CPRS0210
      DERIV = LOGIC .EQ. 2                                             CPRS0220
      PRINT = IPRINT .LT. 0                                            CPRS0230
      NI = N + 1                                                       CPRS0240
      SSS = SIN(SSLIP / 57.295779)                                     CPRS0250
      CSS = COS(SSLIP / 57.295779)                                     CPRS0260
      GO TO ( 20 , 170 , 20 ),ISECT                                    CPRS0270
C  ARBITRARY OR ELLIPTICAL SECTIONS                                    CPRS0280
   20 DO 160 I = 1,N                                                   CPRS0290
      ET = ETA(I)                                                      CPRS0300
      ZE = ZETA(I) - ZC(K)                                             CPRS0310
      IF(ET) 110 , 30 , 70                                             CPRS0320
   30 IF(ZE) 40 , 50 , 60                                              CPRS0330
   40 THETA(I) = 0.0                                                   CPRS0340
      GO TO 150                                                        CPRS0350
   50 GO TO 360                                                        CPRS0360
   60 THETA(I) = PI                                                    CPRS0370
      GO TO 150                                                        CPRS0380
   70 IF(ZE) 80 , 90 , 100                                             CPRS0390
   80 THETA(I) = ATAN(-ET / ZE)                                        CPRS0400
      GO TO 150                                                        CPRS0410
   90 THETA(I) = PI / 2.0                                              CPRS0420
      GO TO 150                                                        CPRS0430
  100 THETA(I) = PI - ATAN(ET / ZE)                                    CPRS0440
      GO TO 150                                                        CPRS0450
  110 IF(ZE) 120 , 130 , 140                                           CPRS0460
  120 THETA(I) = 2.0 * PI - ATAN(ET / ZE)                              CPRS0470
      GO TO 150                                                        CPRS0480
  130 THETA(I) = 1.5 * PI                                              CPRS0490
      GO TO 150                                                        CPRS0500
  140 THETA(I) = PI + ATAN(-ET/ZE)                                     CPRS0510
  150 ST(I) = SIN(THETA(I))                                            CPRS0520
      CT(I) = COS(THETA(I))                                            CPRS0530
  160 CONTINUE                                                         CPRS0540
      GO TO 210                                                        CPRS0550
  170 DT = PI / N                                                      CPRS0560
      IF(ISYMM .GT. 0)DT = 2.0 * DT                                    CPRS0570
      THETA(1) = 0.0                                                   CPRS0580
      DO 180 I = 2,NI                                                  CPRS0590
  180 THETA(I) = THETA(I-1) + DT                                       CPRS0600
      DO 190 I = 1,N                                                   CPRS0610
      THETA(I) = 0.5 * (THETA(I) + THETA(I+1))                         CPRS0620
      BETA(I) = THETA(I)                                               CPRS0630
      ST(I) = SIN(THETA(I))                                            CPRS0640
      CT(I) = COS(THETA(I))                                            CPRS0650
      SB(I) = ST(I)                                                    CPRS0660
  190 CB(I) = CT(I)                                                    CPRS0670
      DO 200 I = 1,N                                                   CPRS0680
      ETA(I) = (Y(I,K) + Y(I+1,K)) / 2.0                               CPRS0690
  200 ZETA(I) = (Z(I,K) + Z(I+1,K)) / 2.0                              CPRS0700
      IF(.NOT. DERIV) GO TO 210                                        CPRS0710
      W(K,6) =  (2.0 / CREF) * (X(K) - XCG) - DCL(K)                   CPRS0720
      V(K,6) = 0.0                                                     CPRS0730
      W(K,7) = 0.0                                                     CPRS0740
      V(K,7) =  (2.0 / SPAN) * (X(K) - XCG)                            CPRS0750
  210 IF(.NOT. PRINT) GO TO 240                                        CPRS0760
      WRITE(6, 220 )K                                                  CPRS0770
  220 FORMAT(1H1,42X,9(5H*****)/                                       CPRS0780
     1         43X,41H* FUSELAGE CROSS FLOW SOLUTION - SEGMENT ,I2,2H */ CPRS0790
     2         43X,9(5H*****))                                         CPRS0800
      WRITE(6, 230 )XB(K),XB(K+1),X(K),RADIUS(K),RADIUS(K+1),RMEAN(K)  CPRS0810
  230 FORMAT(1H0,26X,8HX-FWD = ,F12.6,5X,8HX-AFT = ,F12.6,5X,          CPRS0820
     1         9HX-MEAN = ,F12.6/27X,8HR-FWD = ,F12.6,5X,8HR-AFT = ,F12.6,CPRS0830
     2         5X,9HR-MEAN = ,F12.6)                                   CPRS0840
  240 DO 330 JA = 1,NCF                                                CPRS0850
      IF(.NOT. PRINT) GO TO 270                                        CPRS0860
  250 IF(JA .EQ. 1)WRITE(6, 260 )                                      CPRS0870
  260 FORMAT(1H0,11X,2HNO,7X,1HY,9X,1HZ,9X,5HCP(0),6X,5HCP(5),5X,      CPRS0880
     1         6HCP(10),5X,6HCP(15),5X,6HCP(20),6X,5HCP(Q),6X,5HCP(R),/) CPRS0890
  270 DO 280 I = 1,N                                                   CPRS0900
      VA(I) = URA(K) * CA(JA) * CSS * ST(I)                            CPRS0910
  280 WA(I) = -URA(K) * CA(JA) * CSS * CT(I)                           CPRS0920
      IF(ISECT .NE. 2)GO TO 300                                        CPRS0930
      UTHETA = 2.0 * (SA(JA)*CSS + W(K,JA) - DCL(K))                   CPRS0940
      VTHETA = 2.0 * (V(K,JA) - SSS)                                   CPRS0950
      DO 290 I = 1,N                                                   CPRS0960
      FACTOR = UTHETA * ST(I) + VTHETA * CT(I)                         CPRS0970
      VS(I,JA) = FACTOR * CT(I)                                        CPRS0980
  290 WS(I,JA) = FACTOR * ST(I)                                        CPRS0990
  300 DO 310 I = 1,N                                                   CPRS1000
  310 US(I) = 2.0 * ((V(K,JA)-SSS)*ST(I)-(SA(JA)*CSS+W(K,JA)-DCL(K))*  CPRS1010
     1  CT(I)) * RP(K)                                                 CPRS1020
C  CALCULATE THE TOTAL SURFACE VELOCITIES AND PRESSURE COEFFICIENTS    CPRS1030
      DO 320 I = 1,N                                                   CPRS1040
      UTOT = CA(JA) * (1.0 + UXA(K)) + US(I)                           CPRS1050
      VTOT = VA(I) + VS(I,JA)                                          CPRS1060
      WTOT = WA(I) + WS(I,JA)                                          CPRS1070
      CP(I,JA) = 1.0 - (VTOT**2 + WTOT**2 + UTOT**2)                   CPRS1080
  320 CONTINUE                                                         CPRS1090
  330 CONTINUE                                                         CPRS1100
      IF(.NOT. PRINT) RETURN                                           CPRS1110
      DO 350 I = 1,N                                                   CPRS1120
      WRITE(6, 340 )I,ETA(I),ZETA(I),(CP(I,JA),JA = 1,NCF)             CPRS1130
  340 FORMAT(13X,I2,9F11.6)                                            CPRS1140
  350 CONTINUE                                                         CPRS1150
      RETURN                                                           CPRS1160
  360 WRITE(6, 370 )                                                   CPRS1170
  370 FORMAT(1H0,10X,44H*** AN INPUT ERROR HAS CAUSED TERMINATION IN,  CPRS1180
     1         21H SUBROUTINE CPRES ***)                               CPRS1190
      STOP                                                             CPRS1200
      END                                                              CPRS1210


*DECK SUMER1                                                           SMR10010
      SUBROUTINE SUMER1(K,N)                                           SMR10020
C     THIS SUBROUTINE SUMS THE NORMAL COMPONENTS OF THE PRESSURE       SMR10030
C     COEFFICIENT TO DETERMINE THE BODY PITCHING MOMENT CHARACTERISTICS.SMR10040
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),   SMR10050
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),              SMR10060
     2              V(40,7),W(40,7)                                    SMR10070
      COMMON/FUSLG2/NCF,DCL(40),X(40),ZC(40),RMEAN(40),                SMR10080
     1              AMEAN(40),RP(40),AP(40),AL(7),SA(7),CA(7),         SMR10090
     2              GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),   SMR10100
     3              SB(20),CB(20),THETA(21),ST(20),CT(20),             SMR10110
     4              YFWD(21,40),Y(21,40),ZFWD(21,40),Z(21,40)          SMR10120
      COMMON/FUSLG3/URA(40),UXA(40),VA(20),WA(20),VS(20,7),WS(20,7),   SMR10130
     1              US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),        SMR10140
     2              CM(40,7),CN(40,7)                                  SMR10150
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           SMR10160
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG    SMR10170
      LOGICAL NSYMM                                                    SMR10180
      NSYMM = ISYMM .GT. 0                                             SMR10190
      DO 40 JA = 1,NCF                                                 SMR10200
      CX(K,JA)=0.0                                                     SMR10210
      CY(K,JA)=0.0                                                     SMR10220
      CZ(K,JA)=0.0                                                     SMR10230
      CM(K,JA)=0.0                                                     SMR10240
      CN(K,JA) = 0.0                                                   SMR10250
      DO 20 I = 1,N                                                    SMR10260
      SG=SIN(GAMMA(I,K))                                               SMR10270
      CG=COS(GAMMA(I,K))                                               SMR10280
      CX(K,JA)=CX(K,JA)+CP(I,JA)*SG*S(I,K)/AREA                        SMR10290
      IF(NSYMM)CY(K,JA) = CY(K,JA) - CP(I,JA) * CG * SB(I) * S(I,K)/AREASMR10300
      CZ(K,JA)=CZ(K,JA)+CP(I,JA)*CG*CB(I)*S(I,K)/AREA                  SMR10310
      CM(K,JA)=CM(K,JA)-CP(I,JA)*((X(K)-XMC)*CG*CB(I)+ZETA(I)*SG)*S(I,K)SMR10320
     1 /(AREA*CREF)                                                    SMR10330
      IF((NSYMM) .OR. (JA .EQ. 7))CN(K,JA) = CN(K,JA) - CP(I,JA) *     SMR10340
     1           ((X(K) - XMC) * CG * SB(I) - ETA(I) * SG) * S(I,K) /  SMR10350
     2           (AREA * CREF)                                         SMR10360
   20 CONTINUE                                                         SMR10370
      IF(NSYMM)GO TO 40                                                SMR10380
      CX(K,JA)=2.0*CX(K,JA)                                            SMR10390
      CZ(K,JA)=2.0*CZ(K,JA)                                            SMR10400
      CM(K,JA) = 2.0 * CM(K,JA)                                        SMR10410
   30 CN(K,JA) = 2.0 * CN(K,JA)                                        SMR10420
   40 CONTINUE                                                         SMR10430
      RETURN                                                           SMR10440
      END                                                              SMR10450


*DECK SUMER2                                                           SMR20010
      SUBROUTINE SUMER2                                                SMR20020
C     THIS SUBROUTINE SUMS THE SECTIONAL FORCES AND MOMENTS.           SMR20030
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB               SMR20040
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),   SMR20050
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),              SMR20060
     2              V(40,7),W(40,7)                                    SMR20070
      COMMON/FUSLG2/NCF,DCL(40),X(40),ZC(40),RMEAN(40),                SMR20080
     1              AMEAN(40),RP(40),AP(40),AL(7),SA(7),CA(7),         SMR20090
     2              GAMMA(20,40),S(20,40),ETA(20),ZETA(20),BETA(20),   SMR20100
     3              SB(20),CB(20),THETA(21),ST(20),CT(20),             SMR20110
     4              YFWD(21,40),Y(21,40),ZFWD(21,40),Z(21,40)          SMR20120
      COMMON/FUSLG3/URA(40),UXA(40),VA(20),WA(20),VS(20,7),WS(20,7),   SMR20130
     1              US(20),CP(20,7),CX(40,7),CY(40,7),CZ(40,7),        SMR20140
     2              CM(40,7),CN(40,7)                                  SMR20150
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           SMR20160
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HX,HH,VL,VH,    SMR20170
     1             NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,   SMR20180
```

```
  2                 ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES                       SMR20190
      COMMON/KEEPEM/CLW(5),CMW(5),CLLW(5),CDW(5),CNW(5),CYW(5),CLQW,          SMR20200
     1          CMQW,CLLPW,CLLRW,CNPW,CNRW,CYPW,CYRW,CLH(5),CMH(5),CLLH(5),SMR20210
     2          CDH(5),CLQH,CMQH,CLLPH,CYH(5),CYBV,CNV(5),CNBV,CLLV(5),       SMR20220
     3          CLLBV,CDV(5),CLLPV,CLLRV,CNPV,CNRV,CYPV,CYRV,CMF(5),CNF(5),SMR20230
     4          CMQF,CNRF                                                      SMR20240
      DIMENSION FX(7),FY(7),FZ(7),FM(7),FN(7)                                  SMR20250
      DO 30 JA = 1,NCF                                                         SMR20260
      FX(JA)=0.0                                                               SMR20270
      FY(JA)=0.0                                                               SMR20280
      FZ(JA)=0.0                                                               SMR20290
      FM(JA)=0.0                                                               SMR20300
      FN(JA) = 0.0                                                             SMR20310
      DO 20 K = 1,NSEG                                                         SMR20320
      FX(JA)=FX(JA)+CX(K,JA)                                                   SMR20330
      FY(JA)=FY(JA)+CY(K,JA)                                                   SMR20340
      FZ(JA)=FZ(JA)+CZ(K,JA)                                                   SMR20350
      FM(JA)=FM(JA)+CM(K,JA)                                                   SMR20360
   20 FN(JA) = FN(JA) + CN(K,JA)                                               SMR20370
   30 CONTINUE                                                                 SMR20380
C   SAVE THE TOTAL STATIC AND DYNAMIC MOMENTS                                  SMR20390
      DO 40 JA = 1,5                                                           SMR20400
      CMF(JA) = FM(JA)                                                         SMR20410
      CNF(JA) = FN(JA)                                                         SMR20420
   40 CONTINUE                                                                 SMR20430
      CALL HEADER(ILINES)                                                      SMR20440
      WRITE(6, 50 )KCASE                                                       SMR20450
   50 FORMAT(1H0,31X,1H*,13(5H*****),/,                                        SMR20460
     1         32X,51H* SECTIONAL AND TOTAL FUSELAGE LOADING FROM PRESSUR,SMR20470
     2         15HE INTEGRATION *,/,                                           SMR20480
     3         32X,1H*,21X,18HFOR COMPOSITE CASE,I4,21X,1H*,/,                 SMR20490
     4         32X,1H*,13(5H*****))                                            SMR20500
      ILINES = ILINES + 5                                                      SMR20510
      WRITE(6, 60 )                                                            SMR20520
   60 FORMAT(1H0,3X,3HSEG,6X,1HX,8X,5HCM(0),6X,5HCN(0),6X,5HCM(5),6X,         SMR20530
     1         5HCN(5),6X,6HCM(10),5X,6HCN(10),5X,6HCM(15),5X,6HCN(15),        SMR20540
     2         5X,6HCM(20),5X,6HCN(20)/)                                       SMR20550
      DO 80 K = 1,NSEG                                                         SMR20560
      WRITE(6, 70 )K,X(K),(CM(K,N),CN(K,N),N=1,5)                              SMR20570
   70 FORMAT(4X,I3,11F11.6)                                                    SMR20580
   80 CONTINUE                                                                 SMR20590
      WRITE(6, 90 )(FM(N)?FN(N),N=1,5)                                         SMR20600
   90 FORMAX(4X,25(5H-----)/2X,5HTOTAL,11X,10F11.6)                            SMR20610
      IF(LOGIC .NE. 2) GO TO 110                                               SMR20620
      WRITE(6, 100 )FM(6),FN(7)                                                SMR20630
  100 FORMAT(1H0,42X,31HFUSELAGE CONTRIBUTION TO CMQ = ,F12.7/                 SMR20640
     1              43X,31HFUSELAGE CONTRIBUTION TO CNR = ,F12.7)              SMR20650
      CMQF = FM(6)                                                             SMR20660
      CNRF = FN(7)                                                             SMR20670
  110 WRITE(6, 120 )                                                           SMR20680
  120 FORMAT(1H0,9X,42H*** NOTE *** ALL FUSELAGE COEFFICIENTS AND,            SMR20690
     1         49H STABILITY DERIVATIVES ARE NON-DIMENSIONALIZED BY,           SMR20700
     2         19H WING AREA AND CREF)                                         SMR20710
      RETURN                                                                   SMR20720
      END                                                                      SMR20730


*DECK MIS1                                                                     MIS10010
```

```
      SUBROUTINE MIS1 ( A, N, ND, B, MD, NERR, D )                             MIS10020
C                                                                              MIS10030
C         A    = IN-OUT = A TWO DIMENSIONAL ARRAY (DIMENSIONED ND BY           MIS10040
C                         ND) CONTAINING AN INPUT MATRIX OF ORDER N.           MIS10050
C                         THE A MATRIX IS DESTROYED BY THE SUBROUTINE.         MIS10060
C                                                                              MIS10070
C         N    = INPUT  = THE ORDER (INTEGER) OF A (N LESS THAN OR             MIS10080
C                         EQUAL TO ND) AND THE NUMBER OF ROWS IN B.            MIS10090
C                                                                              MIS10100
C         ND   = INPUT  = THE MAXIMUM DIMENSIONS (INTEGER) OF THE              MIS10110
C                         SQUARE MATRIX A.                                     MIS10120
C                                                                              MIS10130
C         B    = IN-OUT = A TWO DIMENSIONAL ARRAY (DIMENSIONED ND BY           MIS10140
C                         MD) CONTAINING THE SECOND INPUT MATRIX.              MIS10150
C                         THIS ARRAY IS MODIFIED AND CONTAINS (UPON            MIS10160
C                         OUTPUT) THE C MATRIX SOLUTION.                       MIS10170
C                                                                              MIS10180
C         MD   = INPUT  = THE NUMBER (INTEGER) OF COLUMNS IN B.                MIS10190
C                                                                              MIS10200
C         NERR = OUTPUT = OUTPUT CODE (INTEGER)                                MIS10210
C                       = 0  IF A IS NON-SINGULAR                              MIS10220
C                       = 1  IF A IS SINGULAR                                  MIS10230
C                       = 2  IF N IS LESS THAN OR EQUAL TO ZERO                MIS10240
C                       = 3  IF N IS GREATER THAN ND                           MIS10250
C                                                                              MIS10260
C         D    = INPUT  = SCALE FACTOR TO SCALE DETERMINENT.                   MIS10270
C              = OUTPUT = THE SCALED DETERMINENT OF A.                         MIS10280
C                                                                              MIS10290
      DIMENSION  A(ND,ND), B(ND,MD)                                            MIS10300
      EQUIVALENCE  (L,FL), (K,FK)                                              MIS10310
C                                                                              MIS10320
C     CHECK THE VALUE OF N                                                     MIS10330
C                                                                              MIS10340
      IF  ( N - 1 ) 20 , 30 , 40                                               MIS10350
C                                                                              MIS10360
   20 D = 0.0E0                                                                MIS10370
      NERR = 2                                                                 MIS10380
      GO TO 200                                                                MIS10390
C                                                                              MIS10400
   30 AIJMAX = A(1,1)                                                          MIS10410
      IF  ( AIJMAX .EQ. 0.0E0 )  GO TO 70                                      MIS10420
      B(1,1) = B(1,1) / AIJMAX                                                 MIS10430
      D = D * AIJMAX                                                           MIS10440
      GO TO 190                                                                MIS10450
C                                                                              MIS10460
   40 IF  ( N .LE. ND )  GO TO 50                                              MIS10470
      NERR = 3                                                                 MIS10480
      GO TO 200                                                                MIS10490
C                                                                              MIS10500
C     START REDUCTION OF MATRIX A                                              MIS10510
C                                                                              MIS10520
   50 DO 140    I = 1, N                                                       MIS10530
C                                                                              MIS10540
C     SEARCH FOR MAXIMUM ELEMENT IN ITH ROW OF A-MATRIX                        MIS10550
C                                                                              MIS10560
      AIJMAX = A(I,1)                                                          MIS10570
      JMAX = 1                                                                 MIS10580
      DO 60    J = 2, N                                                        MIS10590
      IF  ( ABS( A(I,J) ) .LE. ABS( AIJMAX ) )  GO TO 60                       MIS10600
      AIJMAX = A(I,J)                                                          MIS10610
```

```
      JMAX = J                                                                 MIS10620
   60 CONTINUE                                                                 MIS10630
C                                                                              MIS10640
C     IF AIJMAX IS ZERO, THE MATRIX IS SINGULAR                                MIS10650
C                                                                              MIS10660
      IF  ( ABS( AIJMAX ) .GT  0.0E0 )  GO TO 80                               MIS10670
   70 D = 0.0E0                                                                MIS10680
      NERR = 1                                                                 MIS10690
      GO TO 200                                                                MIS10700
C                                                                              MIS10710
C     NORMALIZE ITH ROW BY AIJMAX (JMAX ELEMENT OF ITH ROW)                    MIS10720
C                                                                              MIS10730
   80 DO 90    J = 1, N                                                        MIS10740
   90 A(I,J) = A(I,J) / AIJMAX                                                 MIS10750
      D = D * AIJMAX                                                           MIS10760
C                                                                              MIS10770
C     NORMALIZE ITH ROW OF B                                                   MIS10780
C                                                                              MIS10790
      DO 100   J = 1, MD                                                       MIS10800
  100 B(I,J) = B(I,J) / AIJMAX                                                 MIS10810
C                                                                              MIS10820
C     USE ROW TRANSFORMATIONS TO GET ZEROS ABOVE AND BELOW THE JMAX            MIS10830
C     ELEMENT OF THE ITH ROW OF A.  APPLY SAME ROW TRANSFORMATIONS             MIS10840
C     TO THE B MATRIX.                                                         MIS10850
C                                                                              MIS10860
      DO 130    K = 1, N                                                       MIS10870
      IF  ( K .EQ. I )  GO TO 130                                              MIS10880
      ARAT = -A(K,JMAX)                                                        MIS10890
      DO 110    J = 1, N                                                       MIS10900
      IF  ( ABS( A(I,J) ) .EQ. 0.0E0 )  GO TO 110                              MIS10910
      A(K,J) = ARAT * A(I,J) + A(K,J)                                          MIS10920
  110 CONTINUE                                                                 MIS10930
      A(K,JMAX) = 0.0E0                                                        MIS10940
      DO 120    J = 1, MD                                                      MIS10950
      IF  ( ABS( B(I,J) ) .EQ. 0.0E0 )  GO TO 120                              MIS10960
      B(K,J) = ARAT * B(I,J) + B(K,J)                                          MIS10970
  120 CONTINUE                                                                 MIS10980
  130 CONTINUE                                                                 MIS10990
C                                                                              MIS11000
C     STORE ROW COUNTER (I) IN TOP ELEMENT OF JMAX COLUMN. THUS,               MIS11010
C     THE TOP ROW OF A WILL CONTAIN THE LOC OF THE PIVOT (UNITY)               MIS11020
C     ELEMENT OF EACH COLUMN (AFTER REDUCTION).                                MIS11030
C                                                                              MIS11040
      L = I                                                                    MIS11050
  140 A(1,JMAX) = FL                                                           MIS11060
C                   THIS STORES INTEGER IN TOP ROW OF A                        MIS11070
C                                                                              MIS11080
C     THE REDUCTION OF A IS NOW COMPLIETE. PERFORM ROW INTERCHANGES            MIS11090
C     AS INDICATED IN THE FIRST ROW OF A.                                      MIS11100
C                                                                              MIS11110
      DO 180    I = 1, N                                                       MIS11120
      K = I                                                                    MIS11130
  150 FK = A(1,K)                                                              MIS11140
C                   THIS PUTS THE INTEGER VALUE IN A INTO K                    MIS11150
      IF  ( K - I ) 150 , 180 , 160                                            MIS11160
C                                                                              MIS11170
C     IF K(1,I) IS LESS THAN I, THEN THAT ROW HAS ALREADY BEEN                 MIS11180
C     INVOLVED IN AN INTERCHANGE, AND WE USE K(1,K) UNTIL WE GET               MIS11190
C     A VALUE OF K GREATER THAN I  (CORRESPONDING TO A ROW STORED              MIS11200
C     BELOW THE ITH ROW).                                                      MIS11210
```

```
                                                                               MIS11220
C                                                                              MIS11230
  160 DO 170    J = 1, MD                                                      MIS11240
      ARAT = B(I,J)                                                            MIS11250
      B(I,J) = B(K,J)                                                          MIS11260
  170 B(K,J) = ARAT                                                            MIS11270
      D = -D                                                                   MIS11280
  180 CONTINUE                                                                 MIS11290
  190 NERR = 0                                                                 MIS11300
  200 RETURN                                                                   MIS11310
      END                                                                      MIS11320


*DECK STAGII                                                                   STI10010
      OVERLAY(OVLY,13,0)                                                       STI10020
      PROGRAM STAGII                                                           STI10030
C                                                                              STI10040
C  THIS SUBROUTINE CONTROLS CALCULATION OF WING DYNAMIC STABILITY              STI10050
C  DERIVATIVES DUE TO ROLLING AND YAWING RATES                                 STI10060
C                                                                              STI10070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                       STI10080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE                   STI10090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) STI10100
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG            STI10110
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),         STI10120
     1              D(40),KK(400),ITYPE(400)                                   STI10130
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)                    STI10140
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)           STI10150
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                                  STI10160
      COMMON/DERIV/UO(40),CLQ,CMQ,CMQMC                                        STI10170
      COMMON/GEFECT/GHITE(10),FGRND,IGND,NHITE                                 STI10180
      DIMENSION CPREAD(400),CPO(400),CPA(400),CPRO(400),CPRA(400),             STI10190
     1          CPP(400),ONSET(400),ONSETS(400),GAMMA(400)                     STI10200
      EQUIVALENCE (CPREAD(1),GAMMA(1)),(CPO(1),ONSET(1)),                      STI10210
     1            (CPA(1),ONSETS(1))                                           STI10220
C                                                                              STI10230
      IF(IGND .EQ. 0) GO TO 30                                                 STI10240
C  COMPUTE DERIVATIVES FOR ALL COMPOSITE CASES IN GROUND EFFECT                STI10250
   20 CALL GEDER(GAMMA,ONSET,ONSETS,NEWMAX)                                    STI10260
      GO TO 50                                                                 STI10270
C  COMPUTE THE COEFFICIENTS AND DERIVATIVES FOR ALL FUNDAMENTAL CASES          STI10280
C  OUT OF GROUND EFFECT                                                        STI10290
   30 CALL FUNDER(EPS,CPO,CPA,CPRO,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,AREA,        STI10300
     1        CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,     STI10310
     2        NROWS,ISYMM,XLEAD,TANLE,XMC)                                     STI10320
C  COMPUTE THE DERIVATIVES FOR ALL COMPOSITE CASES OUT OF GROUND EFFECT        STI10330
   40 CALL COMDER(EPS,CPO,CPA,CPRO,CPRA,CPP,CPREAD,DEL,CHORD,Y,CMU,           STI10340
     1        DELTA,AREA,CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,        STI10350
     2        NCASES,NROWS,ISYMM,XLEAD,TANLE,XMC)                              STI10360
   50 CONTINUE                                                                 STI10370
      END                                                                      STI10380


*DECK FUNDER                                                                   FUND0010
      SUBROUTINE FUNDER(EPS,CPO,CPA,CPRO,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,FUND0020
     1 AREA,CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,FUND0030
```

```
     2    NROWS,ISYMM,XL,TL,XMC)                                          FUND0040
C                                                                         FUND0050
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL AERODYNAMIC COEFFICIENTS   FUND0060
C  AND STABILITY DERIVATIVES FOR THE FUNDAMENTAL CASES                    FUND0070
C                                                                         FUND0080
      COMMON/SOLVI/CP(400,10)                                             FUND0090
      DIMENSION CPO(NEWMAX),CPA(NEWMAX),CPRO(400),CPRA(400),CPP(400)      FUND0100
      DIMENSION DEL(400),EPS(400,10),EP(400)                              FUND0110
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)           FUND0120
      DIMENSION NW(40),IJ(40)                                             FUND0130
C                                                                         FUND0140
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED           FUND0150
      IREAD = 2 * (NMAX+NJT)                                              FUND0160
      IFUDGE = 0                                                          FUND0170
      CLLP = 0.00                                                         FUND0180
      CNP2 = 0.00                                                         FUND0190
      CNPO = 0.00                                                         FUND0200
C  DEFINE THE DUMMY CP ARRAYS FOR THE ALPHA CASE                          FUND0210
C  AND THE CP ARRAY FOR THE ROLLING FUNDAMENTAL CASE                      FUND0220
      DO 20 I = 1,NEWMAX                                                  FUND0230
      CPA(I) = 0.0                                                        FUND0240
      CPRA(I) = 0.0                                                       FUND0250
      CPP(I) = 2.0 * CP(I,NCASES)                                         FUND0260
   20 CONTINUE                                                            FUND0270
      DO 130 N = 1,NCASES                                                 FUND0280
      CALL HEADER(ILINES)                                                 FUND0290
      LCASE = N                                                           FUND0300
      IREAD = IREAD + 1                                                   FUND0310
C  CALCULATE THE SPANWISE AND TOTAL LOADING                               FUND0320
      IF(N .NE. NCASES) GO TO 30                                          FUND0330
      CONST = 0.0                                                         FUND0340
      CALL SUMIT4(CLLP,DUMI,DUMI,DUMI,DUMI,CPP,CPP,CPP,CONST,EP)          FUND0350
      CLLP = 0.5 * CLLP                                                   FUND0360
   30 CONTINUE                                                            FUND0370
C  READ THE FIRST RUN SOLUTION                                            FUND0380
      CALL READMS(1,CPO,NEWMAX,IREAD)                                     FUND0390
C  DEFINE CP AND EP ARRAYS FOR THE PRESENT SECOND RUN FUNDAMENTAL CASE    FUND0400
      DO 40  I = 1,NEWMAX                                                 FUND0410
      CPO(I) = 2.0 * CPO(I)                                               FUND0420
      CPRO(I) = 2.0 * CP(I,LCASE)                                         FUND0430
      EP(I) = EPS(I,LCASE)                                                FUND0440
   40 CONTINUE                                                            FUND0450
C  CALCULATE THE STABILITY DERIVATIVES                                    FUND0460
      CALL SUMIT1(CPO,CPA,CPP,CPRO,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,   FUND0470
     1   XMC,AREA,DUMI,DUM2,CNP2,CLLRO,DUM3,DUM5,DUM6,CYP2,               FUND0480
     2   NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)                                 FUND0490
      IF(N .LT. NCASES) GO TO 50                                          FUND0500
      GO TO 110                                                           FUND0510
   50 CALL SUMIT2(CPO,CPA,CPRO,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,FUND0520
     1      XMC,CNRO,DUM1,DUM2,CNR20,DUM3,DUM4,CYRO,DUM5,DUM6,            FUND0530
     2      CYR20,DUM7,DUM8,NW,IJ,NROWS,ISYMM,NEWMAX)                     FUND0540
C  PRINT THE STABILITY DERIVATIVE DATA                                    FUND0550
   60 WRITE(6, 70 ) LCASE                                                 FUND0560
   70 FORMAT(1H0,32X,13(4H****), 3H*** / 33X,                             FUND0570
     1        49H*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE, I3,   FUND0580
     2        3H  * / 33X,13(4H****), 3H*** /)                            FUND0590
      WRITE(6, 80 ) CLLRO                                                 FUND0600
   80 FORMAT(1H0/ 25X, 44HROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO,   FUND0610
     1        15H YAWING, CLLR =, F12.7)                                  FUND0620
      WRITE(6, 90 ) CNRO,CNR20                                            FUND0630
   90 FORMAT(1H0/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,       FUND0640
     1      53H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS //   FUND0650
     2        49X, 25HCN(R) = CNR*R + CNR2*R**2 //                        FUND0660
     3        49X, 13HWHERE   CNR =,F14.9 / 56X, 6HCNR2 =,F12.7)          FUND0670
      WRITE(6, 100 ) CYRO,CYR20                                           FUND0680
  100 FORMAT(1H0/ 23X,43HSIDE FORCE COEFFICIENT DUE TO YAWING, CY(R),     FUND0690
     1        29H MAY BE CALCULATED AS FOLLOWS//                          FUND0700
     2        49X,25HCY(R) = CYR*R + CYR2*R**2//                          FUND0710
     3        49X,13HWHERE   CYR =,F14.9/ 56X,6HCYR2 =,F12.7)             FUND0720
      GO TO 130                                                           FUND0730
  110 WRITE(6, 70 ) LCASE                                                 FUND0740
      WRITE(6, 120 ) CLLP,CNP2,CYP2                                       FUND0750
  120 FORMAT(1H0/ 28X, 38HROLLING MOMENT COEFF DERIVATIVE DUE TO,         FUND0760
     1        16H ROLLING, CLLP =, F12.7 ///                              FUND0770
     2        17X, 42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,         FUND0780
     3        44H ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS //          FUND0790
     4        53X, 17HCN(P) = CNP2*P**2 //                                FUND0800
     5        53X, 13HWHERE  CNP2 =,F12.7///                              FUND0810
     6        30X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),        FUNL0820
     7        29H MAY BE CALCULATED AS FOLLOWS//                          FUND0830
     8        53X,17HCY(P) = CYP2*P**2//                                  FUND0840
     9        53X,13HWHERE  CYP2 =,F12.7)                                 FUND0850
  130 CONTINUE                                                            FUND0860
      RETURN                                                              FUND0870
      END                                                                 FUND0880
```

```
*DECK COMDER                                                              COMD0010
      SUBROUTINE COMDER(EPS,CPO,CPA,CPRO,CPRA,CPP,CPREAD,DEL,CHORD,Y,     COMD0020
     1    CMU,DELTA,AREA,CLQ,CMQ,CMQMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,   COMD0030
     2    NCASES,NROWS,ISYMM,XL,TL,XMC)                                   COMD0040
C                                                                         COMD0050
C  THIS SUBROUTINE CONTROLS CALCULATION OF STABILITY DERIVATIVES          COMD0060
C  FOR ALL COMPOSITE CASES                                                COMD0070
C                                                                         COMD0080
      COMMON/SPIRIT/ NEWMX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                   COMD0090
      COMMON/SOLVI/CP(400,10)                                             COMD0100
      COMMON/COMPOS/FACTOR(10,24),NCC                                     COMD0110
      COMMON/FUSLGI/PI,NSEG,NSEGI,NF(40),ISECT,ICAMF,XB(41),ZCL(41),      COMD0120
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                 COMD0130
     2              V(40,7),W(40,7)                                       COMD0140
      DIMENSION CPO(NEWMAX),CPA(NEWMAX),CPRO(400),CPRA(400),CPP(400),     COMD0150
     1          CPREAD(NEWMAX)                                            COMD0160
      DIMENSION DEL(400),EPS(400,10),EP(400)                              COMD0170
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)           COMD0180
      DIMENSION NW(40),IJ(40)                                             COMD0190
      DIMENSION PUT(102)                                                  COMD0200
C                                                                         COMD0210
C  CYCLE THROUGH ALL COMPOSITE CASES                                      COMD0220
      IFUDGE = 1                                                          COMD0230
      DO 110 M = 1,NCC                                                    COMD0240
      MCASE = M                                                           COMD0250
      NCI = NCASES - 1                                                    COMD0260
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED           COMD0270
      IREAD = 2 * (NMAX+NJT) + 1                                          COMD0280
C  FIND WHERE THE WING TOTAL STATIC COEFFICIENTS ARE STORED ON UNIT 9     COMD0290
      IREAD9= 2*NMAX + NSEG + MCASE                                       COMD0300
C  DEFINE CP AND EP ARRAYS FOR THIS SECOND RUN COMPOSITE CASE             COMD0310
C  NOTE THAT CPP HAS PREVIOUSLY BEEN DEFINED IN FUNDER
      DO 30 I = 1,NEWMAX                                                  COMD0320
      CPRO(I) = 0.00                                                      COMD0330
      EP(I) = 0.00                                                        COMD0340
      CPRA(I) = 2.0 * CP(I,1)                                             COMD0350
      DO 20 N = 1,NCI                                                     COMD0360
      CPRO(I) = CPRO(I) + 2.0 * FACTOR(N,M) * CP(I,N)                     COMD0370
      EP(I) = EP(I) + FACTOR(N,M) * EPS(I,N)                              COMD0380
   20 CONTINUE                                                            COMD0390
   30 CONTINUE                                                            COMD0400
C  DEFINE CP FOR THE FIRST RUN COMPOSITE CASE                             COMD0410
      IF(M .EQ. 1) CALL READMS(1,CPA,NEWMAX,IREAD)                        COMD0420
      DO 40 I = 1,NEWMAX                                                  COMD0430
      CPO(I) = 0.00                                                       COMD0440
      CPA(I) = 2.0 * CPA(I)                                               COMD0450
   40 CONTINUE                                                            COMD0460
      IREAD = IREAD - 1                                                   COMD0470
      DO 70 N = 1,NCI                                                     COMD0480
      IREAD = IREAD + 1                                                   COMD0490
      IF(FACTOR(N,M) .EQ. 0.0) GO TO 70                                   COMD0500
   50 CALL READMS(1,CPREAD,NEWMAX,IREAD)                                  COMD0510
      DO 60 I = 1,NEWMAX                                                  COMD0520
      CPREAD(I) = 2.0 * CPREAD(I)                                         COMD0530
      CPO(I) = CPO(I) + FACTOR(N,M) * CPREAD(I)                           COMD0540
   60 CONTINUE                                                            COMD0550
   70 CONTINUE                                                            COMD0560
C  CALCULATE THE COMPOSITE CASE DERIVATIVES                               COMD0570
   80 CALL SUMIT1(CPO,CPA,CPP,CPRO,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,   COMD0580
     1      XMC,AREA,CNPO,CNPA,CNP2,CLLRO,CLLRA,CYPO,CYPA,CYP2,           COMD0590
     2      NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)                              COMD0600
   90 CALL SUMIT2(CPO,CPA,CPRO,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,  COMD0610
     1 XMC,CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,CYRO,CYRA,CYRA2,            COMD0620
     2 CYR20,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)                       COMD0630
C  PRINT A SUMMARY TABLE OF ALL STABILITY DERIVATIVES                     COMD0640
  100 CALL STABLE(CLQ,CMQ,CMQMC,CLLP,CNPO,CNPA,CNP2,CYPO,CYPA,CYP2,       COMD0650
     1      CLLRO,CLLRA,CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,               COMD0660
     2      CYRO,CYRA,CYRA2,CYR20,CYR2A,CYR2A2,MCASE)                     COMD0670
C  READ THE WING TOTAL STATIC COEFFICIENTS FROM UNIT 9                    COMD0680
      IF(LOGIC .EQ. 3) GO TO 110                                          COMD0690
      CALL READMS(9,PUT,102,IREAD9)                                       COMD0691
C  DEFINE THE DYNAMIC COEFFICIENTS FOR STORAGE                            COMD0700
      PUT(31) = CLQ                                                       COMD0710
      PUT(32) = CMQMC                                                     COMD0720
      PUT(33) = CLLP                                                      COMD0730
      PUT(34) = CLLRO                                                     COMD0740
      PUT(35) = CNPO                                                      COMD0750
      PUT(36) = CNRO                                                      COMD0760
      PUT(37) = CYPO                                                      COMD0770
      PUT(38) = CYRO                                                      COMD0780
C  STORE THE WING TOTAL STATIC AND DYNAMIC COEFFICIENTS ON UNIT 9         COMD0790
      CALL WRITMS(9,PUT,102,IREAD9)                                       COMD0800
C  READ THE WING COEFFICIENTS WITH VECTORED THRUST FROM UNIT 9            COMD0810
      IF(LOGIC .EQ. 1) GO TO 110                                          COMD0820
      IREAD9 = IREAD9 + NCC                                               COMD0830
      CALL READMS(9,PUT,102,IREAD9)                                       COMD0840
C  STORE THE WING TOTAL STATIC AND DYNAMIC COEFFICIENTS ON UNIT 9         COMD0850
      PUT(31) = CLQ                                                       COMD0860
      PUT(32) = CMQMC                                                     COMD0870
      PUT(33) = CLLP                                                      COMD0880
      PUT(34) = CLLRO                                                     COMD0890
                                                                          COMD0900
      PUT(35) = CNPO                                                      COMD0910
      PUT(36) = CNRO                                                      COMD0920
      PUT(37) = CYPO                                                      COMD0930
      PUT(38) = CYRO                                                      COMD0940
      CALL WRITMS(9,PUT,102,IREAD9)                                       COMD0950
  110 CONTINUE                                                            COMD0960
      RETURN                                                              COMD0970
      END                                                                 COMD0980
```

```
*DECK GEDER                                                               GEDR0010
      SUBROUTINE GEDER(DUMMY,ONSET,ONSETS,NEWMAX)                         GEDR0020
C                                                                         GEDR0030
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL DYNAMIC STABILITY          GEDR0040
C  DERIVATIVES IN GROUND PROXIMITY                                        GEDR0050
C                                                                         GEDR0060
      COMMON/SPIRIT/ NEWMX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                   GEDR0070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE              GEDR0080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),JW(40),IJ(40)  GEDR0090
      COMMON/COMPOS/FACTOR(10,24),NCC                                     GEDR0100
      COMMON/FCASE3/EPS(400,10),BETA(400,10),THETA(40,10),THS(40,10)      GEDR0110
      COMMON/SOLVI/GAMMAS(400,10)                                         GEDR0120
      COMMON/FUSLGI/PI,NSEG,NSEGI,NF(40),ISECT,ICAMF,XB(41),ZCL(41),      GEDR0130
     1              RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                 GEDR0140
     2              V(40,7),W(40,7)                                       GEDR0150
      DIMENSION DUMMY(NEWMAX),ONSET(NEWMAX),ONSETS(NEWMAX)                GEDR0160
      DIMENSION GAM(400),GAMR(400),U(400),UR(400),UQ(400),UP(400)         GEDR0170
      DIMENSION UA(400),UAR(400),GAMP(400),GAMA(400),GAMAR(400)           GEDR0180
      DIMENSION GAMMA(400,10),EP(400),BTA(400)                            GEDR0190
      DIMENSION PUT(102)                                                  GEDR0200
      NC1 = NCASES - 1                                                    GEDR0210
      IREAD = 2 * (NMAX + NJT)                                            GEDR0220
C  READ THE SOLUTION OF THE FIRST RUN                                     GEDR0230
      DO 30 N = 1,NCASES                                                  GEDR0240
      IREAD = IREAD + 1                                                   GEDR0250
      CALL READMS(1,DUMMY,NEWMAX,IREAD)                                   GEDR0260
      DO 20 I = 1,NEWMAX                                                  GEDR0270
   20 GAMMA(I,N) = DUMMY(I)                                               GEDR0280
   30 CONTINUE                                                            GEDR0290
C  CYCLE THROUGH ALL COMPOSITE CASES                                      GEDR0300
      DO 400 M = 1,NCC                                                    GEDR0310
      IREAD = -1                                                          GEDR0320
C  SUM UP THE VORTICITY                                                   GEDR0330
      DO 50 I = 1,NEWMAX                                                  GEDR0340
      GAM(I) = 0.0                                                        GEDR0350
      GAMR(I) = 0.0                                                       GEDR0360
      EP(I) = 0.0                                                         GEDR0370
      BTA(I) = 0.0                                                        GEDR0380
      DO 40 N = 1,NC1                                                     GEDR0390
      IF(FACTOR(N,M) .EQ. 0.0)GO TO 40                                    GEDR0400
      GAM(I) = GAM(I) + FACTOR(N,M) * GAMMA(I,N)                          GEDR0410
      GAMR(I) = GAMR(I) + FACTOR(N,M) * GAMMAS(I,N)                       GEDR0420
      EP(I) = EP(I) + FACTOR(N,M) * EPS(I,N)                              GEDR0430
      BTA(I) = BTA(I) + FACTOR(N,M) * BETA(I,N)                           GEDR0440
   40 CONTINUE                                                            GEDR0450
   50 CONTINUE                                                            GEDR0460
C  CALCULATE THE ONSET FLOW ARRAYS                                        GEDR0470
      DO 80 I = 1,NEWMAX                                                  GEDR0480
```

```
      U(I) = 0.0
      UR(I) = 0.0
      UQ(I) = 0.0
      UA(I) = 0.0
      UP(I) = 0.0
      UAR(I) = 0.0
      IREAD = IREAD + 2
      CALL READMS(9,ONSET,NEWMAX,IREAD)
      IRI = IREAD + 1
      CALL READMS(9,ONSETS,NEWMAX,IRI)
      DO 70 J = 1,NEWMAX
      UA(I) = UA(I) + ONSET(J) * GAMMA(J,1)
      UQ(I) = UQ(I) + ONSET(J) * GAMMA(J,NCASES)
      UAR(I) = UAR(I) + ONSETS(J) * GAMMAS(J,1)
      UP(I) = UP(I) + ONSETS(J) * GAMMAS(J,NCASES)
      DO 60  N = 1,NC1
      IF(FACTOR(N,M) .EQ. 0.0)GO TO 60
      U(I) = U(I) + ONSET(J) * FACTOR(N,M) * GAMMA(J,N)
      UR(I) = UR(I) + ONSETS(J) * FACTOR(N,M) * GAMMAS(J,N)
   60 CONTINUE
   70 CONTINUE
   80 CONTINUE
      IREAD9 = 2*NMAX + NSEG + M
C
C  CALCULATE THE PITCHING (Q) STABILITY DERIVATIVES
      DO 90  I = 1,NEWMAX
   90 DUMMY(I) = GAMMA(I,NCASES) * (1.0 + U(I)) + UQ(I) * GAM(I)
      CALL SUMIT3(CLQ0,CMQ0,DUMMY)
      DO 100 I = 1,NEWMAX
  100 DUMMY(I) = GAMMA(I,NCASES) * UA(I) + UQ(I) * GAMMA(I,1)
      CALL SUMIT3(CLQA,CMQA,DUMMY)
      DO 110 I = 1,NEWMAX
  110 DUMMY(I) = UQ(I) * GAMMA(I,NCASES)
      CALL SUMIT3(CLQ2,CMQ2,DUMMY)
C  CALCULATE THE ROLLING (P) STABILITY DERIVATIVES
      DO 120 I = 1,NEWMAX
      DUMMY(I) = GAMMAS(I,NCASES) * (1.0 + U(I)) + UP(I) * GAM(I)
  120 GAMP(I) = GAMMAS(I,NCASES)
      CONST = 2.0
      CALL SUMIT4(CLLP0,CNGP0,BLANK,CNSP0,CYP0,DUMMY,GAMP,GAM,CONST,EP)
      DO 130 I = 1,NEWMAX
      DUMMY(I) = GAMMAS(I,NCASES) * UA(I) + GAMMA(I,1) * UP(I)
  130 GAMA(I) = GAMMA(I,1)
      CALL SUMIT4(CLLPA,BLANK,CNGPA,CNSPA,CYPA,DUMMY,GAMA,GAMP,CONST,EP)
      IF(ISYMM .LT. 0) GO TO 150
      DO 140 I = 1,NEWMAX
  140 DUMMY(I) = UP(I) * GAMMAS(I,NCASES)
      CONST = 1.0
      CALL SUMIT4(CLLP2,BLANK,BLANK,CNSP2,CYP2,DUMMY,GAMP,GAMP,CONST,EP)
C  COMBINE THE DATA INTO THE COMPLETE COEFFICIENTS
  150 CNP0 = CNGP0 / 57.295779 + CNSP0
      CNPA = CNGPA + CNSPA
      CNP20 = CNSP2
      IF(ISYMM .GE. 0) GO TO 160
      CLLP2 = 0.0
      CNP20 = 0.0
      CYP2 = 0.0
C  CALCULATE THE YAWING (R) STABILITY DERIVATIVES
  160 DO 170 I = 1,NEWMAX
  170 DUMMY(I) = GAMR(I) * (1.0 + U(I)) + UR(I) * GAM(I)
      CONST = 2.0
      CALL SUMIT4(CLLR0,CNGR0,BLANK,CNSR0,CYR0,DUMMY,GAMA,GAM,CONST,EP)
      DO 180 I = 1,NEWMAX
      DUMMY(I) = GAMMAS(I,1) * (1.0 +  U(I)) + UA(I) * GAMR(I) +
     1           UAR(I) * GAM(I) + UR(I) * GAMMA(I,1)
  180 GAMAR(I) = GAMMAS(I,1)
      CALL SUMIT4(CLLRA,BLANK,CNGRAA,CNSRAA,CYRAA,DUMMY,GAMA,GAMR,
     1            CONST,EP)
C  MODIFY THE SUCTION PORTION OF THIS TERM
      CALL SUMIT4(CLLRA,CNGRAB,BLANK,CNSRAB,CYRAB,DUMMY,GAMAR,GAM,
     1            CONST,EP)
      DO 190 I = 1,NEWMAX
  190 DUMMY(I) = UA(I) * GAMMAS(I,1) + UAR(I) * GAMMA(I,1)
      CALL SUMIT4(CLLRA2,BLANK,CNGRA2,CNSRA2,CYRA2,DUMMY,GAMA,GAMAR,
     1            CONST,EP)
      IF(ISYMM .LT. 0) GO TO 230
      DO 200 I = 1,NEWMAX
  200 DUMMY(I) = UR(I) * GAMR(I)
      CONST = 1.0
      CALL SUMIT4(CLLR20,BLANK,BLANK,CNSR20,CYR20,DUMMY,GAMA,GAMR,
     1            CONST,EP)
      DO 210 I = 1,NEWMAX
  210 DUMMY(I) = UAR(I) * GAMR(I) + UR(I) * GAMMAS(I,1)
      CONST = 2.0
      CALL SUMIT4(CLLR2A,BLANK,BLANK,CNSR2A,CYR2A,DUMMY,GAMAR,GAMR,
     1            CONST,EP)
      DO 220 I = 1,NEWMAX
  220 DUMMY(I) = UAR(I) * GAMAR(I)
      CONST = 1.0
      CALL SUMIT4(CLR2A2 ,BLANK,BLANK,CSR2A2,CYR2A2,DUMMY,GAMAR,GAMAR,
     1            CONST,EP)
C  COMBINE THE DATA INTO THE COMPLETE COEFFICIENTS
  230 CNR0 = CNGR0 / 57.295779 + CNSR0
      CNRA = (CNGRAA + CNGRAB) / 57.295779 + CNSRAA + CNSRAB
      CNRA2 = CNGRA2 + CNSRA2
      CNR20 = CNSR20
      CNR2A = CNSR2A
      CNR2A2 = CSR2A2
      CYRA = CYRAA + CYRAB
      IF(ISYMM .GE. 0) GO TO 240
      CLLR20 = 0.0
      CLLR2A = 0.0
      CLR2A2 = 0.0
      CNR20 = 0.0
      CNR2A = 0.0
      CNR2A2 = 0.0
      CNR2A3 = 0.0
      CYR20  = 0.0
      CYR2A = 0.0
      CYR2A2 = 0.0
C  PRINT A TABLE OF ALL DYNAMIC STABILITY DERIVATIVES
  240 CALL HEADER(ILINES)
      WRITE(6, 250 )M
  250 FORMAT(1H0,27X,3H***,14(5H*****),/,
     1       28X,48H*  STABILITY DERIVATIVE DATA FOR COMPOSITE CASE ,I2,
     2       23H IN GROUND PROXIMITY  */,28X,3H***,14(5H*****))
      WRITE(6, 260 )(N,N=1,10),(FACTOR(N,M),N=1,10)
  260 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),
     1       3X),5X,2HA(,I2,1H)/10X,10F12.6)
      WRITE(6, 270 )CLQ0,CLQA,CLQ2
  270 FORMAT(1H0,30X,46HLIFT COEFFICIENT DUE TO PITCHING, CL(Q) MAY BE,
     1       22H CALCULATED AS FOLLOWS/53X,25HCL(Q) = CLQ*Q + CLQ2*Q**2/
     2       47X,30HWHERE  CLQ = CLQ0 + CLQA*ALPHA/55X, 7HCLQ0 = ,F12.7/
     3       55X,7HCLQA = ,F12.7/,55X,7HCLQ2 = ,F12.7/)
      WRITE(6, 280 )CMQ0,CMQA,CMQ2
  280 FORMAT(1H0,19X,47HPITCHING MOMENT COEFFICIENT ABOUT XMC DUE TO PI,
     1       52HTCHING ABOUT XCG, CM(Q) MAY BE CALCULATED AS FOLLOWS/
     2       53X,25HCM(Q) = CMQ*Q + CMQ2*Q**2/47X,
     3       30HWHERE  CMQ = CMQ0 + CMQA*ALPHA/55X,7HCMQ0 = ,F12.7/
     4       55X,7HCMQA = ,F12.7/,55X,7HCMQ2 = ,F12.7/)
      WRITE(6, 290 )CLLP0,CLLPA,CLLP2
  290 FORMAT(1H0,25X,46HROLLING MOMENT COEFFICIENT DUE TO ROLLING, CLL,
     1       32H(P) MAY BE CALCULATED AS FOLLOWS/51X,
     2       28HCLL(P) = CLLP*P + CLLP2*P**2/46X,
     3       33HWHERE  CLLP = CLLP0 + CLLPA*ALPHA/54X,8HCLLP0 = ,F12.7/
     4       54X,8HCLLPA = ,F12.7/54X,8HCLLP2 = ,F12.7/)
      WRITE(6,300 )CNP0,CNPA,CNP20
  300 FORMAT(1H0,27X,47HYAWING MOMENT COEFFICIENT DUE TO ROLLING, CN(P),
     1       29H MAY BE CALCULATED AS FOLLOWS/53X,
     2       25HCN(P) = CNP*P + CNP2*P**2/48X,
     3       30HWHERE  CNP = CNP0 + CNPA*ALPHA/55X,
     4       7HCNP0 = ,F12.7/55X,7HCNPA = ,F12.7/54X,8HCNP20 = ,F12.7/)
      WRITE(6, 310 )CYP0,CYPA,CYP2
  310 FORMAT(1H0,29X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),
     1       29H MAY BE CALCULATED AS FOLLOWS/53X,
     2       25HCY(P) = CYP*P + CYP2*P**2/47X,
     3       30HWHERE  CYP = CYP0 + CYPA*ALPHA/55X,7HCYP0 = ,F12.7/
     4       55X,7HCYPA = ,F12.7/55X,7HCYP2 = ,F12.7/)
      CALL HEADER(ILINES)
      WRITE(6, 320 )CLLR0,CLLRA,CLLRA2,CLLR20,CLLR2A,CLR2A2
  320 FORMAT(1H0,25X,45HROLLING MOMENT COEFFICIENT DUE TO YAWING, CLL,
     1       32H(R) MAY BE CALCULATED AS FOLLOWS/51X,
     2       28HCLL(R) = CLLR*R + CLLR2*R**2/37X,
     3       51HWHERE  CLLR = CLLR0 + CLLRA*ALPHA + CLLRA2*ALPHA**2/
     4       54X,8HCLLR0 = ,F12.7/54X,8HCLLRA = ,F12.7/53X,9HCLLRA2 = ,
     5       F12.7/36X,43HAND CLLR2 = CLLR20 + CLLR2A*ALPHA + CLLR2A2,
     6       9H*ALPHA**2/53X,9HCLLR20 = ,F12.7/53X,9HCLLR2A = ,F12.7/
     7       52X,10HCLLR2A2 = ,F12.7/)
      WRITE(6,330 )CNR0,CNRA,CNRA2,CNR20,CNR2A,CNR2A2
  330 FORMAT(1H0,27X,46HYAWING MOMENT COEFFICIENT DUE TO YAWING, CN(R),
     1       29H MAY BE CALCULATED AS FOLLOWS/53X,
     2       25HCN(R) = CNR*R + CNR2*R**2/39X,
     3       47HWHERE  CNR = CNR0 + CNRA*ALPHA + CNRA2*ALPHA**2/
     4       55X,7HCNR0 = ,F12.7/55X,7HCNRA = ,F12.7/54X,8HCNRA2 = ,
     5       F12.7/39X,40HAND  CNR2 = CNR20 + CNR2A*ALPHA + CNR2A2,
     6       9H*ALPHA**2/54X,8HCNR20 = ,F12.7/54X,8HCNR2A = ,F12.7/
     7       53X,9HCNR2A2 = ,F12.7/)
      WRITE(6, 340 )CYR0,CYRA,CYRA2,CYR20,CYR2A,CYR2A2
  340 FORMAT(1H0,29X,43HSIDE FORCE COEFFICIENT DUE TO YAWING, CY(R),
     1       29H MAY BE CALCULATED AS FOLLOWS/53X,
     2       25HCY(R) = CYR*R + CYR2*R**2/39X,
     3       47HWHERE  CYR = CYR0 + CYRA*ALPHA + CYRA2*ALPHA**2/
     4       55X,7HCYR0 = ,F12.7/55X,7HCYRA = ,F12.7/54X,8HCYRA2 = ,
     5    F12.7/39X,49HAND  CYR2 = CYR20 + CYR2A*ALPHA + CYR2A2*ALPHA**2/
     6       54X,8HCYR20 = ,F12.7/54X,8HCYR2A = ,F12.7/53X,9HCYR2A2 = ,
     7       F12.7/)
 PRINT A TABLE OF DERIVATIVE TERMS WHICH DEPEND ON ALPHA
      CALL HEADER(ILINES)
      WRITE(6, 350 ) M
  350 FORMAT(1H0, 31X, 11(5H*****) / 32X,
     1       55H*  VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK
     2       / 32X, 1H*, 6X,19HFOR COMPOSITE CASE , I2,
     3       20H IN GROUND PROXIMITY, 6X,1H*  / 32X, 11(5H*****) )
      WRITE(6, 360 ) (N,N=1,10),(FACTOR(N,M),N=1,10)
  360 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/10X,9(5X,2HA(,I1,1H),
     1       3X),5X,2HA(,I2,1H) / 10X,10F12.6)
      WRITE(6, 370 )
  370 FORMAT(1H0,5HALPHA, 2X, 3HCLQ, 7X, 3HCMQ, 8X, 4HCLLP, 6X, 3HCNP,
     1       7X, 4HCNP2, 6X, 3HCYP, 8X, 4HCLLR, 6X, 5HCLLR2, 5X, 3HCNR,
     2       7X, 4HCNR2, 6X, 3HCYR, 7X, 4HCYR2)
      A = -11.0
      DO 390 I = 1,41
      A = A + 1.0
      A2 = A * A
      CLQ = CLQ0 + CLQA*A
      CMQ = CMQ0 + CMQA*A
      CLLP = CLLP0 + CLLPA*A
      CNP = CNP0 + CNPA*A
      CNP2 = CNP20
      CYP = CYP0 + CYPA*A
      CLLR = CLLR0 + CLLRA*A + CLLRA2*A2
      CLLR2 = CLLR20 + CLLR2A*A + CLR2A2*A2
      CNR = CNR0 + CNRA*A + CNRA2*A2
      CNR2 = CNR20 + CNR2A*A + CNR2A2*A2
      CYR = CYR0 + CYRA*A + CYRA2*A2
      CYR2 = CYR20 + CYR2A*A + CYR2A2*A2
      WRITE(6, 380 ) A,CLQ,CMQ,CLLP,CNP,CNP2,CYP,CLLR,CLLR2,CNR,CNR2,
     1               CYR,CYR2
  380 FORMAT(1H ,F4.1,F11.7,F10.7,F11.7,3F10.7,F11.7,5F10.7)
  390 CONTINUE
C  READ THE WING TOTAL STATIC COEFFICIENTS
      IF(LOGIC .EQ. 3) GO TO 400
      CALL READMS(9,PUT,102,IREAD9)
C  DEFINE THE BASIC WING DYNAMIC COEFFICIENTS, ZERO ALPHA AND RATES ONLY
      PUT(31) = CLQ0
      PUT(32) = CMQ0
      PUT(33) = CLLP0
      PUT(34) = CLLR0
      PUT(35) = CNP0
      PUT(36) = CNR0
      PUT(37) = CYP0
      PUT(38) = CYR0
C  STORE THE WING TOTAL STATIC AND DYNAMIC COEFFICIENTS
      CALL WRITMS(9,PUT,102,IREAD9)
C  READ THE WING COEFFICIENTS WITH VECTORED THRUST
      IF(LOGIC .EQ. 1) GO TO 400
      IREAD9 = IREAD9 + NCC
      CALL READMS(9,PUT,102,IREAD9)
C  STORE THE WING TOTAL STATIC AND DYNAMIC COEFFICIENTS
      PUT(31) = CLQ0
      PUT(32) = CMQ0
      PUT(33) = CLLP0
      PUT(34) = CLLR0
      PUT(35) = CNP0
      PUT(36) = CNR0
      PUT(37) = CYP0
      PUT(38) = CYR0
      CALL WRITMS(9,PUT,102,IREAD9)
  400 CONTINUE
      RETURN
```

```
      END                                                               GEDR2970


*DECK SUMIT1
      SUBROUTINE SUMIT1(CPO,CPA,CPP,CPRO,CPRA,DEL,EPS,CHORD,Y,DELTA,CMU,SMT10010
     1  XL,TL,XMC,AREA,CNPO,CNPA,CNP2,CLLRO,CLLRA,CYPO,CYPA,CYP2,       SMT10020
     2  NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)                                SMT10030
C                                                                       SMT10040
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE    SMT10050
C  THE TERMS OF CNP AND CLLR DERIVATIVES                                SMT10060
C                                                                       SMT10070
      COMMON /DERIV/ UO(40),CLQ,CMQ,CMQMC                               SMT10080
      DIMENSION CPO(NEWMAX),CPA(NEWMAX),CPP(400),CPRO(400),CPRA(400)    SMT10090
      DIMENSION DEL(400),EPS(400)                                       SMT10100
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)         SMT10110
      DIMENSION NW(40),IJ(40)                                           SMT10120
C                                                                       SMT10130
C  INITIALIZE THE DERIVATIVE TERMS                                      SMT10140
   20 CNPO = 0.00                                                       SMT10150
      CNPA = 0.00                                                       SMT10160
      CNP2 = 0.00                                                       SMT10170
      CYPO = 0.00                                                       SMT10180
      CYPA = 0.00                                                       SMT10190
      CYP2 = 0.00                                                       SMT10200
      CLLRO = 0.00                                                      SMT10210
      CLLRA = 0.00                                                      SMT10220
C                                                                       SMT10230
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE                             SMT10240
      I = 0                                                             SMT10250
      DO 110 K = 1,NROWS                                                SMT10260
C                                                                       SMT10270
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE                            SMT10280
C  LEADING EDGE CONTRIBUTION                                            SMT10290
      I = I + 1                                                         SMT10300
C  YAWING DUE TO ROLLING                                                SMT10310
   30 TERMP = DEL(I) * (CPP(I) + 0.50*CPP(I+1))                         SMT10320
      EP = -EPS(I) / (57.295779 * UO(K))                                SMT10330
      DPO = TERMP * EP                                                  SMT10340
      DPA = TERMP                                                       SMT10350
      SPO = 0.3490658 * DEL(I) * CPP(I) * CPO(I)                        SMT10360
      SPA = 0.3490658 * DEL(I) * CPP(I) * CPA(I)                        SMT10370
      SP2 = 0.1745329 * DEL(I) * CPP(I)**2                              SMT10380
C  ROLLING DUE TO YAWING                                                SMT10390
   40 CLGO = DEL(I) * (CPRO(I) + 0.50*CPRO(I+1))                        SMT10400
      CLGA = DEL(I) * (CPRA(I) + 0.50*CPRA(I+1))                        SMT10410
C                                                                       SMT10420
C  REGULAR EVD CONTRIBUTIONS                                            SMT10430
   50 NWK = NW(K)                                                       SMT10440
      DO 80  L = 2,NWK                                                  SMT10450
      I = I + 1                                                         SMT10460
      CPP1 = CPP(I+1)                                                   SMT10470
      CPRO1 = CPRO(I+1)                                                 SMT10480
      CPRA1 = CPRA(I+1)                                                 SMT10490
      IF(L .LT. NWK) GO TO 60                                           SMT10500
      CPP1 = 0.00                                                       SMT10510
      CPRO1 = 0.00                                                      SMT10520
      CPRA1 = 0.00                                                      SMT10530
      IF(CMU(K) .LT. 0.0001) GO TO 60                                   SMT10540
                                                                        SMT10550
```

```
      IJK = IJ(K)                                                       SMT10560
      CPP1 = CPP(IJK)                                                   SMT10570
      CPRO1 = CPRO(IJK)                                                 SMT10580
      CPRA1 = CPRA(IJK)                                                 SMT10590
   60 EP = -EPS(I) / (57.295779 * UO(K))                                SMT10600
      TERMP = 0.50 * DEL(I) * (CPP(I) + CPP1)                           SMT10610
      DPO = DPO + TERMP * EP                                            SMT10620
      DPA = DPA + TERMP                                                 SMT10630
   70 CLGO = CLGO + 0.50 * DEL(I) * (CPRO(I) + CPRO1)                   SMT10640
      CLGA = CLGA + 0.50 * DEL(I) * (CPRA(I) + CPRA1)                   SMT10650
   80 CONTINUE                                                          SMT10660
C                                                                       SMT10670
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE                             SMT10680
      FACTOR = CHORD(K) * Y(K) * DELTA(K)                               SMT10690
      FACTO = CHORD(K) * DELTA(K) * TL(K)                               SMT10700
      FACT = FACTO * (XL(K)-XMC)                                        SMT10710
   90 CNPO = CNPO + (DPO - SPO) * FACTOR - SPO * FACT                   SMT10720
      CNPA = CNPA + (DPA/57.295779 - SPA) * FACTOR - SPA * FACT         SMT10730
      CNP2 = CNP2 + SP2 * FACTOR + SP2 * FACT                           SMT10740
      CYPO = CYPO + SPO * FACTO                                         SMT10750
      CYPA = CYPA + SPA * FACTO                                         SMT10760
      CYP2 = CYP2 + SP2 * FACTO                                         SMT10770
  100 CLLRO = CLLRO + CLGO * FACTOR                                     SMT10780
      CLLRA = CLLRA + CLGA * FACTOR                                     SMT10790
  110 CONTINUE                                                          SMT10800
C                                                                       SMT10810
C  PUT THE TERMS IN FINAL FORM                                          SMT10820
      FACTOR = 1.00 / AREA                                              SMT10830
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA                             SMT10840
  120 CNPO = FACTOR * CNPO                                              SMT10850
      CNPA = FACTOR * CNPA                                              SMT10860
      CNP2 = -FACTOR * CNP2                                             SMT10870
      CLLRO = -FACTOR * CLLRO                                           SMT10880
      CLLRA = -FACTOR * CLLRA                                           SMT10890
      FACTOR = 2.00 * FACTOR                                            SMT10900
      CYPO = FACTOR * CYPO                                              SMT10910
      CYPA = FACTOR * CYPA                                              SMT10920
      CYP2 = FACTOR * CYP2                                              SMT10930
      IF(IFUDGE .NE. 0) GO TO 130                                       SMT10940
      CNPO = 0.00                                                       SMT10950
      CYPO = 0.00                                                       SMT10960
  130 IF(ISYMM .GE. 0) GO TO 140                                        SMT10970
      CNP2 = 0.00                                                       SMT10980
      CYP2 = 0.00                                                       SMT10990
  140 RETURN                                                            SMT11000
      END                                                               SMT11010


*DECK SUMIT2
      SUBROUTINE SUMIT2(CPO,CPA,CPRO,CPRA,DEL,EPS,CHORD,Y,DELTA,AREA,   SMT20010
     1  CMV,XL,TL,XMC,CNRO,CNRA,CNRA2,CNR2O,CNR2A,CNR2A2,CYRO,CYRA,     SMT20020
     2  CYRA2,CYR2O,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)              SMT20030
C                                                                       SMT20040
C  THIS SUBROUTINE INTEGRATES CP CHO DWISE AND SPANWISE TO CALCULATE    SMT20050
C  THE TERMS OF THE YAWING AND SIDE FORCE COEFFICIENTS DUE TO YAWING    SMT20060
C                                                                       SMT20070
      COMMON /DERIV/ U4(40),CLQ,CMQ,CMQMC                               SMT20080
      DIMENSION CPO(NEWMAX),CPA(NEWMAX),CPRO(400),CPRA(400)             SMT20090
                                                                        SMT20100
```

```
      DIMENSION DEL(400),EPS(400)                                       SMT20110
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)         SMT20120
      DIMENSION NW(40),IJ(40)                                           SMT20130
C                                                                       SMT20140
C  INITIALIZE THE YAWING COEFFICIENT TERMS                              SMT20150
   20 CNRO = 0.00                                                       SMT20160
      CNRA = 0.00                                                       SMT20170
      CNRA2 = 0.00                                                      SMT20180
      CNR2O = 0.00                                                      SMT20190
      CNR2A = 0.00                                                      SMT20200
      CNR2A2 = 0.00                                                     SMT20210
      CYRO = 0.00                                                       SMT20220
      CYRA = 0.00                                                       SMT20230
      CYRA2 = 0.00                                                      SMT20240
      CYR2O = 0.00                                                      SMT20250
      CYR2A = 0.00                                                      SMT20260
      CYR2A2 = 0.00                                                     SMT20270
      I = 0                                                             SMT20280
      DO 110 K = 1,NROWS                                                SMT20290
C  INTEGRATE THE COEFFICIENT TERMS CHOROWISE                            SMT20300
C  LEADING EDGE CONTRIBUTIONS                                           SMT20310
      I = I + 1                                                         SMT20320
   30 TERMO = DEL(I) * (CPRO(I) + 0.50*CPRO(I+1))                       SMT20330
      TERMA = DEL(I) * (CPRA(I) + 0.50*CPRA(I+1))                       SMT20340
      EP = -EPS(I) / (57.295779 * UO(K))                                SMT20350
      DRO = TERMO * EP                                                  SMT20360
      DRA = TERMO/57.295779 + TERMA * EP                                SMT20370
      DRA2 = TERMA                                                      SMT20380
   40 SRO = 0.3490658 * DEL(I) * DPRO(I) * CPO(I)                       SMT20390
      SRA = 0.3490658 * DEL(I) * (CPRO(I)*CPA(I) + CPRA(I)*CPO(I))      SMT20400
      SRA2 = 0.3490658 * DEL(I) * (CPRA(I)*CPA(I))                      SMT20410
      IF(ISYMM .LT. 0) GO TO 50                                         SMT20420
      SR2O = 0.1745329 * DEL(I) * CPRO(I)**2                            SMT20430
      SR2A = 0.3490658 * DEL(I) * CPRO(I)*CPRA(I)                       SMT20440
      SR2A2 = 0.1745329 * DEL(I) * CPRA(I)**2                           SMT20450
C  REGULAR EVD CONTRIBUTIONS                                            SMT20460
   50 NWK = NW(K)                                                       SMT20470
      DO 80  L = 2,NWK                                                  SMT20480
      I = I + 1                                                         SMT20490
      CPRO1 = CPRO(I+1)                                                 SMT20500
      CPRA1 = CPRA(I+1)                                                 SMT20510
      IF(L .LT. NWK) GO TO 60                                           SMT20520
      CPRO1 = 0.00                                                      SMT20530
      CPRA1 = 0.00                                                      SMT20540
      IF(CMU(K) .LT. 0.0001) GO TO 60                                   SMT20550
      IJK = IJ(K)                                                       SMT20560
      CPRO1 = CPRO(IJK)                                                 SMT20570
      CPRA1 = CPRA(IJK)                                                 SMT20580
   60 EP = -EPS(I) / (57.295779 * UO(K))                                SMT20590
C  REGULAR EVD CONTRIBUTIONS                                            SMT20600
      TERMO = 0.50 * DEL(I) * (CPRO(I)+CPRO1)                           SMT20610
   70 TERMA = 0.50 * DEL(I) * (CPRA(I)+CPRA1)                           SMT20620
      DRO = DRO + TERMO * EP                                            SMT20630
      DRA = DRA + TERMO/57.295779 + TERMA * EP                          SMT20640
      DRA2 = DRA2 + TERMA                                               SMT20650
   80 CONTINUE                                                          SMT20660
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE                             SMT20670
      FACTOR = CHORD(K) * Y(K) * DELTA(K)                               SMT20680
      FACTO = CHORD(K) * TL(K) * DELTA(K)                               SMT20690
      FACT = FACTO * (XL(K)-XMC)                                        SMT20700
```

```
   90 CNRO = CNRO + (DRO-SRO) * FACTOR - SRO * FACT                     SMT20710
      CNRA = CNRA + (DRA-SRA) * FACTOR - SRA * FACT                     SMT20720
      CNRA2 = CNRA2 + (DRA2/57.295779-SRA2) * FACTOR - SRA2 * FACT      SMT20730
      CYRO = CYRO + SRO * FACTO                                         SMT20740
      CYRA = CYRA + SRA * FACTO                                         SMT20750
      CYRA2 = CYRA2 + SRA2 * FACTO                                      SMT20760
      IF(ISYMM .LT. 0) GO TO 110                                        SMT20770
  100 CNR2O = CNR2O - SR2O * FACTOR -SR2O * FACT                        SMT20780
      CNR2A = CNR2A - SR2A * FACTOR - SR2A * FACT                       SMT20790
      CNR2A2 = CNR2A2 - SR2A2 * FACTOR - SR2A2 * FACT                   SMT20800
      CYR2O = CYR2O + SR2O * FACTO                                      SMT20810
      CYR2A = CYR2A + SR2A * FACTO                                      SMT20820
      CYR2A2 = CYR2A2 + SR2A2 * FACTO                                   SMT20830
  110 CONTINUE                                                          SMT20840
C  PUT THE TERMS IN FINAL FORM                                          SMT20850
      FACTOR = 1.00 / AREA                                              SMT20860
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA                             SMT20870
  120 CNRO = FACTOR * CNRO                                              SMT20880
      CNRA = FACTOR * CNRA                                              SMT20890
      CNRA2 = FACTOR * CNRA2                                            SMT20900
      CNR2O = FACTOR * CNR2O                                            SMT20910
      CNR2A = FACTOR * CNR2A                                            SMT20920
      CNR2A2 = FACTOR * CNR2A2                                          SMT20930
      FACTOR = 2.00 * FACTOR                                            SMT20940
      CYRO = FACTOR * CYRO                                              SMT20950
      CYRA = FACTOR .* CYRA                                             SMT20960
      CYRA2 = FACTOR * CYRA2                                            SMT20970
      CYR2O = FACTOR * CYR2O                                            SMT20980
      CYR2A = FACTOR * CYR2A                                            SMT20990
      CYR2A2 = FACTOR * CYR2A2                                          SMT21000
  130 RETURN                                                            SMT21010
      END                                                               SMT21020


*DECK SUMIT3                                                            SMT30010
      SUBROUTINE SUMIT3(CL,CM,CP)                                       SMT30020
C  THIS SUBROUTINE INTEGRATES THE DYNAMIC STABILITY DERIVATIVES FOR     SMT30030
C  PITCH IN GROUND PROXIMITY                                            SMT30040
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) SMT30050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            SMT30060
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  SMT30070
     1             D(40),KK(400),ITYPE(400)                             SMT30080
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           SMT30090
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     SMT30100
      DIMENSION CP(400)                                                 SMT30110
      I = 0                                                             SMT30120
      CL = 0.0                                                          SMT30130
      CM = 0.0                                                          SMT30140
      DO 50 K = 1,NROWS                                                 SMT30150
      I = I + 1                                                         SMT30160
      CLK = DEL(I) * (CP(I) + 0.5 * CP(I+1))                            SMT30170
      CMK = -DEL(I)**2 * (CP(I) + CP(I+1)/3.0) / 3.0                    SMT30180
      NWK = NW(K)                                                       SMT30190
      DO 40 L = 2,NWK                                                   SMT30200
      I = I + 1                                                         SMT30210
      CPI = CP(I)                                                       SMT30220
      IF(L .EQ. NWK)GO TO 20                                            SMT30230
      CPI1 = CP(I+1)                                                    SMT30240
```

```
      GO TO 30                                                          SMT30250
   20 CPI1 = 0.0                                                        SMT30260
      IF(CMU(K) .GT. 0.0)CPI1 = CP(IJ(K))                               SMT30270
   30 CLI = 0.5 * DEL(I) * (CPI + CPI1)                                 SMT30280
      CLK = CLK + CLI                                                   SMT30290
      CMK = CMK - CLI*XB(I) - (CPI+2.0*CPI1)*DEL(I)**2/6.00             SMT30300
   40 CONTINUE                                                          SMT30310
      CDEL = CHORD(K) * DELTA(K)                                        SMT30320
      CCDEL = CDEL * CHORD(K)                                           SMT30330
      CL = CL + CLK * CDEL                                              SMT30340
      CM = CM + CMK * CCDEL                                             SMT30350
   50 CONTINUE                                                          SMT30360
      FACTOR = 4.0 / AREA                                               SMT30370
      IF(ISYMM .LT. 1)FACTOR = 8.0 / AREA                               SMT30380
      CL = CL * FACTOR                                                  SMT30390
      CM = (CM * FACTOR + CL * XMC) / CREF                              SMT30400
      RETURN                                                            SMT30410
      END                                                               SMT30420
```

```
*DECK SUMIT4                                                            SMT40010
      SUBROUTINE SUMIT4(CLL,CNG1,CNG2,CNS,CY,CP,GAM1,GAM2,CONST,EPS)    SMT40020
C  THIS SUBROUTINE INTEGRATES THE DYNAMIC STABILITY DERIVATIVES FOR     SMT40030
C  ROLL AND YAW IN GROUND PROXIMITY                                     SMT40040
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) SMT40050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            SMT40060
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XB(400),XI(400),DEL(400),  SMT40070
     1             D(40),KK(400),ITYPE(400)                             SMT40080
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                           SMT40090
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             SMT40100
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     SMT40110
      DIMENSION CP(400),GAM1(400),GAM2(400),EPS(400)                    SMT40120
      I = 0                                                             SMT40130
      CLL = 0.0                                                         SMT40140
      CNG1 = 0.0                                                        SMT40150
      CNG2 = 0.0                                                        SMT40160
      CNS = 0.0                                                         SMT40170
      CY = 0.0                                                          SMT40180
      DO 50 K = 1,NROWS                                                 SMT40190
      I = I + 1                                                         SMT40200
      CLLK = DEL(I) * (CP(I) + 0.5 * CP(I+1))                           SMT40210
      CNG1K = -(DEL(I) * (GAM1(I) + 0.5 * GAM1(I+1))) * EPS(I) / Y(K)   SMT40220
      CNG2K = (DEL(I) * (GAM2(I) + 0.5 * GAM2(I+1))) / 57.295779        SMT40230
      CS = 0.698131б * DEL(I) * CONST * GAM1(I) * GAM2(I)               SMT40240
      NWK = NW(K)                                                       SMT40250
      DO 40 L = 2,NWK                                                   SMT40260
      I = I + 1                                                         SMT40270
      CPI = CP(I)                                                       SMT40280
      G1I = GAM1(I)                                                     SMT40290
      G2I = GAM2(I)                                                     SMT40300
      IF(L .EQ. NWK)GO TO 20                                            SMT40310
      CPI1 = CP(I+1)                                                    SMT40320
      G1I1 = GAM1(I+1)                                                  SMT40330
      G2I1 = GAM2(I+1)                                                  SMT40340
      GO TO 30                                                          SMT40350
   20 CPI1 = 0.0                                                        SMT40360
      G1I1 = 0.0                                                        SMT40370
      G2I1 = 0.0                                                        SMT40380
      IF(CMU(K) .LT. 0.0001) GO TO 30                                   SMT40390
      CPI1 = CP(IJ(K))                                                  SMT40400
      G1I1 = GAM1(IJ(K))                                                SMT40410
      G2I1 = GAM2(IJ(K))                                                SMT40420
   30 CLLI = 0.5 * DEL(I) * (CPI + CPI1)                                SMT40430
      CLLK = CLLK + CLLI                                                SMT40440
      CNG1K = CNG1K - (0.5 * DEL(I) * (G1I + G1I1)) * EPS(I) / Y(K)     SMT40450
      CNG2K = CNG2K + (0.5 * DEL(I) * (G2I + G2I1)) / 57.295779         SMT40460
   40 CONTINUE                                                          SMT40470
      FACTOR = CHORD(K) * DELTA(K) * Y(K)                               SMT40480
      FACTO = CHORD(K) * DELTA(K) * TANLE(K)                            SMT40490
      FACT = FACTO * (XLEAD(K) - XMC)                                   SMT40500
      CLL = CLL - CLLK * FACTOR                                         SMT40510
      CNG1 = CNG1 + CNG1K * FACTOR                                      SMT40520
      CNG2 = CNG2 + CNG2K * FACTOR                                      SMT40530
      CNS = CNS - CS * (FACTOR + FACT)                                  SMT40540
      CY = CY + CS * FACTO                                              SMT40550
   50 CONTINUE                                                          SMT40560
      FACTOR = 2.0 / AREA                                               SMT40570
      IF(ISYMM .LT. 0)FACTOR = 4.0 / AREA                               SMT40580
      CLL = CLL * FACTOR                                                SMT40590
      CNG1 = CNG1 * FACTOR                                              SMT40600
      CNG2 = CNG2 * FACTOR                                              SMT40610
      CNS = CNS * FACTOR / 2.0                                          SMT40620
      CY = CY * FACTOR                                                  SMT40630
      RETURN                                                            SMT40640
      END                                                               SMT40650
```

```
*DECK STABLE
      SUBROUTINE STABLE(CLQ,CMQ,CMQMC,CLLP,CNPO,CNPA,CNP2,CYPO,CYPA,    STAB0010
     1       CYP2,CLLRO,CLLRA,CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2,       STAB0020
     2        CYRO,CYRA,CYRA2,CYR20,CYR2A,CYR2A2,MCASE)                 STAB0030
C                                                                       STAB0040
C  THIS SUBROUTINE CALCULATES AND PRINTS A COMPLETE SUMMARY TABLE       STAB0050
C  OF ALL STABILITY DERIVATIVE DATA FOR EACH COMPOSITE CASE             STAB0060
C                                                                       STAB0070
      COMMON/COMPOS/FACTOR(10,24),NCC                                   STAB0080
C                                                                       STAB0090
C  PRINT ALL CONSTANT DERIVATIVES                                       STAB0100
      CALL HEADER(ILINES)                                               STAB0110
      WRITE(6, 20 ) MCASE                                               STAB0120
   20 FORMAT(1H0,33X,13(4H****),1H* /  34X,                             STAB0130
     1        47H*  STABILITY DERIVATIVE DATA FOR COMPOSITE CASE, I3,   STAB0140
     2        3H  */ 34X,13(4H****),1H* /)                              STAB0150
      WRITE(6, 30 ) (N,N=1,10), (FACTOR(N,MCASE),N=1,10)                STAB0160
   30 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/ 10X,9(4X,2HA(,I1,1H), STAB0170
     1     2X),3X,2HA(,I2,1H) / 10X,10F10.6)                            STAB0180
      WRITE(6, 40 ) CLQ,CMQ,CMQMC                                       STAB0190
   40 FORMAT(1H0/ 26X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,   STAB0200
     1        17H ABOUT XCG, CLQ =,F10.6 /                              STAB0210
     2        14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,STAB0220
     3        33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /              STAB0230
     4        16X,42HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,        STAB0240
     5        35H DUE TO PITCHING ABOUT XCG, CMQMC =, F10.6)            STAB0250
      WRITE(6, 50 ) CLLP,CNPO,CNPA,CNP2                                 STAB0260
   50 FORMAT(1H0/28X,38HROLLING MOMENT COEFF DERIVATIVE DUE TO,         STAB0270
     1        16H ROLLING, CLLP =,F12.7 ///                             STAB0280
     2        17X,51HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING,,STAB0290
     3        35H CN(P) MAY BE CALCULATED AS FOLLOWS /                  STAB0300
     4        48X,25HCN(P) = CNP*P + CNP2*P**2 /                        STAB0310
     5        42X,31HWHERE   CNP = CNPO + CNPA*ALPHA /                  STAB0320
     6        52X, 6HCNPO =,F12.7/52X, 6HCNPA =,F12.7/52X,6HCNP2 =,F12.7)STAB0340
      WRITE(6, 60 ) CYPO,CYPA,CYP2                                      STAB0350
   60 FORMAT(1H0/26X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),   STAB0360
     1        29H MAY BE CALCULATED AS FOLLOWS/                         STAB0370
     2        48X,25HCY(P) = CYP*P + CYP2*P**2/                         STAB0380
     3        42X,31HWHERE   CYP = CYPO + CYPA*ALPHA/                   STAB0390
     4        52X, 6HCYPO =,F12.7/52X,6HCYPA =,F12.7/52X,6HCYP2 =,F12.7) STAB0400
      WRITE(6, 70 ) CLLRO,CLLRA                                         STAB0410
   70 FORMAT(1H0/15X,38HROLLING MOMENT COEFF DERIVATIVE DUE TO,         STAB0420
     1        52H YAWING ABOUT XCG, CLLR MAY BE CALCULATED AS FOLLOWS / STAB0430
     2        48X,26HCLLR = CLLRO + CLLRA*ALPHA /                       STAB0440
     3        48X,15HWHERE   CLLRO =,F13.7 / 56X,7HCLLRA =,F13.7)       STAB0450
      WRITE(6, 80 ) CNRO,CNRA,CNRA2,CNR20,CNR2A,CNR2A2                  STAB0460
   80 FORMAT(1H0/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,     STAB0470
     1        53H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS / STAB0480
     2        48X,25HCN(R) = CNR*R + CNR2*R**2 /                        STAB0490
     3        36X,48HWHERE   CNR = CNRO + CNRA*ALPHA + CNRA2*ALPHA**2 / STAB0500
     4        52X,6HCNRO =,F13.7/52X,6HCNRA =,F13.7/51X,7HCNRA2 =,F13.7/ STAB0510
     5     36X,51HAND    CNR2 = CNR20 + CNR2A*ALPHA + CNR2A2*ALPHA**2/  STAB0520
     6   51X,7HCNR20 =,F13.7/51X,7HCNR2A =,F13.7/50X,8HCNR2A2 =,F13.7)  STAB0530
      WRITE(6, 90 ) CYRO,CYRA,CYRA2,CYR20,CYR2A,CYR2A2                  STAB0540
   90 FORMAT(1H0/13X,39HSIDE FORCE COEFFICIENT ABOUT XMC DUE TO,        STAB0550
     1        53H YAWING ABOUT XCG, CY(R) MAY BE CALCULATED AS FOLLOWS / STAB0560
     2        48X,25HCY(R) = CYR*R + CYR2*R**2 /                        STAB0570
     3        36X,48HWHERE   CYR = CYRO + CYRA*ALPHA + CYRA2*ALPHA**2 / STAB0580
     4        52X,6HCYRO =,F13.7/52X,6HCYRA =,F13.7/51X,7HCYRA2 =,F13.7/ STAB0590
     5     36X,51HAND    CYR2 = CYR20 + CYR2A*ALPHA + CYR2A2*ALPHA**2/  STAB0600
     6   51X,7HCYR20 =,F13.7/51X,7HCYR2A =,F13.7/50X,8HCYR2A2 =,F13.7)  STAB0610
C  PRINT TABLE OF DERIVATIVE TERMS WHICH DEPEND ON ALPHA                STAB0620
      CALL HEADER(ILINES)                                               STAB0630
      WRITE(6, 100 )                                                    STAB0640
  100 FORMAT(1H0,32X,3H***,10(5H*****),/,                               STAB0650
     1 33X,53H* VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK *,/,  STAB0660
     2 33X,3H***,10(5H*****),//,                                        STAB0670
     3 9X,5HALPHA,12X,3HCNP,10X,4HCNP2,14X,3HCYP,10X,4HCYP2,            STAB0680
     4 14X,4HCLLR, 9X,3HCNR,10X,4HCNR2)                                 STAB0690
      ALPHA = -11.00                                                    STAB0700
      DO 150 M = 1,41                                                   STAB0710
  110 ALPHA = ALPHA + 1.00                                              STAB0720
      CNP = CNPO + CNPA * ALPHA                                         STAB0730
      CYP = CYPO + CYPA * ALPHA                                         STAB0740
  120 CLLR = CLLRO + CLLRA * ALPHA                                      STAB0750
      CNR = CNRO + CNRA * ALPHA + CNRA2 * ALPHA**2                      STAB0760
  130 CNR2 = CNR20 + CNR2A * ALPHA + CNR2A2 * ALPHA**2                  STAB0770
      WRITE(6, 140 ) ALPHA,CNP,CNP2,CYP,CYP2,CLLR,CNR,CNR2              STAB0780
  140 FORMAT(1H , 5X,F10.6,5H  *  ,2F13.7,5H  *  ,2F13.7,5H             STAB0790
     1           3F13.7)                                                STAB0800
  150 CONTINUE                                                          STAB0810
      RETURN                                                            STAB0820
      END                                                               STAB0830
```

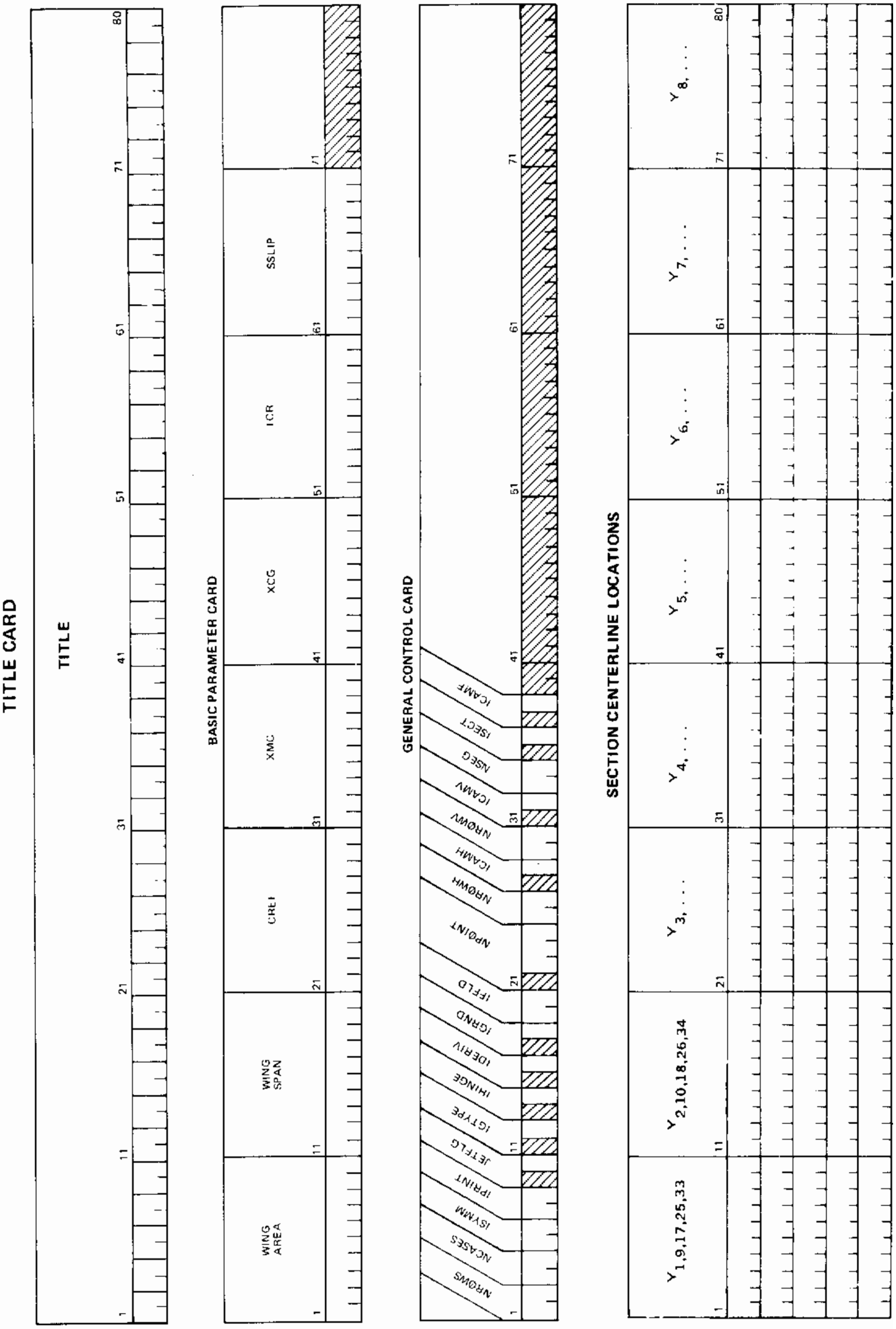
```
*DECK STAG12                                                            ST120010
      OVERLAY(OVLY,14,0)                                                ST120020
```

```
      PROGRAM STAG12                                                    ST120030
C                                                                       ST120040
C  THIS SUBROUTINE RETREIVES FROM UNIT 9 ALL THE STATIC AND DYNAMIC     ST120050
C  AERODYNAMIC COEFFICIENTS FOR THE WING, TAIL AND FUSELAGE, AND PRINTS ST120060
C  A TABLE FOR FIVE ANGLES OF ATTACK FOR EACH COMPONENT AND THE         ST120070
C  COMPLETE AIRCRAFT                                                    ST120080
C                                                                       ST120090
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR,ISTAB                ST120100
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE            ST120110
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) ST120120
      COMMON/COMPOS/FACTOR(10,24),NCC                                   ST120130
      COMMON/GEFECT/GHITE(10),IGRND,IGND,NHITE                          ST120140
      COMMON/TAIL1/ARH,TRH,SWH,SPANH,ARV,TRV,SWV,SPANV,HL,HH,VL,VH,     ST120150
     1            NMAXH,NMAXV,NTOTAL,NTPNP,NSIDE,NROWH,NROWV,ICAMH,     ST120160
     2            ICAMV,NXTYPE,NCTYPE,ISENSE,KCASE,ICASES               ST120170
      COMMON/FUSLG1/PI,NSEG,NSEG1,NF(40),ISECT,ICAMF,XB(41),ZCL(41),    ST120180
     1             RADIUS(41),AMAJ(41),AMIN(41),CSA(41),                ST120190
     2             V(40,7),W(40,7)                                      ST120200
      COMMON/KEEPEM/CLW(5),CMW(5),CLLW(5),CDW(5),CNW(5),CYW(5),CLQW,    ST120210
     1        CMQW,CLLPW,CLLRW,CNPW,CNRW,CYPW,CYRW,CLH(5),CMH(5),CLLH(5),ST120220
     2        CDH(5),CLQH,CMQH,CLLPH,CYV(5),CYBV,CNV(5),CNBV,CLLV(5),   ST120230
     3        CLLBV,CDV(5),CLLPV,CLLRV,CNPV,CNRV,CYPV,CYRV,CMF(5),CNF(5),ST120240
     4        CMQF,CNRF                                                 ST120250
      DIMENSION SUML(5),SUMM(5),SUMLL(5),SUMD(5),SUMN(5),SUMY(5)        ST120260
      LOGICAL NOSTAB,VT                                                 ST120270
      NOSTAB = ISTAB .EQ. 0                                             ST120280
      VT = JETFLG .EQ. 2                                                ST120290
C                                                                       ST120300
      IPASS = 0                                                         ST120310
      NCMU = NEWCMU                                                     ST120330
C  FIND THE PLACE TO BEGIN READING THE STORED COEFFICIENTS              ST120340
      IREAD = 2*NMAX + NSEG + 1                                         ST120350
   20 CONTINUE                                                          ST120360
      IPASS = IPASS + 1                                                 ST120370
      DO 330 M = 1,NCC                                                  ST120380
      IF((IPASS .EQ. 1) .AND. VT) NEWCMU = 1                            ST120390
   30 CALL HEADER(ILINES)                                               ST120400
      NEWCMU = NCMU                                                     ST120410
      WRITE(6, 40 ) M                                                   ST120420
   40 FORMAT(1H0  23X,18(4H****) /                                      ST120430
     1        24X,34H*  SUMMARY OF AIRCRAFT AERODYNAMIC,                ST120440
     2        32H COEFFICIENTS FOR COMPOSITE CASE,I3,3H  * /            ST120450
     3        24X,18(4H****))                                           ST120460
C  READ THE DATA FOR THIS COMPOSITE CASE                                ST120470
      CALL READMS(9,CLW,102,IREAD)                                      ST120480
C  FIND THE NEXT RECORD                                                 ST120490
      IREAD = IREAD + 1                                                 ST120500
C  WING                                                                 ST120510
      WRITE(6, 50 )                                                     ST120520
   50 FORMAT(1H0/ 54X,12H*** WING ***)                                  ST120530
      WRITE(6, 60 )                                                     ST120540
   60 FORMAT(1H ,18X, 5HALPHA, 6X,3HCCL,10X,4HCMMC,9X,3HCLL,10X,5HCOITZ,ST120550
     1     8X,2HCN,11X,3HCCY)                                           ST120560
      IF(IPASS .GT. 1) GO TO 100                                        ST120570
C  WITH VECTORED THRUST POWER OFF                                       ST120580
      DO 90 N = 1,5                                                     ST120590
      ALPHA = 5 * (N-1)                                                 ST120600
      A2 = ALPHA * ALPHA                                                ST120610
      CL = CLW(1) + CLW(2)*ALPHA                                        ST120620
      CM = CMW(1) + CMW(2)*ALPHA                                        ST120630
```

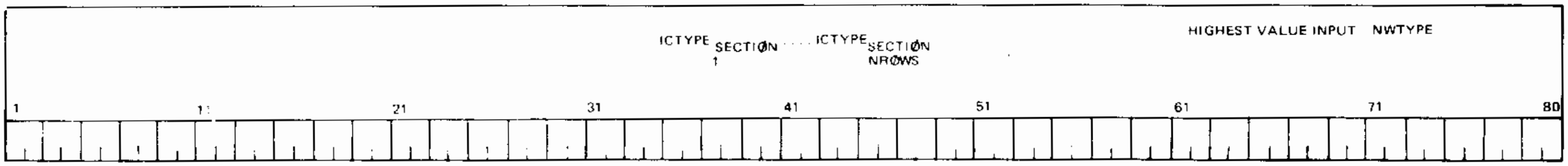
```
      CLL = CLLW(1) + CLLW(2)*ALPHA
      CD = CDW(1) + CDW(2)*ALPHA + CDW(3)*A2
      CN = CNW(1) + CNW(2)*ALPHA + CNW(3)*A2
      CY = CYW(1) + CYW(2)*ALPHA + CYW(3)*A2
      IF(IGND .EQ. 0) GO TO 70
      CL = CL + CLW(3)*A2
      CM = CM + CMW(3)*A2
      CLL = CLL + CLLW(3)*A2
   70 WRITE(6, 80 ) ALPHA,CL,CM,CLL,CD,CN,CY
   80 FORMAT(1H ,18X,F4.1,6F13.6)
      SUML(N) = CL
      SUMM(N) = CM
      SUMLL(N) = CLL
      SUMD(N) = CD
      SUMN(N) = CN
      SUMY(N) = CY
   90 CONTINUE
      GO TO 140
C  WITH VECTORED THRUST POWER ON
  100 DO 110 N = 1,5
      ALPHA = 5 * (N-1)
      WRITE(6, 80 ) ALPHA,CLW(N),CMW(N),CLLW(N),CDW(N),CNW(N),CYW(N)
      SUML(N) = CLW(N)
      SUMM(N) = CMW(N)
      SUMLL(N) = CLLW(N)
      SUMD(N) = CDW(N)
      SUMN(N) = CNW(N)
      SUMY(N) = CYW(N)
  110 CONTINUE
      IF(NOSTAB) GO TO 140
      WRITE(6, 120 )
  120 FORMAT(1H0,16X, 3HCLQ,10X, 3HCMQ,10X, 4HCLLP, 9X, 4HCLLR, 9X,
     1        3HCNP,10X, 3HCNR,10X, 3HCYP,10X, 3HCYR)
      WRITE(6, 130 ) CLQW,CMQW,CLLPW,CLLRW,CNPW,CNRW,CYPW,CYRW
  130 FORMAT(1H , 9X,8F13.6)
C  HORIZONTAL TAIL
  140 IF(NROWH  .EQ. 0) GO TO 180
      WRITE(6, 150 )
  150 FORMAT(1H0/ 49X,23H*** HORIZONTAL TAIL ***)
      WRITE(6, 160 )
  160 FORMAT(1H ,18X,5HALPHA,6X,3HCCL,10X,4HCMMC, 9X,3HCLL,10X,4HCCDI,
     1        9X,2HCN,11X,3HCCY)
      DO 170 N = 1,5
      ALPHA = 5 * (N-1)
      WRITE(6, 80 ) ALPHA,CLH(N),CMH(N),CLLH(N),CDH(N)
  170 CONTINUE
      IF(NOSTAB) GO TO 180
      WRITE(6, 120 )
      WRITE(6, 130 ) CLQH,CMQH,CLLPH
C  VERTICAL TAIL
  180 IF(NROWV .EQ. 0) GO TO 240
      WRITE(6, 190 )
  190 FORMAT(1H0/ 50X,21H*** VERTICAL TAIL ***)
      WRITE(6, 160 )
      DO 210 N = 1,5
      ALPHA = 5 * (N-1)
      WRITE(6, 200 ) ALPHA,CLLV(N),CDV(N),CNV(N),CYV(N)
  200 FORMAT(1H ,18X,F4.1,26X,4F13.6)
  210 CONTINUE
      IF(NOSTAB) GO TO 240
```

```
      WRITE(6, 220 ) CYBV,CNBV,CLLBV
  220 FORMAT(1H0,42X, 3HCYB,10X, 3HCNB,10X, 4HCLLB / 36X,3F13.6)
      WRITE(6, 120 )
      WRITE(6, 230 ) CLLPV,CLLRV,CNPV,CNRV,CYPV,CYRV
  230 FORMAT(1H ,35X,6F13.6)
C  FUSELAGE
  240 IF(NSEG .EQ. 0) GO TO 290
      WRITE(6, 250 )
  250 FORMAT(1H0/ 52X,16H*** FUSELAGE ***)
      WRITE(6, 160 )
      DO 270 N = 1,5
      ALPHA = 5 * (N-1)
      WRITE(6, 260 ) ALPHA,CMF(N),CNF(N)
  260 FORMAT(1H ,18X,F4.1,13X,F13.6,26X,F13.6)
  270 CONTINUE
      IF(NOSTAB) GO TO 290
      WRITE(6, 120 )
      WRITE(6, 280 ) CMQF,CNRF
  280 FORMAT(1H ,22X,F13.6,39X,F13.6)
C  SUM UP ALL THE COMPONENT STATIC COEFFICIENTS
  290 WRITE(6, 300 )
  300 FORMAT(1H1/ 50X,5(4H****),1H* / 50X,21H* COMPLETE AIRCRAFT * /
     1        50X, 5(4H****),1H*)
      WRITE(6, 60 )
  310 DO 320 N = 1,5
      ALPHA = 5 * (N-1)
      SUML(N) = SUML(N) + CLH(N)
      SUMM(N) = SUMM(N) + CMH(N) + CMF(N)
      SUMLL(N) = SUMLL(N) + CLLH(N) + CLLV(N)
      SUMD(N) = SUMD(N) + CDH(N) + CDV(N)
      SUMN(N) = SUMN(N) + CNV(N) + CNF(N)
      SUMY(N) = SUMY(N) + CYV(N)
      WRITE(6, 80 )ALPHA,SUML(N),SUMM(N),SUMLL(N),SUMD(N),SUMN(N),SUMY(N
     1)
  320 CONTINUE
C  SUM UP ALL THE COMPONENT DYNAMIC COEFFICIENTS
      IF(NOSTAB) GO TO 330
      WRITE(6, 120 )
      CLQ = CLQW + CLQH
      CMQ = CMQW + CMQH + CMQF
      CLLP = CLLPW + CLLPH + CLLPV
      CLLR = CLLRW + CLLRV
      CNP = CNPW + CNPV
      CNR = CNRW + CNRV + CNRF
      CYP = CYPW + CYPV
      CYR = CYRW + CYRV
      WRITE(6, 130 ) CLQ,CMQ,CLLP,CLLR,CNP,CNR,CYP,CYR
  330 CONTINUE
  340 IF(IPASS .GT. 1) GO TO 350
      IREAD = 2*NMAX + NSEG + NCC + 1
      IF(LOGIC .EQ. 2) GO TO 20
  350 CONTINUE
      END
```

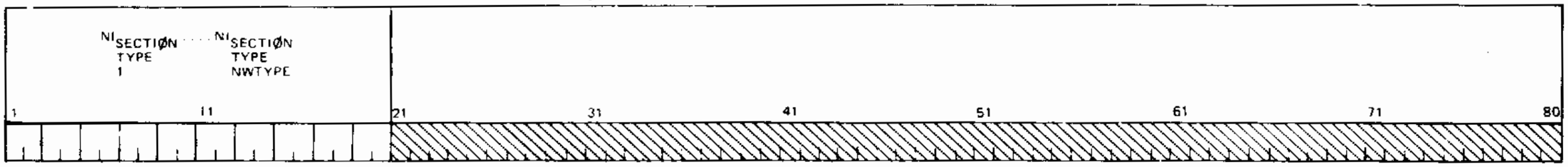
APPENDIX VI - LOAD SHEETS



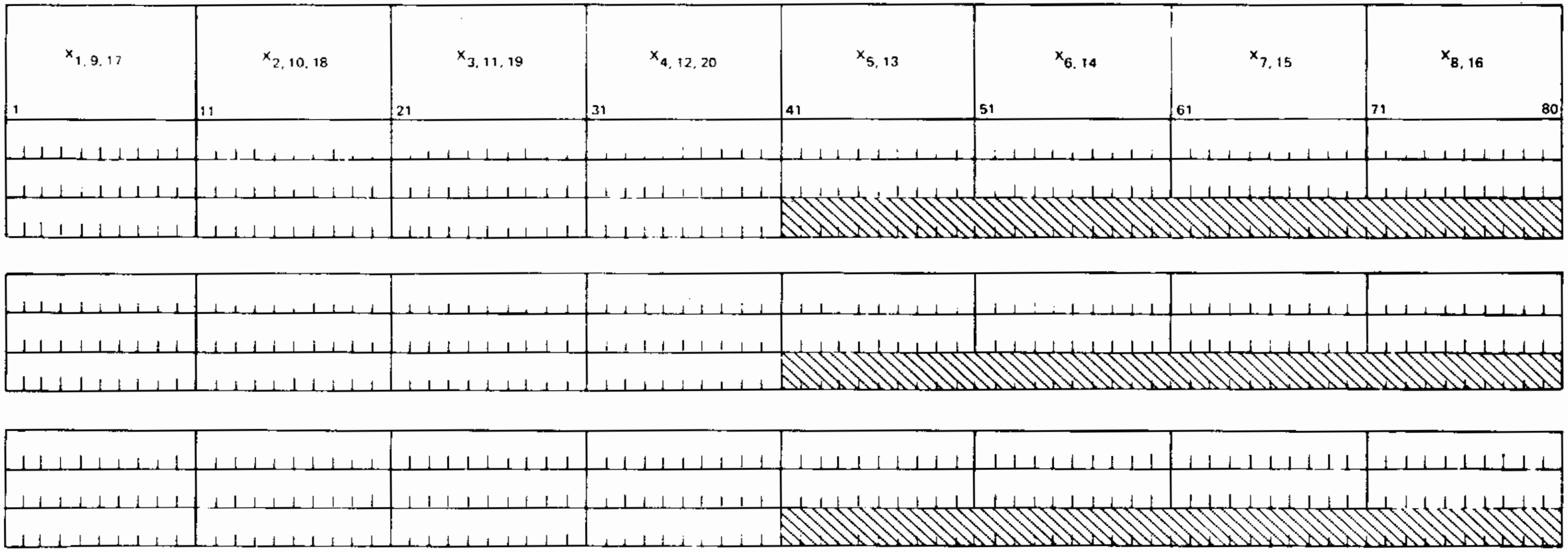
WING SECTION TYPE CARD



NUMBER OF CHORDWISE WING ELEMENTS



WING CHORDWISE ELEMENT COORDINATES



AS MANY AS 10 SETS PERMITTED (1 ≤ NWTYPE ≤ 10)

LEADING AND TRAILING EDGE COORDINATES

Y 1	X_{LEAD} 11	X_{TRAIL} 21	ØNLY REQUIRED FØR SECTIØNS ADJACENT TØ LEADING ØR TRAILING EDGE BREAKS (THIS INPUT REQUIRED ØNLY IF IGTYPE = 1) 31 41 51 61 71 80
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
			▨
9 ▨	▨	▨	▨

AS MANY AS NRØWS CARDS PERMITTED (NRØWS ≤ 40)

TRAPEZOIDAL WING PARAMETERS

ASPECT RATIØ 1	SWEEP 11	TAPER RATIØ 21	(ØNLY REQUIRED IF IGTYPE = 2) 31 41 51 61 71 80
			▨

GROUND HEIGHT CARD

				GHITE1, GHITE2, ...			UP TO 8 GROUND HEIGHTS ALLOWED
1	11	21	31	41	51	61	71

JET SECTION TYPE CARD

				IJTYPE$_{\text{SECTION } 1}$ . . . . IJTYPE$_{\text{SECTION NROWS}}$			HIGHEST VALUE INPUT = NJTYPE NUMBER OF NONZERO VALUES = NROWSJ
1	11	21	31	41	51	61	71 80

NUMBER OF CHORDWISE JET ELEMENTS

NI$_{\text{SECTION}}$ TYPE 1 . . .	NI$_{\text{SECTION}}$ TYPE NJTYPE						
1	11	21	31	41	51	61	71 80

JET CHORDWISE ELEMENT COORDINATES

$\bar{x}_{1,9,17}$	$\bar{x}_{2,10,18}$	$\bar{x}_{3,11,19}$	$\bar{x}_{4,12,20}$	$\bar{x}_{5,13}$	$\bar{x}_{6,14}$	$\bar{x}_{7,15}$	$\bar{x}_{8,16}$
1	11	21	31	41	51	61	71 80

AS MANY AS 10 SETS PERMITTED (NJTYPE 10)

FUNDAMENTAL CASE CONTROL CARD

INTWST	INHING	INDELJ	INCAMB	INBETA	▨
1					11 21 31 41 51 61 71 80
					▨

TWIST DISTRIBUTION CARDS

$\text{TWIST}_{1,9,}$	$\text{TWIST}_{2,10,\ldots}$	$\text{TWIST}_{3,11,\ldots}$	$\text{TWIST}_{4,12,\ldots}$	$\text{TWIST}_{5,13,\ldots}$	$\text{TWIST}_{6,14,\ldots}$	$\text{TWIST}_{7,15,\ldots}$	$\text{TWIST}_{8,16,\ldots}$
1	11	21	31	41	51	61	71 80

LEADING EDGE HEIGHT CARDS

$\text{HL}_{1,9,\ldots}$	$\text{HL}_{2,10,\ldots}$	$\text{HL}_{3,11,\ldots}$	$\text{HL}_{4,12,\ldots}$	$\text{HL}_{5,13,\ldots}$	$\text{HL}_{6,14,\ldots}$	$\text{HL}_{7,15,\ldots}$	$\text{HL}_{8,16,\ldots}$
1	11	21	31	41	51	61	71 80

JET DEFLECTION CARDS

$\text{DJ}_{1,9,\ldots}$	$\text{DJ}_{2,10,\ldots}$	$\text{DJ}_{3,11,\ldots}$	$\text{DJ}_{4,12,\ldots}$	$\text{DJ}_{5,13,\ldots}$	$\text{DJ}_{6,14,\ldots}$	$\text{DJ}_{7,15,\ldots}$	$\text{DJ}_{8,16,\ldots}$
	11	21	31	41	51	61	71 80

CAMBER TYPE FLAG CARD

ICT_1	ICT_2	ICT_3	ICT_4	. .	. .	. .		. .	$ICT_{NRØWS}$	
1					11		21 31 41 51		61	71 80

CAMBER ANGLE CARDS

$AC_{1,9,17}$	$AC_{2,10,18}$	$AC_{3,11,19}$	$AC_{4,12,20}$	$AC_{5,13}$	$AC_{6,14}$	$AC_{7,15}$	$AC_{8,16}$
1	11	21	31	41	51	61	71 80
				▨	▨	▨	▨
				▨	▨	▨	▨
				▨	▨	▨	▨

(AS MANY AS 10 SETS PERMITTED (1 ≤ NCT ≤ 10))

TRAILING EDGE CAMBER ANGLE CARDS

$ACTE_{1,9,...}$	$ACTE_{2,10,...}$	$ACTE_{3,11,...}$	$ACTE_{4,12,...}$	$ACTE_{5,13,...}$	$ACTE_{6,14,...}$	$ACTE_{7,15,...}$	$ACTE_{8,16,...}$
1	11	21	31	41	51	61	71 80

(NRØWS) VALUES REQUIRED)

HINGE SECTION TYPE CARD

$IHT_{R\O W_1}$	....	$IHT_{R\O W_{NR\O WS}}$	(HIGHEST VALUE INPUT - NHT)

1 11 21 31 41 51 61 71 80

HINGE PARAMETER CARDS

XBH_1	ILT_1	$BETA_1$	XBH_2	ILT_2	$BETA_2$	XBH_3	ILT_3	$BETA_3$	XBH_4	ILT_4	$BETA_4$
1 10	11	12 20	21 30	31	32 40	41 50	51	52 60	61 70	71	72 80

COMPOSITE CASE CARDS

N_1 1 2	FACTØR N_1 3 8	N_2 9 10	FACTØR N_2 11 16	N_3 17 18	FACTØR N_3 19 24	N_4 25 26	FACTØR N_4 27 32	N_5 33 34	FACTØR N_5 35 40	N_6 41 42	FACTØR N_6 43 48	N_7 49 50	FACTØR N_7 51 56	N_8 57 58	FACTØR N_8 59 64	N_9 65 66	FACTØR N_9 67 72	N_{10} 73 74	FACTØR N_{10} 75 80
9 ▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨

HORIZONTAL TAIL PLANFORM CARD

1	11	21	31	41	51	61	71
ASPECT RATIØ	TAPER RATIØ	SWEEP	SPAN	LENGTH	HEIGHT		

HORIZONTAL TAIL SECTION CENTERLINE CARDS

1	11	21	31	41	51	61	71
$Y_{H_{1,9,17}}$	$Y_{H_{2,10,18}}$	$Y_{H_{3,11,19}}$	$Y_{H_{4,12,20}}$	$Y_{H_{5,13}}$	$Y_{H_{6,14}}$	$Y_{H_{7,15}}$	$Y_{H_{8,16}}$

HORIZONTAL TAIL SECTION TYPE CARD

1	11	21	31	41	51	61	71
			IHX_1 .... $IHX_{NRØWH}$			HIGHEST VALUE = NXTYPE	

HORIZONTAL TAIL CAMBER TYPE CARD

1	11	21	31	41	51	61	71
			IHC_1 .... $IHC_{NRØWH}$			ØMIT IF ICAMH - 0 HIGHEST VALUE - NCTYPE	

VERTICAL TAIL PLANFORM CARD

ASPECT RATIØ	TAPER RATIØ	SWEEP	SPAN	LENGTH	HEIGHT	
1	11	21	31	41	51	61 71

VERTICAL TAIL SECTION CENTERLINE CARDS

$Z_{V_{1,9}}$	$Z_{V_{2,10}}$	Z_{V_3}	Z_{V_4}	Z_{V_5}	Z_{V_6}	Z_{V_7}	Z_{V_8}
1	11	21	31	41	51	61	71

VERTICAL TAIL SECTION TYPE CARD

IVX_1 .... $IVX_{NRØWV}$

HIGHEST VALUE = NXTYPE

1	11	21	31	41	51	61	71

VERTICAL TAIL CAMBER TYPE CARD

IVC_1 .... $IVC_{NRØWV}$

ØMIT IF ICAMV = 0
HIGHEST VALUE = NCTYPE

1	11	21	31	41	51	61	71

NUMBER OF CHORDWISE TAIL ELEMENTS

$NX_{SECTION}$ TYPE 1 · · · ·	$NX_{SECTION}$ TYPE NXTYPE						
1	11	21	31	41	51	61	71
		▨	▨	▨	▨	▨	▨

NUMBER OF TAIL CHORDWISE CAMBER ANGLES

$NC_{SECTION}$ TYPE 1 · · ·	$NC_{SECTION}$ TYPE NCTYPE						
1	11	21	31	41	51	61	71
		▨	▨	▨	▨	▨	▨

TAIL CHORDWISE ELEMENT COORDINATES

$XBT_{1,9}$	$XBT_{2,10}$	XBT_3	XBT_4	XBT_5	XBT_6	XBT_7	XBT_8
1	11	21	31	41	51	61	71
		▨	▨	▨	▨	▨	▨

		▨	▨	▨	▨	▨	▨

AS MANY AS 10 SETS PERMITTED (1 ≤ NXTYPE ≤ 10)

TAIL CHORDWISE CAMBER ANGLES

$CAMBER_{1,9}$	$CAMBER_{2,10}$	$CAMBER_3$	$CAMBER_4$	$CAMBER_5$	$CAMBER_6$	$CAMBER_7$	$CAMBER_8$
1	11	21	31	41	51	61	71
		▨	▨	▨	▨	▨	▨

		▨	▨	▨	▨	▨	▨

AS MANY AS 10 SETS PERMITTED (0 ≤ NCTYPE ≤ 10)

FUSELAGE SECTION BOUNDARY CARDS

$XF_{1,9,17..}$	$XF_{2,10,18}$	$XF_{3,11,19..}$	$XF_{4,12,20}$	$XF_{5,13,21}$	$XF_{6,14,22..}$	$XF_{7,15,23}$	$XF_{8,16,24..}$
1	11	21	31	41	51	61	71
	▨	▨	▨	▨	▨	▨	▨

INPUT (NSEG + 1) VALUES

FUSELAGE SECTION VERTICAL DISPLACEMENT

$ZCL_{1,9,17..}$	$ZCL_{2,10,18..}$	$ZCL_{3,11,19..}$	$ZCL_{4,12,20..}$	$ZCL_{5,13,21..}$	$ZCL_{6,14,22..}$	$ZCL_{7,15,23..}$	$ZCL_{8,16,24..}$
1	11	21	31	41	51	61	71
	▨	▨	▨	▨	▨	▨	▨

INPUT (NSEG + 1) VALUES
ØMIT IF ICAMF = 0

NUMBER OF FUSELAGE SECTION ELEMENTS

NF_1 . . . . . NF_{NSEG}

$3 \leq NF \leq 20$

1	11	21	31	41	51	61	71

FUSELAGE SECTION INPUT FOR ARBITRARY CROSS-SECTIONS

$Y_{1, 9, 17}$ 1	$Y_{2, 10, 18}$ 11	$Y_{3, 11, 19}$ 21	$Y_{4, 12, 20}$ 31	$Y_{5, 13, 21}$ 41	$Y_{6, 14}$ 51	$Y_{7, 15}$ 61	$Y_{8, 16}$ 71
					▨	▨	▨

Z

					▨	▨	▨

MINIMUM ØF (NSEG + 1) SETS
MAXIMUM ØF 2 (NSEG) SETS
ØMIT IF ISECT ≠ 1

FUSELAGE SECTION INPUT FOR CIRCULAR CROSS-SECTIONS

$R_{1, 9, 17 \ldots}$ 1	$R_{2, 10, 18 \ldots}$ 11	$R_{3, 11, 19 \ldots}$ 21	$R_{4, 12, 20 \ldots}$ 31	$R_{5, 13, 21 \ldots}$ 41	$R_{6, 14, 22 \ldots}$ 51	$R_{7, 15, 23 \ldots}$ 61	$R_{8, 16, 24 \ldots}$ 71
	▨	▨	▨	▨	▨	▨	▨

INPUT (NSEG + 1) VALUES
ØMIT IF ISECT ≠ 2

FUSELAGE SECTION INPUT FOR ELLIPTICAL CROSS-SECTIONS

AMIN$_{1, 9, 17}$ . . . 1	AMIN$_{2, 10, 18}$ . . . 11	AMIN$_{3, 11, 19}$ . . . 21	AMIN$_{4, 12, 20}$ . . . 31	AMIN$_{5, 13, 21}$ . . . 41	AMIN$_{6, 14, 22}$ . . . 51	AMIN$_{7, 15, 23}$ . . . 61	AMIN$_{8, 16, 24}$ . . . 71
	▨	▨	▨	▨	▨	▨	▨

AMAX

	▨	▨	▨	▨	▨	▨	▨

INPUT (NSEG + 1) VALUES
ØMIT IF ISECT ≠ 3

OFF-BODY FLOW FIELD INPUT

X 1	Y 11	Z 21	31	41	51	61	71

INPUT NPØINT CARDS
MAXIMUM ØF 200 CARDS

JET STRENGTH CARDS

$CMU_{\text{JET SECTION } 1} \ldots CMU_{\text{JET SECTION NROWSJ}}$

1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80

(AS MANY SETS AS DESIRED MAY BE INCLUDED HERE)

1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80
9							

VECTORED JET PARAMETERS

CJ 1	D0 11	THJ 21	DELZ 31	X0 41	Y0 51	Z0 61	DIHED 71–77	ISY 78	JETREP 79	LAST 80

(AS MANY SETS AS DESIRED MAY BE INCLUDED HERE)

1	
9	

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)* McDonnell Douglas Corporation Douglas Aircraft Company Long Beach, California 90846	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
METHODS FOR PREDICTING THE AERODYNAMIC AND STABILITY AND CONTROL CHARACTERISTICS OF STOL AIRCRAFT. VOL. II STOL AERODYNAMIC METHODS COMPUTER PROGRAM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Technical Report

5. AUTHOR(S) *(First name, middle initial, last name)*
Mark I. Goldhammer
Norman F. Wasson

6. REPORT DATE December 1973	7a. TOTAL NO. OF PAGES 210	7b. NO. OF REFS 4

8a. CONTRACT OR GRANT NO. F33615-71-C-1861	9a. ORIGINATOR'S REPORT NUMBER(S) MDC J5965-02
b. PROJECT NO. 643A	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	AFFDL-TR-73-146, Volume II

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Flight Dynamics Laboratory Wright-Patterson Air Force Base Ohio 45433

13. ABSTRACT

This report describes the STOL Aerodynamic Methods Computer Program, which is intended to aid the engineer in the design and analysis of STOL aircraft employing internally ducted jet flaps, externally blown jet flaps, and mechanical flap systems with vectored thrust. The program provides capabilities to predict either the overall aerodynamic characteristics of a configuration or the aerodynamics of the following:

1. Jet-wing (in or out of ground effect)
2. Wing and vectored jets
3. Fuselage in the flow field of the jet-wing and/or vectored jets
4. Empennage in the flow field of the jet-wing and/or vectored jets

The program includes the capabilities for investigating the effects of arbitrary wing planforms with arbitrary high lift systems, including partial span flaps, slats, and jets; arbitrary camber, twist, and jet deflection; fuselages with arbitrary cross-sections and upsweep; arbitrary empennage arrangements, including conventional tails, mid-tails, and T-tails; and a capability to calculate the off-body flow field induced by the jet-wing and/or vectored at arbitrary points.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
AERODYNAMICS AERODYNAMIC STABILITY AND CONTROL COMPUTER PROGRAM FINITE ELEMENT METHODS HIGH LIFT SYSTEMS JET FLAP LIFTING SURFACE THEORY NUMERICAL ANALYSIS SHORT TAKEOFF AND LANDING AIRCRAFT SLENDER BODY THEORY VORTICITY WING THEORY						

Unclassified
Security Classification

☆U.S.Government Printing Office: 1974 — 657-015/234