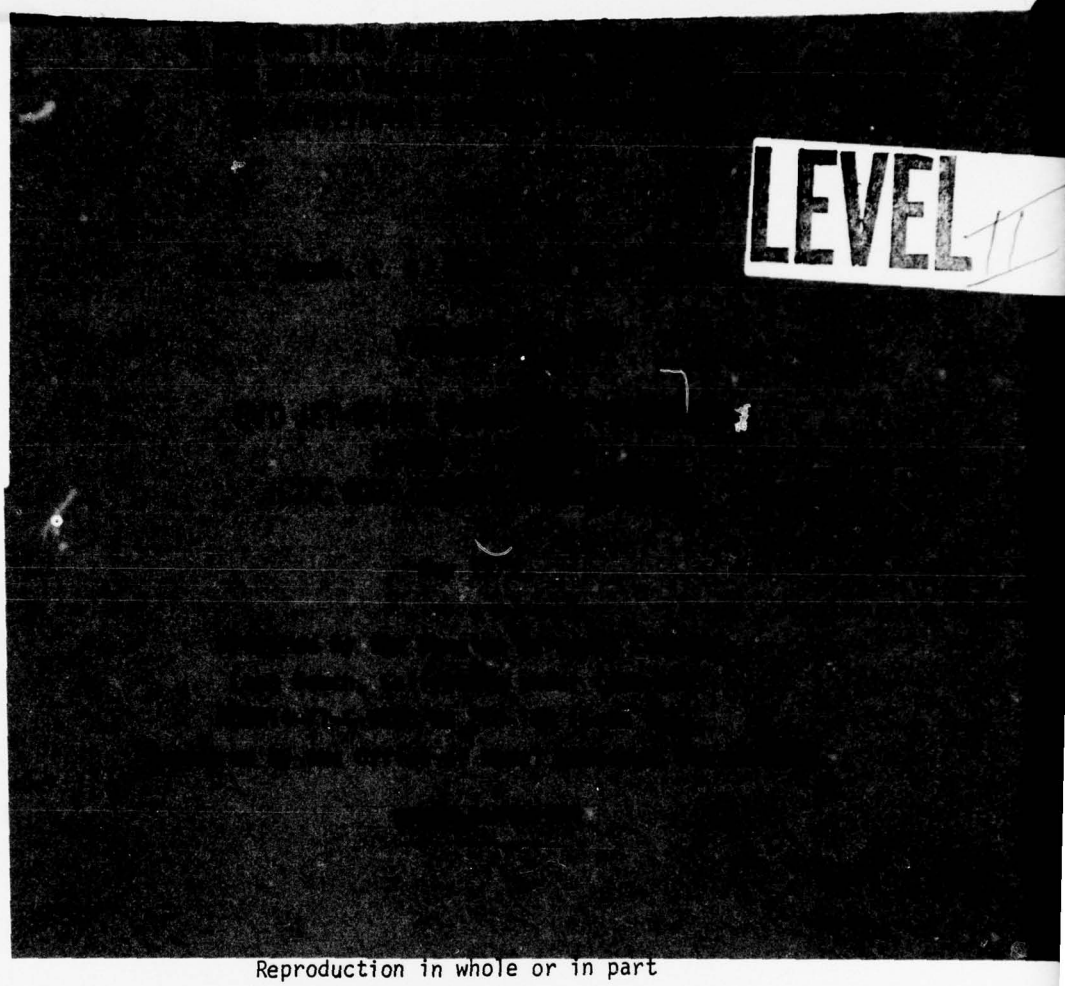
AD-A058 821 DOUGLAS AIRCRAFT CO LONG BEACH CALIF F/G 20/4
A THEORETICAL METHOD FOR CALCULATING THE AERODYNAMIC CHARACTERI--ETC(U)
1977 M L LOPEZ, C SHEN, N F WASSON N00014-71-C-0250
UNCLASSIFIED MDC-J5519-02-VOL-2 NL

1 OF 2
AD A058 821

THE ENCLOSED REPORT SUPERSEDES THE EARLIER EDITION WHICH,
ACCORDING TO OUR RECORDS, IS IN YOUR POSSESSION.

DDC FILE COPY
AD A058821



Reproduction in whole or in part
is permitted for any purpose of the
United States Government

DDC
SEP 19 1978
F

78 09 05 161

Approved for public release;
distribution unlimited

DOUGLAS AIRCRAFT COMPANY
MCDONNELL DOUGLAS CORPORATION

NOMENCLATURE

(12) 128p.

(4)

ISSUED May 73
SECOND EDITION 1977

(14) REPORT NO. MDC-J5519-02 VOL-

(11) 1977

(6) A THEORETICAL METHOD FOR CALCULATING
THE AERODYNAMIC CHARACTERISTICS
OF ARBITRARY JET FLAPPED WINGS Volume II. EVD
Jet-Wing Computer Program User's Manual
(CDC 6000 Series Computers, Second Edition)

(10) Michael Cheng-Chung Norman
N. L. Lopez, C. C. Shen and N. F. Wasson

VOLUME II N/A

DDC
RECEIVED
SEP 19 1978
F

EVD JET-WING COMPUTER PROGRAM
USER'S MANUAL
(CDC 6000 SERIES COMPUTERS)

Prepared by the Douglas Aircraft Company,
Long Beach, California under Contract
(15) N00014-71-C-0250/NR 215-189 (Code 461),
Sponsored by the Office of Naval Research, Aeronautics

(9) Final technical rept.,

Reproduction in whole or in part
is permitted for any purpose of the
United States Government

Approved for public release;
distribution unlimited

116 400

This report describes the EVD Jet-Wing Computer Program, developed by the V/STOL Technology Development Group - Aerodynamics, of the Douglas Aircraft Company, McDonnell Douglas Corporation. Development of the program was begun in 1970 under sponsorship of the McDonnell Douglas Independent Research and Development Program (IRAD), and resulted in the Mark I version. From April, 1971 through April, 1972, the program was entirely rewritten, drawing heavily on the basic features developed in the Mark I version, but incorporating extensive improvements in program flexibility, efficiency and the amount of information available to the user. This work, culminating in the Mark II version of the program, was sponsored by Office of Naval Research Contract N00014-71-C-0250.

The major portion of the program was written by Mr. N. F. Wasson under the direction of Dr. C. C. Shen and Mr. M. L. Lopez, Technical Director. Other programming contributions were made by Mr. N. D. Halsey on the momentum "induced drag" method, and by Messrs. J. L. Hess, T. M. Ridell, and D. N. Smyth on the matrix solution. The technical work of Messrs. M. I. Goldhammer, B. K. Lakin, and W. V. Whitman in applying the program to various configurations has also been of great value in its design, verification and correction.

This report has been reviewed and is approved.

M. L. Lopez, Section Manager
V/STOL Systems
Aerodynamic Technology

E. S. Rutowski, Chief
Aerodynamics Engineer
Technology Development

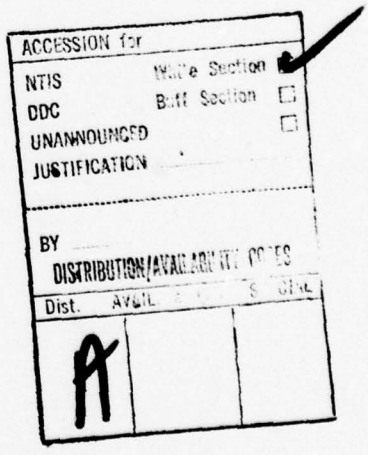
O. R. Dunn, Director
of Aerodynamics

ABSTRACT

This report describes the EVD Jet-Wing Computer Program, which is based upon the Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Theory described in Volume I of this report. This program provides a capability for determining the aerodynamic characteristics of wings of arbitrary planform, and includes the following:

1. Spanwise and chordwise loading
2. Spanwise variation of induced drag
3. A capability to investigate the effects of:
   a. Part span flaps
   b. Part span blowing
   c. Pitching, rolling, yawing and sideslip
4. Total lift and induced drag (momentum method), pitching, yawing and rolling moments, etc.

The program has the capabilities for investigating the effects of a variation of leading and trailing flap deflection, camber, twist, jet deflection and jet momentum.



PREFACE TO REVISED EDITION

This report was originally published in May 1973 in two volumes: Volume I, The Elementary Vortex Distribution Jet-Wing Lifting Surface Theory and Volume II, EVD Jet-Wing Computer Program User's Manual. In this new edition several typographical errors have been corrected, a minor change made in the computer program, and the "Limited Distribution" of Volume II removed.

Michael L. Lopez
Douglas Aircraft Company
McDonnell Douglas Corporation

CONTENTS

		Page
Foreword		ii
Abstract		iii
Index of Figures		v
Nomenclature		vi
1.0	Introduction	1
2.0	Fundamentals of Program Use	2
	2.1 Philosophy of Program Use	2
	2.2 Brief Review of Linearized Approach	3
	2.3 Geometry Preparation	4
	2.4 Use of Fundamental Cases	7
	2.5 Use of Composite Cases	12
	2.6 Final Comments	13
3.0	Program Description	14
	3.1 General Description	14
	3.2 Subroutine Descriptions	16
	3.3 EVD Function-Numerical Restrictions	32
4.0	Input Data Instructions	35
	4.1 Input Description	35
	4.2 Input Restrictions	49
5.0	Output Data Description.	51
	5.1 Wing Characteristics and Control Flags	53
	5.2 Fundamental Case Geometry	53
	5.3 Sectional Jet Strength	54
	5.4 Fundamental Case Coefficients	54
	5.5 Composite Case Coefficients	55
	5.6 Stability Derivative Output	58
6.0	Operational Considerations	60
	6.1 Program Versions Available	60
	6.2 Overlay Structure	60
	6.3 Peripheral Storage Devices	60
	6.4 Program Running Time	62
References		64
Appendix A	Sample Problem	A1
Appendix B	Error Messages	B1
Appendix C	Program Listing	C1
Appendix D	Load Sheets	D1

THIS PAGE INTENTIONALLY LEFT BLANK

INDEX OF FIGURES

FIGURE	TITLE	PAGE
1	Program Representation of a Jet-Wing	7
2	Representation of Wing Sections by Equivalent Camber	9
3	Treatment of Fundamental and Composite Cases	11
4	General Program Functional Arrangement	15
5	Qualitative Program Flow Charts	17
6	Arrangement of Input Card Deck	36
7	Program Planform Dimensions and Notation	37
8	Reference Coordinate System	39
9	Program Overlay Structure	61
10	Computer Program Operational Time Requirements	63

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
A	A	I/Ø	Fundamental case scale factor
AC	ε_c	I/Ø	Local incidence due to camber
ACTE	ε_{ct}	I/Ø	Local incidence due to camber at trailing edge
ALFIN	α_{i_∞}	Ø	Downwash angle or jet angle at infinity downstream
ALPHA	α	Ø	Angle of attack
ARATIØ	AR	I/Ø	Aspect ratio
AREA	S	I/Ø	Wing planform reference area
BETA	β	I/Ø	Deflection angle, (δ_f, δ_J, or δ_s)
CBGL	$C_{B_{\Gamma_L}}$	Ø	Root bending moment coefficient due to pressure on left wing
CBGR	$C_{B_{\Gamma_R}}$	Ø	Root bending moment coefficient due to pressure on right wing
CBJL	$C_{B_{J_L}}$	Ø	Root bending moment coefficient due to jet reaction on left wing
CBJR	$C_{B_{J_R}}$	Ø	Root bending moment coefficient due to jet reaction on right wing
CBL	C_{B_L}	Ø	Total root bending moment coefficient on left wing
CBR	C_{B_R}	Ø	Total root bending moment coefficient on right wing
CCD	$(C_{D_i})_p$	Ø	Total induced drag coefficient calculated by pressure integration
CCJ	C_J	Ø	Total jet momentum coefficient
CCL	C_L	Ø	Total lift coefficient
CCM	C_m	Ø	Total pitching moment coefficient (about wing apex)
CCS	C_S	Ø	Total leading edge suction coefficient

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
CCT	C_T	Ø	Total net thrust coefficient
CCY	C_Y	Ø	Total side force coefficient
CD	c_{d_i}	Ø	Sectional induced drag coefficient
CDITZ	$(C_{D_i})_M$	Ø	Total induced drag coefficient, calculated by the momentum method
CHØRD	c	Ø	Wing sectional chord
CL	c_ℓ	Ø	Sectional lift coefficient
CLL	C_l	Ø	Rolling moment coefficient
CLLP	C_{l_p}	Ø	Rolling moment coefficient derivative due to rolling
CLLR	C_{l_r}	Ø	Rolling moment coefficient derivative due to yawing
CLQ	C_{L_q}	Ø	Lift coefficient derivative due to pitching
CM	c_m	Ø	Sectional pitching moment coefficient (about the local leading edge)
CMAC	$\bar{\bar{c}}$	Ø	Mean aerodynamic chord (MAC)
CMQ	C_{m_q}	Ø	Pitching moment coefficient derivative due to pitching about center of gravity
CMU	c_μ	I/Ø	Sectional jet momentum coefficient
CN	C_n	Ø	Yawing moment coefficient
CNI	C_{n_i}	Ø	Total yawing moment coefficient due to induced drag (including leading edge suction)
CN(P)	C_{n_p}	Ø	Yawing moment coefficient derivative due to rolling, dependent on rolling rate
CN(R)	C_{n_r}	Ø	Yawing moment coefficient derivative due to yawing, dependent on yawing rate
CP	Δc_p	Ø	Pressure coefficient, $\Delta c_p = c_{p_{\text{lower surface}}} - c_{p_{\text{upper surface}}}$

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
CPMBL	$\frac{Y_{C.L.}}{b/2}$	Ø	Spanwise center of lift of left wing, in terms of half span
CPMBR	$\frac{Y_{C.L.}}{b/2}$	Ø	Spanwise center of lift of right wing, in terms of half span
CREF	$\tilde{c}$	I/Ø	Wing reference chord
CS	c_s	Ø	Sectional leading edge suction coefficient
CTO	c_t	Ø	1. Sectional thrust coefficient 2. Tip chord of wing
CXCL	$\frac{X_{C.L.}}{\tilde{c}}$	Ø	x-coordinate of center of lift, in terms of reference chord CXCL = - (CCMG + CCML)/CCL labeled (X/CREF)
CXCLB	$\frac{X_{C.L.}}{b/2}$	Ø	x-coordinate of center of lift, in terms of half span CXCLB = - ((CCMG + CCML)/CCL)* CREF labeled (X/B/2)
CXCP	$\frac{X_{C.P.}}{\tilde{c}}$	Ø	x-coordinate of center of pressure, in terms of reference chord
CXCPB	$\frac{X_{C.P.}}{b/2}$	Ø	x-coordinate of center of pressure, in terms of half span
CY(P)	C_{Y_p}	Ø	Side force coefficient derivative due to rolling, dependent on rolling rate
CY(R)	C_{Y_r}	Ø	Side force coefficient derivative due to yawing, dependent on yawing rate
D	d	Ø	Chordwise distance from trailing edge to infinity EVD vortex point
DEL	$\overline{\delta}$	Ø	Chordwise length of an EVD element in terms of local chord
DELTA	$\overline{\Delta}$	Ø	Half of the spanwise width of an EVD element or a spanwise division, normalized by b/2
DJ	δ_J	I/Ø	Jet deflection angle relative to the wing trailing edge slope

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
EPS	ε_i	Ø	Total local incidence angle at a downwash control point on the wing
GAMMA	γ	Ø	Vorticity intensity
HL	h_ℓ	I/Ø	Height of leading edge above the x-y plane
I	i	Ø	EVD element sequence number
ICT		I	Camber type flag for each wing section
ICTYPE		I	Wing chordwise division type flag for each section
IDERIV		I	Stability derivative flag
IGTYPE		I/Ø	Planform geometry type flag
IHINGE		I/Ø	Hinge EVD flag
IHT		I	Hinge type flag for each wing section
IJTYPE		I	Jet chordwise division type flag for each section
ILT		I	Flap type flag
INBETA		I	Hinge angle input flag
INCAMB		I	Camber angle input flag
INDELJ		I	Jet angle input flag
INHITE		I	Leading edge height input flag
INTWST		I	Twist angle input flag
IPRINT		I/Ø	Output control flag
ISYMM		I/Ø	Symmetry control flag
J	J		Sectional jet momentum
JETFLG		I/Ø	Jet control flag
K	k		Sequence number of spanwise divisions or sections

NOMENCLATURE

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
N		I/Ø	Fundamental case sequence number
NCASES	K	I/Ø	Number of fundamental cases
NCT		I	Number of camber types
NHT		I	Number of hinge types
NI		I	Number of chordwise divisions of a particular chordwise division type (wing or jet)
NJ		Ø	Number of chordwise divisions of jet section
NJTYPE		I	Number of jet chordwise division types
NRØWS		Ø	Number of wing sections or spanwise divisions
NRØWSJ		I	Number of sections having a trailing jet
NW		Ø	Number of chordwise division of each section
NWTYPE		I	Number of wing chordwise division types
SPAN	b	I/Ø	Wing span
SWEEP	$\Lambda_{c/4}$	I/Ø	Wing quarter-chord sweep angle
TANLE	$\tan\Lambda_{\ell}$	Ø	Tangent of sectional leading edge sweep angle
THETA	θ	Ø	Total jet deflection angle relative to the free stream
TITLE		I/Ø	Run title
TR	λ	I	Wing taper ratio
TWIST	α_o	I/Ø	Twist angle for each wing section
TYPE		Ø	EVD type flag
U	U		Free stream velocity
X	x	Ø	Chordwise coordinate of the Cartesian coordinate system

NAME	SYMBOL	INPUT/ OUTPUT	DEFINITION
XB	$\bar{x}$	I/Ø	Chordwise distance of an EVD element from the leading edge, in terms of local chord
XBH	$\bar{x}_h$	I/Ø	Chordwise distance of a hinge point from the leading edge, in terms of local chord
XCL/C	$\bar{x}_{c.l.}$	Ø	Chordwise location of center of lift at a section measured from local leading edge, in terms of local chord
XCP/C	$\bar{x}_{c.p.}$	Ø	Chordwise location of center of pressure at a section measured from local leading edge, in terms of local chord
XCG	$x_{c.g.}$	I/Ø	x-coordinate of the center of gravity about which pitching rate derivatives are taken
XI	x_i	Ø	x-coordinate of an EVD element
XLEAD	x_ℓ	I/Ø	x-coordinate of the leading edge
XMC	$X_{m.c.}$	I/Ø	x-coordinate of the moment center about which pitching moments are taken
XTRAIL	x_t	I/Ø	x-coordinate of the trailing edge
Y	y	I/Ø	y-coordinate of a section

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
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	u	Indicates a contribution due to jet reaction at a wing section

SUFFIX	SUBSCRIPT	DEFINITION
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
0	o	Indicates $\alpha = 0$

1.0 INTRODUCTION

This report describes the Mark II version of the EVD Jet-Wing Computer Program, intended for use as a preliminary design tool for the analysis of arbitrary wings incorporating trailing edge blowing. The program is based upon the linearized Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Theory described in Volume I of this report (Reference 1). The program is written in the Fortran IV language and can be adapted for use on many existing large computing sytems. Versions are currently in operation on the CDC 6000 series and IBM 360 and 370 series computers.

In order to execute the program, the user must describe the arbitrary geometric configuration to the computer by using several different types of input cards containing specific information in a fixed order. Deflection, for example, of a wing control surface may be accomplished by using the linear "fundamental case" feature of the program. Superposition of the aerodynamic characteristics associated with fundamental cases so as to provide the overall aerodynamic characteristics of a complete configuration may be obtained by specifying the "composite case" feature of the program. The number of jet strength cases which can be produced for each geometric configuration is limited only by the computing time available. Printed output from the program contains the jet-wing aerodynamic characteristics in coefficient form. Detailed spanwise and chordwise loading and certain dynamic stability derivatives may also be included at the option of the user.

Maximum realization of the potential of this program can only be achieved by a user who is thoroughly familiar with the applications and limitations of the linearized theory upon which it is based, and who is willing to apply his own engineering knowledge and judgment in a methodical manner to each configuration analyzed. It is suggested that both volumes of this report be read before running the first problem, and that a few simple problems be tried in order to develop understanding and familiarity with the program before large complicated problems are attempted.

2.0 FUNDAMENTALS OF PROGRAM USE

The EVD jet-wing computer program is based upon the linearized EVD jet-wing lifting surface theory described in Reference 1. This program has several unique features which enable it to be used by the aerodynamicist in the evaluation, design and development of V/STOL aircraft systems which utilize powered high lift systems based on the jet flap principle (e.g., EBF, augmentor wing, etc.). The scope of this program is not limited, however, to the analysis of jet flap wings. By selecting several options available to the user, this program may be used as a classical lifting surface method. Hence, as an aerodynamic tool, the analysis of conventional wings can be regarded as being within the scope of the present computer program.

The program is written entirely in the Fortran IV language, and can be adopted for use on any large-scale computer system. The program consists of several primary components designed to minimize computer storage requirements while maintaining maximum computer efficiency and speed. This modular arrangement also results in easier program checkout and correction.

The programming philosophy has sought to maintain the independence of each small component of the program wherever possible. This approach also facilitates the addition of new capabilities, with minimum alteration of the existing components.

2.1 Philosophy of Program Use

It is fundamentally important that the user of the EVD jet-wing computer program understand the basic limitations and approximations adopted in the solution of the jet-wing problem. The program is not a "magic box" with which he can expect to achieve valid results while supplying poorly prepared input data. He must, therefore, understand each step of his role in the analysis of a specific problem, including use of engineering knowledge and judgement in every decision he makes in the preparation of computer input data. Only then can he expect the program, which is essentially an engineering tool, to reliably and accurately complete the task assigned to it. This point may be summarize by the quaint term which applies to all computer programs: "GIGO" - Garbage In, Garbage Out!

Judgement is also required in the interpretation and application of the computer output data. The user should be aware of those factors not included in the program analysis, which may have an effect on the final result. He should also have some understanding of the way in which these factors would be likely to influence the program results. For example, if there are regions of flow separation on a real wing, the program would be expected to overpredict the lift of that wing. On the other hand, a thick wing is expected to have higher lift than the wing of zero thickness which is considered by the present program. The quantitative effects of such factors are, of course, very difficult to accurately predict, but the user should at least be aware of the trends which they are expected to produce.

The user is also cautioned against extrapolating the linearized results into regions where strong nonlinear behavior might be expected. For example, the small angle assumptions of the linearized approach make it unlikely that the program would accurately predict the characteristics of a wing with, say, a flap deflected at 60 degrees. While the program would give an indication of the trends produced by such a nonlinear configuration, and in certain cases linearized results have been shown to be remarkably good in seemingly nonlinear situations, it would nevertheless be dangerous to use the output data as though it were generally an accurate prediction of the aerodynamic characteristics under all conditions.

2.2 Brief Review of the Linearized Approach

The assumptions and restrictions of the EVD jet-wing lifting surface theory are discussed in detail in Volume I of this report. For the convenience of the reader, the basic assumptions are restated here:

On the Wing:

a. The wing is thin, and is represented by the mean camber line.

b. All local incidences are small, but may be discontinuous.

On the Jet:

a. The jet is thin.

b. The jet deflection, relative to the freestream, is small.

General Flowfield:

a. Incompressible, inviscid, irrotational.

b. No mixing occurs between the jet and the external flow.

c. All spanwise components of velocity are considered small relative to the freestream velocity.

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

The above assumptions imply the existence of certain other conditions which are of practical importance in the use of the program. Though they may not seem to be obviously derived from the basic assumptions, these conditions nevertheless are 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 trailing vorticity.

b. All potential field influences of the wing and jet originate in the wing and jet plane.

c. All boundary conditions are satisfied in the wing and jet plane.

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

e. Since the lifting surface theory is for a jet-wing system alone, no account is taken for the effects of fuselage, nacelles, empennage, etc.

f. There are no losses or dissipation associated with the jet sheet emergence from the wing trailing edge.

g. There are no gaps or slots in the wing (program restriction only).

2.3 Geometry Preparation

The EVD jet-wing computer program is capable of evaluating the aerodynamic characteristics of jet-wings of arbitrary wing planform and arbitrary jet spanwise distribution of momentum. In all cases, the jet sheet correctly extends to infinity downstream. The jet-wing system may be considered as symmetric, anti-symmetric, or non-symmetric. A symmetric jet-wing is defined as one for which, under all conditions, the computed aerodynamic loading would

be symmetric about the x-axis. This implies, therefore, symmetry about the x-axis of all the following characteristics:

Wing:

a. Planform geometry
b. Local incidence of each element

Jet:

c. Spanwise location
d. Strength at each spanwise section
e. Deflection

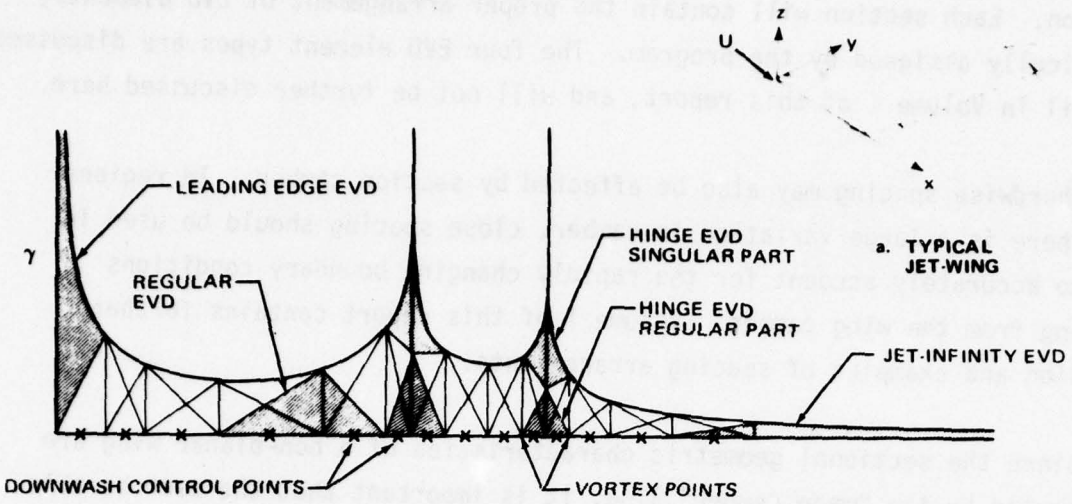
Anti-symmetric jet-wings are defined as having anti-symmetric computed aerodynamic loading. Thus, they must be symmetric in all of the above items except for (b) and (e), which must be anti-symmetric. Since anti-symmetry in the angle of attack of the wing is meaningless, the program will always treat the anti-symmetric jet-wing as being at zero angle of attack only. If an otherwise anti-symmetric wing were at an angle of attack, its loading would be non-symmetric, and it would, therefore, have to be treated as completely non-symmetric. Non-symmetric jet-wings are defined as having non-symmetry in any one or more of the characteristics listed in Items (a) through (e) above.

It is, however, possible for the user to superimpose by hand the results of anti-symmetric runs with those of symmetric or non-symmetric runs. Such a situation may arise, for example, when a quantity of data has been generated for various configurations of a particular wing and the effects of anti-symmetric ailerons are then required. It may be more efficient to compute the aileron effects alone in one additional run rather than rerunning all the previous configurations non-symmetrically with ailerons included. The superimposed loading will be in general non-symmetric and both the right and left halves of the wing must be considered. The linear aerodynamic coefficients (e.g., C_L, C_M, C_l) may be superimposed directly at zero angle of attack, and the variation with angle of attack will be the original symmetric or non-symmetric values. The non-linear coefficients (e.g., C_D, C_Y, C_n) cannot be superimposed directly. Instead, they must be obtained by an integration of the chordwise and spanwise loading.

The problem analysis begins with the user definition of the planform geometry of the wing and jet. The wing and jet are broken up into spanwise sections of arbitrary width. The arrangement of these sections and their size are mainly determined by two factors. First is the occurrence of planform discontinuities; for example, flap edges, flap extensions, leading and trailing edge brakes, jet limits, etc. Second is the allowance for spanwise variation of loading. Since the program loading is assumed to be that which occurs at the center of each section and is constant in the spanwise direction on each section, there should be close spanwise spacing in regions where rapid loading variation might be expected. For example, near the wing tip or near the edge of a highly loaded flap or strong jet, there will normally be a rapid change in spanwise loading. The spanwise sections should, therefore, be narrow in these regions so that the "stairstep" loading of the program can adequately represent a smooth loading variation.

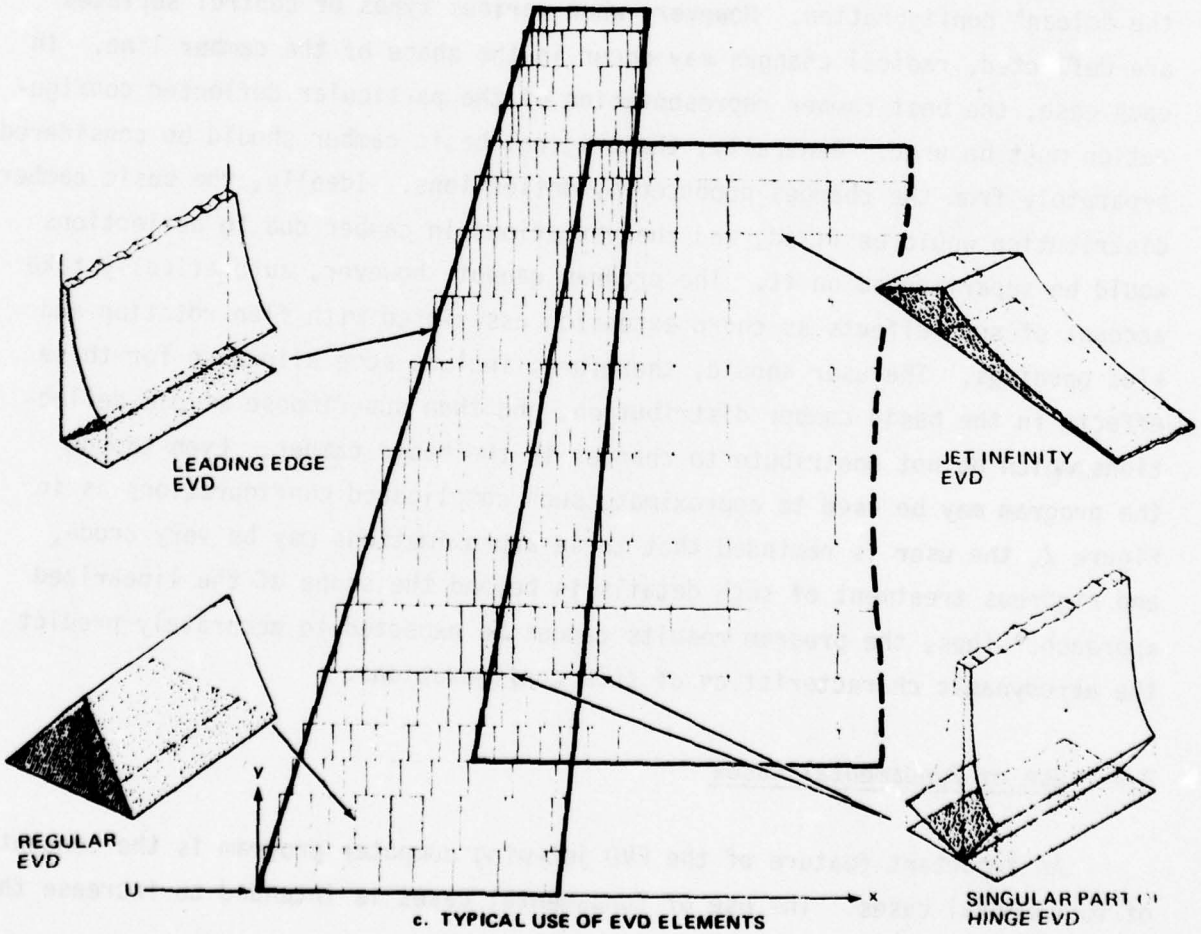
Each spanwise section is further broken up into chordwise divisions so that the jet-wing planform is represented by an array of rectangular elements with which are associated an equal number of Elementary Vortex Distribution elements. This representation is shown in Figure 1 for a typical jet-wing. Note that no two adjacent elements overlap, nor are there any gaps. Also, that any one of four types of EVD elements, (i.e., Regular, Leading Edge, Hinge, and Infinity), can be selected.

Given close uniform spacing, the program would always yield a good, smooth chordwise loading. But the linear nature of loading given by the Regular EVD elements allows much larger spacing to be used in regions where the loading is expected to nearly linear, that is, where the slope of the loading is nearly constant. In addition, in order to more accurately approximate the chordwise loading, special EVD elements have been developed. These EVD elements provide an accurate representation of chordwise loading in these special regions which are normally difficult to approximate with the Regular EVD. These regions are near the leading edge, near the abrupt turning angle produced at a flap or jet "hinge" point, and far downstream on the jet sheet. Use of the special EVD elements makes chordwise spacing less critical and reduces the required number of elements on each section. Most of the chordwise EVD elmenents are of the regular type, representing a linear loading



a. TYPICAL JET-WING

b. REPRESENTATION OF TYPICAL SECTIONAL VORTEX DISTRIBUTION



c. TYPICAL USE OF EVD ELEMENTS

FIGURE 1. PROGRAM REPRESENTATION OF A JET WING

variation. Each section will contain the proper arrangement of EVD elements, automatically assigned by the program. The four EVD element types are discussed in detail in Volume I of this report, and will not be further discussed here.

Chordwise spacing may also be affected by section camber. In regions where there is a large variation in camber, close spacing should be used in order to accurately account for the rapidly changing boundary conditions resulting from the wing camber. Volume I of this report contains further discussion and examples of spacing arrangements.

Since the sectional geometric characteristics of a non-planar wing are approximated by the "mean camber" line, it is important that the user be able to translate a typical wing section into a suitable camber distribution. See Figure 2 for the camber line representation of various complex wing sections. All of the wing sections originate with some smooth camber distribution of the "clean" configuration. However, when various types of control surfaces are deflected, radical changes may occur in the shape of the camber line. In each case, the best camber representation of the particular deflected configuration must be used. Generally, the original basic camber should be considered separately from the changes produced by deflections. Ideally, the basic camber distribution would be fixed, and the variations in camber due to deflections would be superimposed on it. The program cannot, however, automatically take account of such effects as chord extension associated with flap rotation and slot openings. The user should, therefore, include some allowance for those effects in the basic camber distribution, and then superimpose simple deflections which do not contribute to changes in the basic camber. Even though the program may be used to approximate such complicated configurations as in Figure 2, the user is reminded that these approximations may be very crude, and rigorous treatment of such details is beyond the scope of the linearized approach. Thus, the program results cannot be expected to accurately predict the aerodynamic characteristics of such configurations.

2.4 Use of Fundamental Cases

An important feature of the EVD jet-wing computer program is the concept of fundamental cases. The use of fundamental cases is intended to increase the

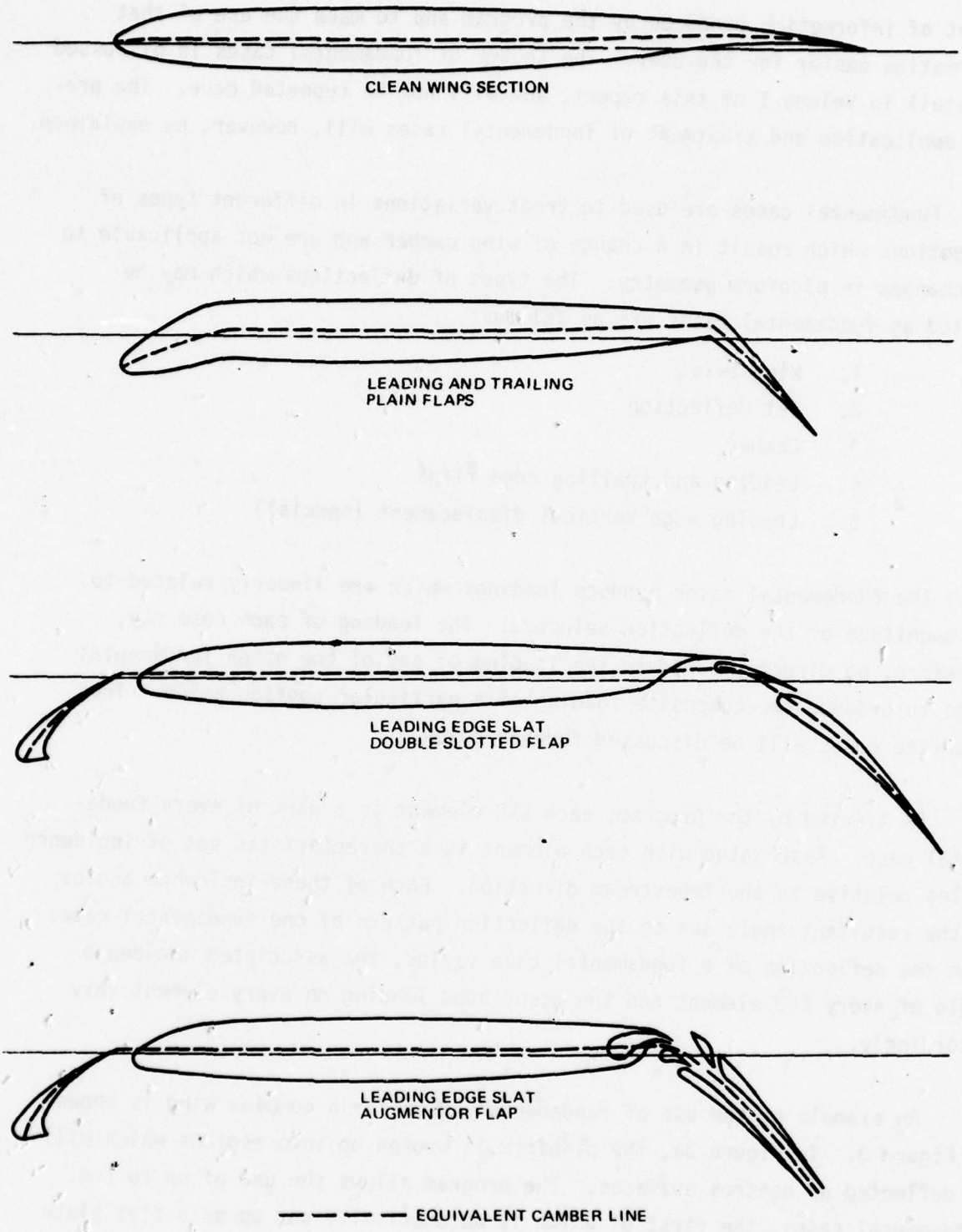


FIGURE 2. REPRESENTATION OF WING SECTIONS BY EQUIVALENT CAMBER

amount of information produced by the program and to make the use of that information easier for the user. The theory of fundamental cases is discussed in detail in Volume I of this report, and will not be repeated here. The program application and treatment of fundamental cases will, however, be explained.

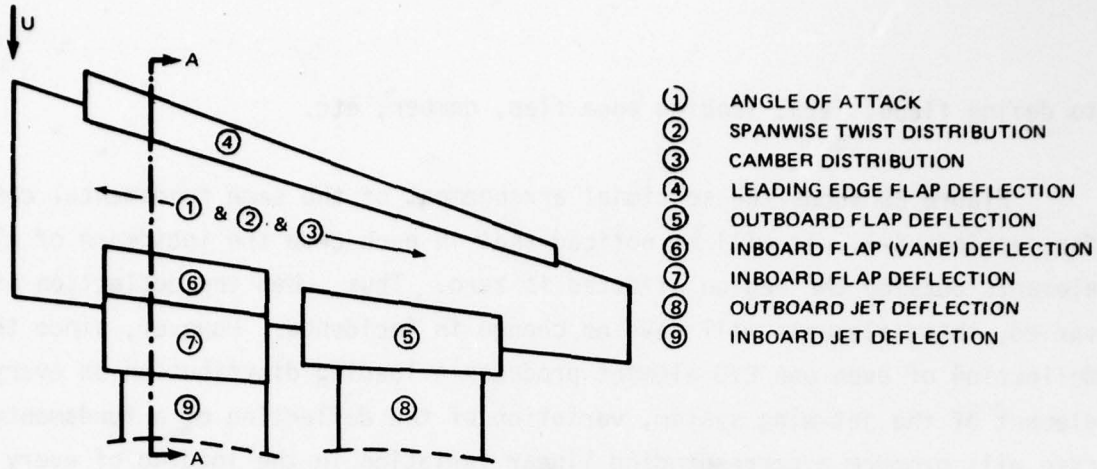
Fundamental cases are used to treat variations in different types of deflections which result in a change of wing camber and are not applicable to any changes in planform geometry. The types of deflections which may be treated as fundamental cases are as follows:

1. Wing twist
2. Jet deflection
3. Camber
4. Leading and trailing edge flaps
5. Leading edge vertical displacement (special)

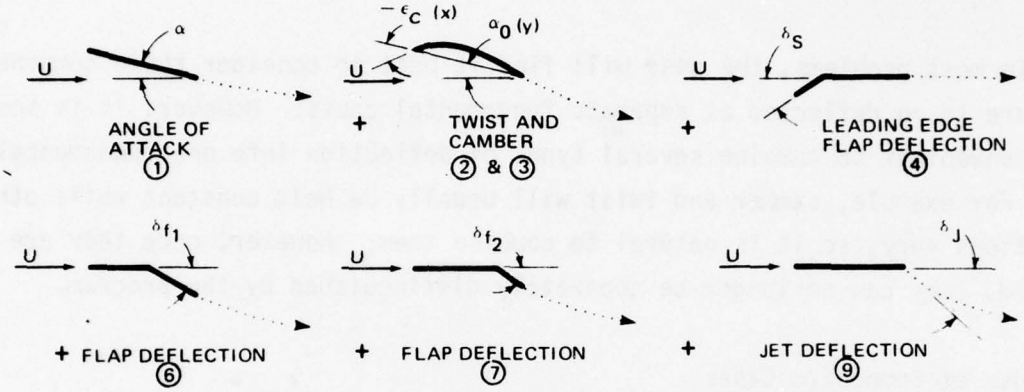
The fundamental cases produce loadings which are linearly related to the magnitude of the deflection selected. The loading of each case may, therefore, be directly added to the loading of any of the other fundamental cases to produce the composite loading of a particular configuration. The composite cases will be discussed further below.

As treated by the program, each EVD element is a part of every fundamental case. Associated with each element is a characteristic set of incidence angles relative to the freestream direction. Each of these incidence angles is the resultant angle due to the deflection pattern of one fundamental case. When the deflection of a fundamental case varies, the associated incidence angle of every EVD element and the associated loading on every element vary accordingly.

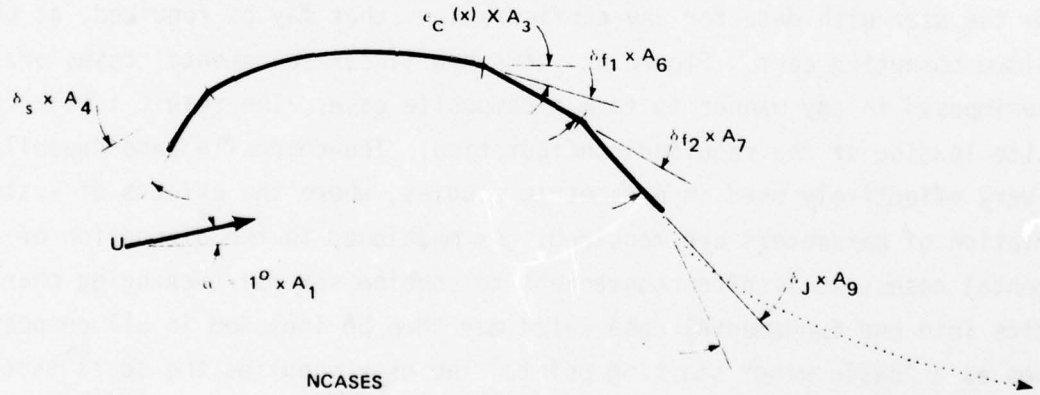
An example of the use of fundamental cases for a complex wing is shown in Figure 3. In Figure 3a, the planform is broken up into regions which will be deflected as control surfaces. The program allows the use of up to ten fundamental cases, the first of which is automatically set up as a flat plate wing at 1° angle of attack. The remaining nine fundamental cases may be used in any manner, and eight of them have arbitrarily been used in the figure



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 3. TREATMENT OF FUNDAMENTAL AND COMPOSITE CASES

to define flaps, jets, leading edge flap, camber, etc.

Figure 3b shows the sectional arrangement of the same fundamental cases for section A-A. It will be noticed that in each case the incidence of all elements outside the region affected 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 of the deflection of a fundamental case will produce a corresponding linear variation in the loading of every EVD element.

In most problems, the user will find it best to consider those components which are to be deflected as separate fundamental cases. However, it is sometimes convenient to combine several types of deflection into one fundamental case. For example, camber and twist will usually be held constant while other deflections vary, so it is natural to combine them. However, once they are combined, they can no longer be separately distinguished by the program.

2.5 Use of Composite Cases

The purpose of the composite case capability in the program is to provide the user with data for any configurations that may be required, at the minimum computing cost. Figure 3c shows how linear fundamental cases are superimposed in any manner to form a composite case. The result is the composite loading of the required configuration. The composite case capability is very effectively used in parametric studies, where the effects of systematic variation of parameters are required. As mentioned in the discussion of fundamental cases, it is often convenient to combine several unchanging characteristics into one fundamental case which may then be included in all composite cases as a "basic wing" starting point. The user supplies the scale factors by which the program is to multiply the incidence and loading of each respective fundamental case before adding them to get the composite incidence and loading. The user is again cautioned, when using composite cases, about exceeding the range of deflections where linear behavior is likely. He should also be careful, when approximating complex sectional geometric characteristics

such as those of a slotted flap system, to limit the range of deflections so as not to significantly change the planform by chord extension.

2.6 Final Comments

The accuracy of the results obtained with this computer program is generally increased as the number of EVD elements increases, assuming that the elements are wisely located by the user. When pursuing accuracy, however, the relative significance of factors which are unaccounted for must be kept in mind. In addition, the computing time is very strongly affected by the number of elements (somewhere between a quadratic and cubic function of the number of elements). Thus, a compromise must be reached between the desire for the best program results on the one hand and, on the other, the practical limitations of computing resources and various unpredictable influences.

It is again strongly recommended that the user read this computing manual thoroughly before attempting the first problem, and also that he read Volume I for valuable background material and insight into the theoretical foundations on which the program is based.

3.0 PROGRAM DESCRIPTION

3.1 General Description

The McDonnell Douglas EVD Jet-Wing Lifting Surface Program is composed of several basic component programs, each of which is independent of the others in logical program flow. These components rely, however, on each other for sharing of information, and all of them are required for satisfactory analysis of a problem. The component programs are tied together by two master routines, only one of which is used in any given run. The choice of which master routine to execute is made by the main routine according to the user's requirements. Figure 4 shows the general functional arrangement of the various program components. The first master routine, called APPLY1, controls execution of ordinary runs, without stability derivative calculations. If stability derivatives are requested by the user, the main routine calls APPLY2, which controls execution for stability derivative runs. Each of the master routines calls subroutines in the four component programs as they are needed.

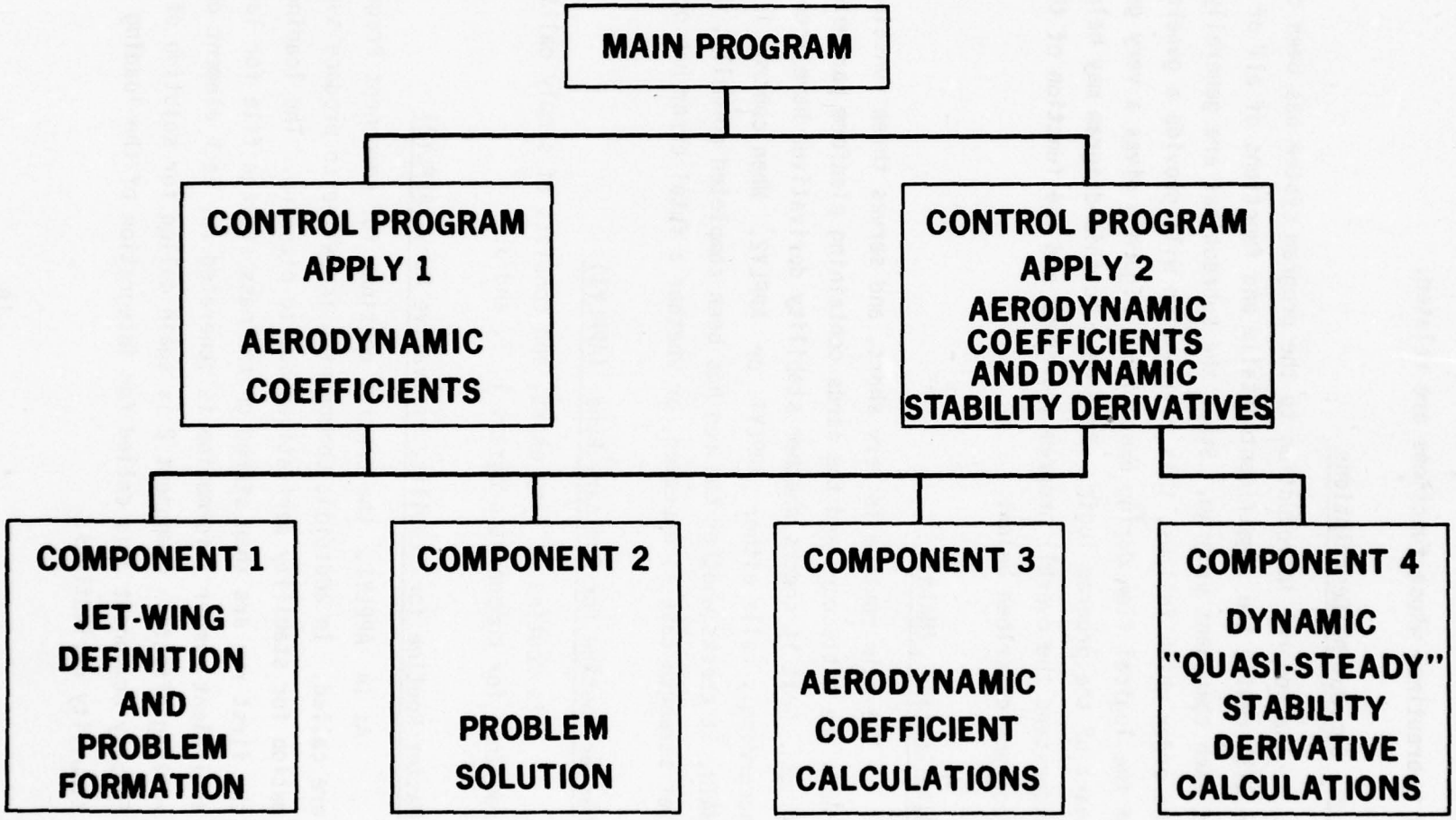
In Component 1, the problem is formed according to the user's input. All input is read, consistency and correctness of input are checked as far as possible, and all geometric parameters are defined.

In Component 2, the mathematical problem is solved. The system of simultaneous linear equations is formed using the downwash influence coefficients of each EVD element. The matrix system is then solved directly, and the solution is achieved in the form of the vorticity strength of each EVD element of the jet wing system.

In Component 3, the vorticity solution is translated into useful aerodynamic coefficients and/or stability derivatives. These include chordwise loading, spanwise distribution of lift, drag and pitching moment, and total lift, drag, and static and dynamic moments.

Component 4 contains supplemental subroutines which are used during calculation of dynamic stability derivatives. This is not a self-contained component program as are the others, but is simply an unstructured collection

FIGURE 4.
GENERAL PROGRAM FUNCTIONAL ARRANGEMENT



of subroutines whose functions are related.

3.2 Subroutine Descriptions

The general introduction to the program system has been completed. Next we will review the significant details and functions of all of the subroutines of each component program. Since the subroutines are generally executed in the order which follows, this review also will provide a general description of the logical flow during execution. Figure 5 gives a very general flow chart of the program logic. Reference to this diagram may help the reader to understand the overall program operation and the function of the various subprograms described below.

Main Routine (MAIN)

The main routine is very short, and serves three functions. It first reads the title card and the cards containing planform parameters and control flags. Next it checks whether stability derivatives have been requested and accordingly calls either APPLY1 or APPLY2. When control is returned to MAIN, it checks whether the run has been completed normally, whether a new jet strength case is expected, or whether a fatal error has occurred.

Master Routine for Ordinary Runs (APPLY1)

This routine is very short, and consists of simply calling the control routines for component programs 1, 2, and 3.

Master Routine for Stability Derivative Runs (APPLY2)

As in APPLY1, the control routines for component Programs 1, 2, and 3 are called. In addition, Component 4 is called to produce supplementary information for stability derivatives due to pitching. The loading solutions for the first run are then stored on the mass storage file for later use, and new equivalent camber information is generated for each element due to yawing and rolling effects. Component 2 is again called for solution of the matrix system. Finally Component 3 is called for integration of the loading to get dynamic stability derivatives.

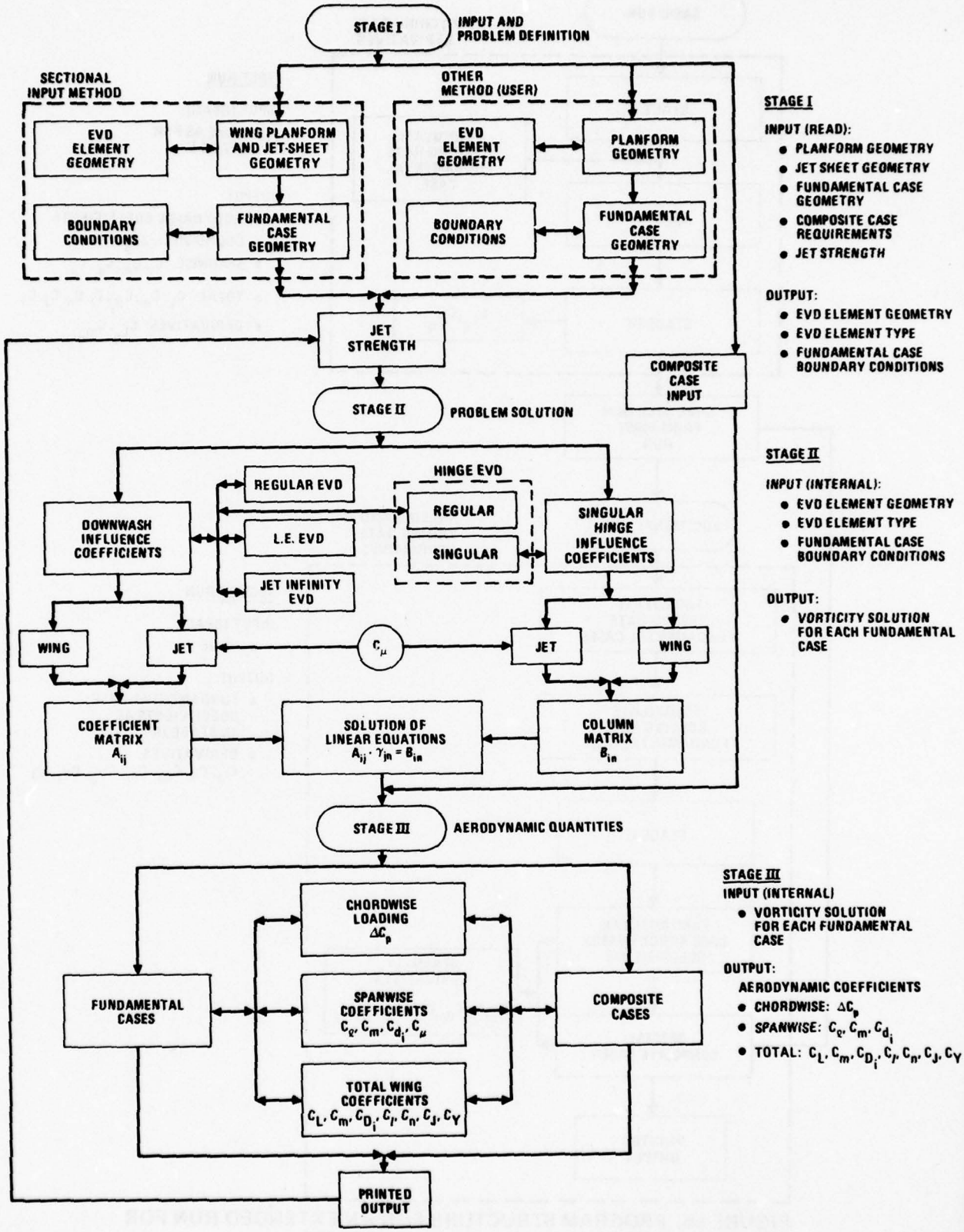
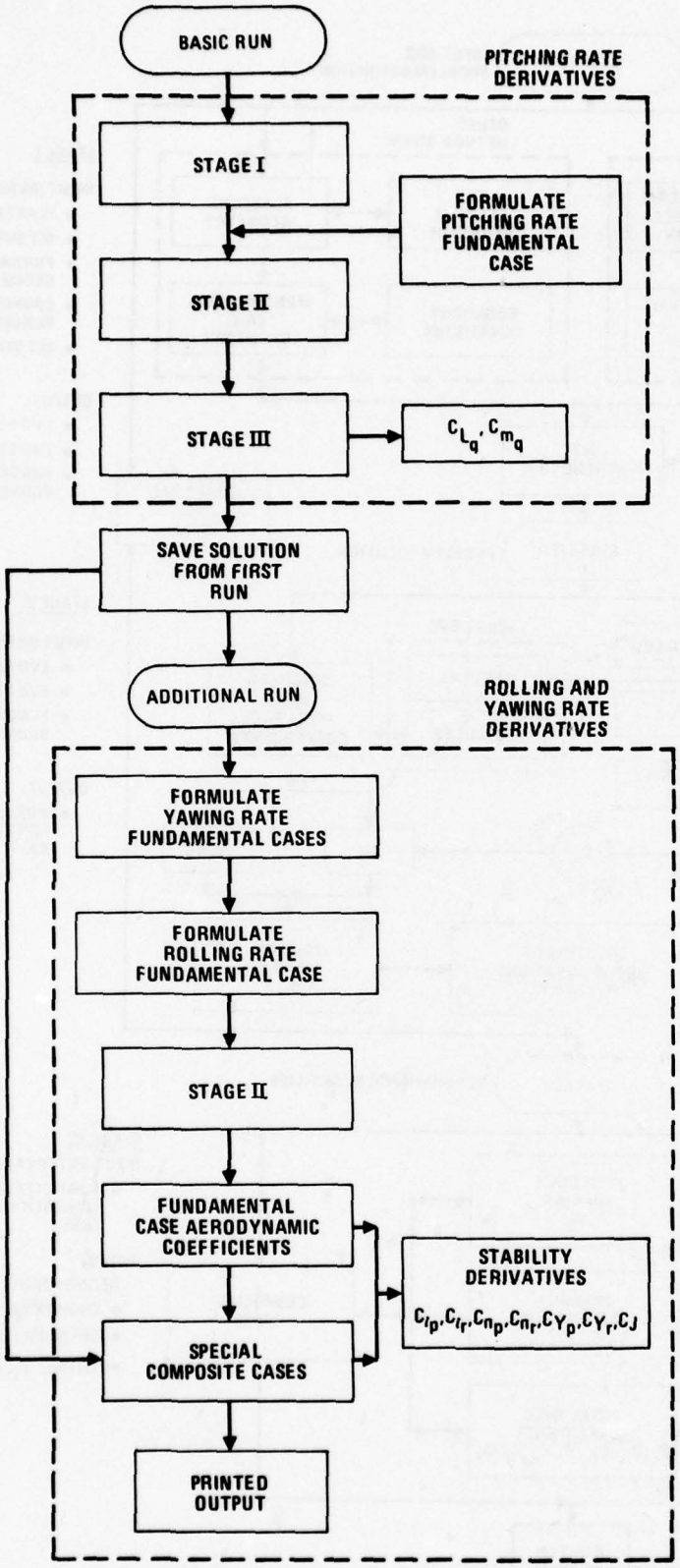


FIGURE 5a. PROGRAM STRUCTURE FOR A BASIC RUN



FIRST RUN

INPUT (READ):

- SAME AS FOR STAGE I

OUTPUT:

AERODYNAMIC COEFFICIENTS

- CHORDWISE: ΔC_p
- SPANWISE: C_ℓ, C_m, C_{d_i}, C_μ
- TOTAL: C_L, C_m, C_{D_i}, C_l, C_n, C_J, C_Y
- DERIVATIVES: C_{L_q}, C_{m_q}

SECOND RUN

INPUT (READ):

- NONE

OUTPUT:

- FUNDAMENTAL CASE COEFFICIENTS AS IN STAGE III
- DERIVATIVES: C_{l_p}, C_{l_r}, C_{n_p}, C_{n_r}, C_{Y_p}, C_{Y_r}, C_J

FIGURE 5b. PROGRAM STRUCTURE FOR AN EXTENDED RUN FOR DYNAMIC STABILITY DERIVATIVES

Control of Problem Formation (STAGE1)

The control subroutine (SGMAIN) for dealing with sectional geometry input and element definition is called after the validity of certain control flags has been checked. Alternate methods of input and element definition may be the user-supplied optional routines for any specified values of the IGTYPE flag. (See Section 4.1 - Input Data Instructions). Next the composite case requirements are read by calling INCOMP. Subroutine BLOWIN is called to read the jet strength requirements, and finally BOXJ is called to prepare additional jet parameters for internal use later in the program.

Sectional Input Method Control (SGMAIN)

The spanwise and chordwise spacing of wing elements are read by calling subroutine INPTS. Next the planform is defined by calling either XLETR1 for irregular planforms, or XLETR2 for simple trapezoidal planforms. The planform parameters are normalized by half the input value of wing span in subroutine NORM1. Inputs defining the spanwise and chordwise spacing of elements on the jet are read by calling subroutine INPUTJ. Subroutine BOXS next combines all of the element and planform input data acquired up to this point in order to define the network of elements required by the program. At this point, the jet-wing planform arrangement of EVD elements is completely defined, but it remains to specify the pattern of deflections required by the user. This is accomplished by a sequence of calls to subroutines INCASE, BEECEE, and OUT1 for each required fundamental case. The inputs describing the elements to be deflected and the amount of deflection are read in INCASE. BEECEE interprets this data and derives certain parameters to be used later in the program. OUT1 prints the resulting parameters for each element.

If shortcut methods can be devised for easier input of all or part of the geometrical data required for a class of wings of particular interest, some of the routines above may be replaced, modified or made optional, as the user requires. However, the result must be the same data as produced by the present system in the correct form for use by remaining components of the program.

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 very short subroutine simply 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, resultant incidence angle, hinge deflection angle, if any, and EVD type 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.

Jet 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 Parameter Definition (BOYJ)

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.

Control of Problem Solution (STAGE2)

This routine simply calls the two overlay segments which set up the system of simultaneous linear equations and solve the matrix system.

Development of Simultaneous Equations (STG2D)

The large square left-side matrix of 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 hinge EVD elements 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. Subroutine PREP then prepares the complete linear equation system for solution by assembling the left and right side matrix data together on tape in the proper form for use by the solution subroutine.

Downwash Influence Coefficients (DWNWSH)

This subroutine calculates the downwash on every element due to the influence of all the other elements. 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, or EVD3 (hinge EVD not yet considered). If the wing is symmetric or anti-symmetric, the appropriate influence coefficients are superimposed (added or subtracted) to reduce the size of the linear system to half the number of equations. The array of influence coefficients on the particular element under consideration forms one row of the coefficient matrix, and these rows are stored, one by one as they are computed, on the mass storage device, Unit 1. Rows corresponding to elements on the wing are stored first, then rows corresponding to elements on the jet.

Regular Downwash (EVD1)

This function computes the downwash at any point in the wing plane due to a triangular distribution of vorticity on an element located anywhere on the wing (see Figure 1).

Leading Edge Downwash (EVD2)

This function computes the downwash at any point in the wing plane due to a square root singular distribution of vorticity on an element located at the leading edge of the wing (see Figure 1).

Jet Infinity Downwash (EVD3)

This function computes the downwash at any point in the 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 (see Figure 1).

Hinge EVD (EVD4)

This function computes the downwash at any point in the 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 1).

Initial Matrix Augmentation (SHUFL1)

This subroutine reads the matrix rows corresponding to elements on the jet, subtracts adjacent rows from each other, and further modifies certain elements near the main diagonal due to the nonlinear influence of jet strength. The augmented rows are then replaced on mass storage Unit 1 immediately behind the original downwash matrix rows.

Additional Matrix Augmentation (SHUFL2)

This subroutine reads the augmented matrix rows from Unit 1, remodifies certain elements near the main diagonal, then restores the rows in their original locations.

Column Matrix Formation (COLUM1)

This subroutine defines the right side column matrix of constant boundary conditions. For elements on the wing, the values are the total incidence angles relative to the freestream, in radians. For the first element of each jet section, the value is the total trailing edge incidence angle plus the jet turning angle, in radians, and for all other jet elements, the values are zero. Each column corresponds to one fundamental case.

Hinge Downwash Influence (HINGE)

For each element on the wing and jet, the downwash influence of each hinge singularity distribution is computed by calling EVD4, added or subtracted according to symmetry requirements, then multiplied by the appropriate hinge turning angle. The influence due to all such hinges are summed as they are calculated, so that one hinge influence factor is produced for each element

on the wing and jet. This is done for one column (fundamental case) at a time.

Column Matrix Hinge Augmentation (COLUM2)

This subroutine adds to the right side column matrix row corresponding to each element on the wing and jet, the hinge downwash influence factor computed in HINGE. For rows corresponding to elements on the jet, additional modifications are made to the hinge factors before adding them to the right side matrix in order to account for the influence of jet strength on the vorticity of each jet hinge element. These operations are done for one column (fundamental case) at a time.

Matrix Preparation for Solution (PREP)

The form in which the matrix solution subroutine expects to find data is row by row, left side then right side on each row, stored on a sequential scratch file. Thus the augmented left side rows are read one by one from the mass storage Unit 1 into a "transfer" array, the elements of the right side matrix are defined as the last values of the transfer array, and the entire array is written as one record onto Scratch Unit 2. In this way all the rows corresponding to elements on the wing and then the jet are assembled or "concatenated" and written to Unit 2 to form the entire matrix system.

Control of Matrix Solution (STG2S)

This control routine calls the matrix solution routine, MATRIX, retrieves the vorticity solution, and calls the back substitution check subroutine if requested.

Matrix Solution (MATRIX)

This subroutine solves the matrix system directly by triangularization. It is able to solve very large matrix systems using a relatively small amount of core storage because it deals with only a portion of the system at one time,

the remainder being stored on scratch files.

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.

Matrix Back Substitution (BAKSUB)

As a check of the matrix solution accuracy, the user may wish to call for a back substitution check. In a row by row manner, the augmented left side matrix is read from mass storage Unit 1 and the corresponding elements are multiplied and summed to form the right side matrix. This right side matrix is printed for reference and should agree with the boundary conditions printed earlier.

Control of Aerodynamic Coefficients (STAGE3)

This routine controls computation of all aerodynamic coefficients, including stability derivatives, for both regular runs and stability derivative runs. If stability derivatives are not requested in the current run, chordwise loading, spanwise loading, and total coefficients are computed by calling STG3FC and STG3FS, respectively, for each fundamental case. Then a summary table of total coefficients for all fundamental cases is printed by calling STG3FT. For each composite case requested, the composite loading routine, STG3C, computes and prints spanwise loading and total aerodynamic coefficients. If stability derivatives are requested, subroutines FUNDER and CØMDER are called, which compute stability derivative coefficients for all fundamental and composite cases, respectively.

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 singular leading edge and hinge EVD elements.

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. If requested, all the above coefficients are then printed. 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. Since no coefficients have been computed for unused fundamental cases, they are initialized to zero before printing.

Composite Case Loading Control (STG3C)

The composite chordwise loading is first computed by superimposing the loading from the required fundamental cases, each multiplied by its respective input scale factor. The loading is also printed, if requested. 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. All spanwise and total aerodynamic coefficients are then printed, including a summary table of all total coefficients. 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 are repeated for each composite case requested.

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 are computed for leading edge singularities and the leading and trailing parts of hinge singularities, respectively.

Spanwise Loading (SLOAD)

This subroutine integrates the chordwise loading at each section to arrive at the sectional values of lift, induced drag and pitching moment. Because of the general nature of the computations, the loading for either fundamental or composite cases can be computed, depending on the element loading given.

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.

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.

Total 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. Only total coefficients for fundamental cases are computed.

Total Loading (TLOAD0)

This subroutine computes total lift and pitching and rolling moments 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.

Total Loading (TLOADX)

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

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.

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.

Fundamental Case Stability Derivatives (FUNDER)

This subroutine controls calculation of stability derivatives for all fundamental cases. Subroutine STG3FC is first called to generate the chordwise loading for the stability fundamental cases. Then both chordwise and

spanwise integration are performed by calling subroutines SUMIT1 and SUMIT2, to yield stability derivatives due to rolling and yawing rates. For the last fundamental case (due to rolling rate) certain stability derivatives are derived directly from chordwise and spanwise loading in the normal manner by calling STG3FS.

Composite Case Stability Derivatives (COMDER)

This subroutine controls computation of stability derivatives for all composite cases. First the composite case loading is assembled, requiring reading of the fundamental case solutions for the first run from mass storage Unit 1. Then the subroutines SUMIT1 and SUMIT2 are called for chordwise and spanwise integration of the loading to obtain the required stability derivatives. Finally a table is printed showing the components of the stability derivatives and the variation of the derivatives with angle of attack.

Stability Derivative Integration (SUMIT1)

This subroutine computes the derivatives of yawing moment due to rolling, and rolling moment due to yawing by chordwise and spanwise integration of the loading on each element.

Stability Derivative Integration (SUMIT2)

This subroutine computes the derivative of yawing moment due to yawing by chordwise and spanwise integration of the loading on each element.

Stability Derivative Table (STABLE)

This subroutine prints, for each composite case, a summary of all 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.

Control of Utility Routines for Stability Runs (STAGE4)

During the first run of a stability derivative run sequence, an extra

fundamental case is created with induced camber intended to simulate a rate of pitching. This is accomplished by calling subroutine BCPICH, and all the resulting geometric data is printed by calling OUT2. During the second run of a stability derivative run sequence, the fundamental case solutions from the first run are saved on mass storage Unit 1 in subroutine SAVECP. The fundamental case geometry is then redefined, with induced camber simulating rates of yawing and rolling, by subroutines BCYAW and BCROLL, respectively. Again, OUT2 gives a record of the new fundamental case geometry.

Fundamental Case Output (OUT2)

This subroutine is the same as subroutine OUT2, and prints the geometric data defining the stability derivative fundamental cases.

Save First 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, directly behind the matrix information previously stored.

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.

3.3 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 of Reference 1), 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*	$\frac{x_i - x_i}{\frac{1}{2}\left(\delta_j + \delta_{j-1}\right)} > 100$	$a\left(\infty, \left(y_i - y_j\right)\right) = -\frac{1}{4\pi}\left(\delta_j + \delta_{j-1}\right)\left(\frac{1}{y_i - y_j - \Delta} - \frac{1}{y_i - y_j + \Delta}\right)$
	$x_i - x_j = 0,\ y=0$	Not encountered
	$x_i - x_j = -\delta_{j-1},\ y=0$	Not encountered
	$x_i - x_j = \delta_j,\ y=0$	Not encountered

*Reference 1: Appendix I, Section I.1

EVD Function	Region	Value Used
Leading* Edge	$\left(x_i-x_j\right)/\delta_j > 100$	$a\left(\infty,\left(y_i-y_j\right)\right) = -\frac{1}{2\pi}\left(\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right)$
	$\left\|\left(x_i-x_j\right)/\delta_j\right\| < 10^{-4}$ $\left\|\left[\left(x_i-x_j\right)/\delta_j\right]-1\right\| < 10^{-6}$	$a\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = a\left(0,\left(y_i-y_j\right)\right)$
Infinity+	$\left(\frac{x_i-x_j+d}{y_i-y_j-\Delta}\right) > 10^6$	$a\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = -\frac{1}{2\pi}\left(\frac{\delta_{j-1}}{2}+d\right)\left(\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right)$
	$\left\|\frac{x_i-x_j+d}{d}\right\| < 10^{-2}$	$a\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = a\left(-d,\left(y_i-y_j\right)\right)$
Hinge‡ (Singular Part)	$\left\|\frac{x_i-x_j}{\frac{1}{2}\left(\delta_j+\delta_{j-1}\right)}\right\| > 7.5$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = \frac{\beta}{\pi^2}\left[-\left(\delta_{j-1}+\delta_j\right)-\left(\delta_{j-1}\log\delta_{j-1}+\delta_j\log\delta_j\right)\right]\left[\frac{1}{y_i-y_j-\Delta}-\frac{1}{y_i-y_j+\Delta}\right]$
	$\left\|\frac{x_i-x_j}{\frac{1}{2}\left(\delta_j+\delta_{j-1}\right)}\right\| < 10^{-4}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(0,\left(y_i-y_j\right)\right)$
	$\left\|\frac{x_i-x_j+\delta_{j-1}}{\delta_{j-1}}\right\| < 10^{-6}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(-\delta_{j-1},\left(y_i-y_j\right)\right)$
	$\left\|\frac{x_i-x_j-\delta_j}{\delta_j}\right\| < 10^{-6}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(\delta_j,\left(y_i-y_j\right)\right)$
	$\left\|\frac{\delta_{j-1}}{\left\|x_i-x_j\right\|}-1\right\| < 10^{-4}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(\delta_{j-1},\left(y_i-y_j\right)\right)$
	$\left\|\frac{\delta_j}{\left\|x_i-x_j\right\|}-1\right\| < 10^{-4}$	$b\left(\left(x_i-x_j\right),\left(y_i-y_j\right)\right) = b\left(\delta_j,\left(y_i-y_j\right)\right)$

* Reference 1: Appendix I, Section I 2
\+ Reference 1: Appendix I, Section I 3
‡ Reference 1: Appendix I, Section I 4

EVD Function	Region	Value Used
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: Values of Hinge EVD regular part are equivalent to Regular EVD.

† Reference 1: Appendix I, Section I 4

4.0 INPUT DATA INSTRUCTIONS

The program input consists of various types of cards containing all the information required to define a particular problem. The number of cards of each type may vary from problem to problem, but the sequence is the same in all runs. As a result of options chosen by the user, some types of cards may not be required, but the sequence of the remaining cards must not be changed. Load sheets for all cards are shown in Appendix D. The input deck is shown diagrammatically in Figure 6.

For symmetric and anti-symmetric wings, all input and output data are for the right half of the wing only. Figure 7 identifies some of the planform coordinates required as input. These will be discussed in detail below. Sectional input normally begins at the right wing tip section, working inboard section by section. However, most data required for description of the jet is only needed at sections where the jet exists, and unblown sections should be skipped as indicated in the instructions below.

Figure 8 shows the reference coordinate system used for both input and output data. This is a wind axis system with the freestream direction always aligned with the x-axis. 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) with respect to the x-axis. All wing sections will still be aligned with the x-axis, including the wing 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 a body axis system.

In the input description which follows, all input values are real floating point numbers (F format), unless otherwise specified. All specified integers should be right-justified in their fields.

4.1 Input Description

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

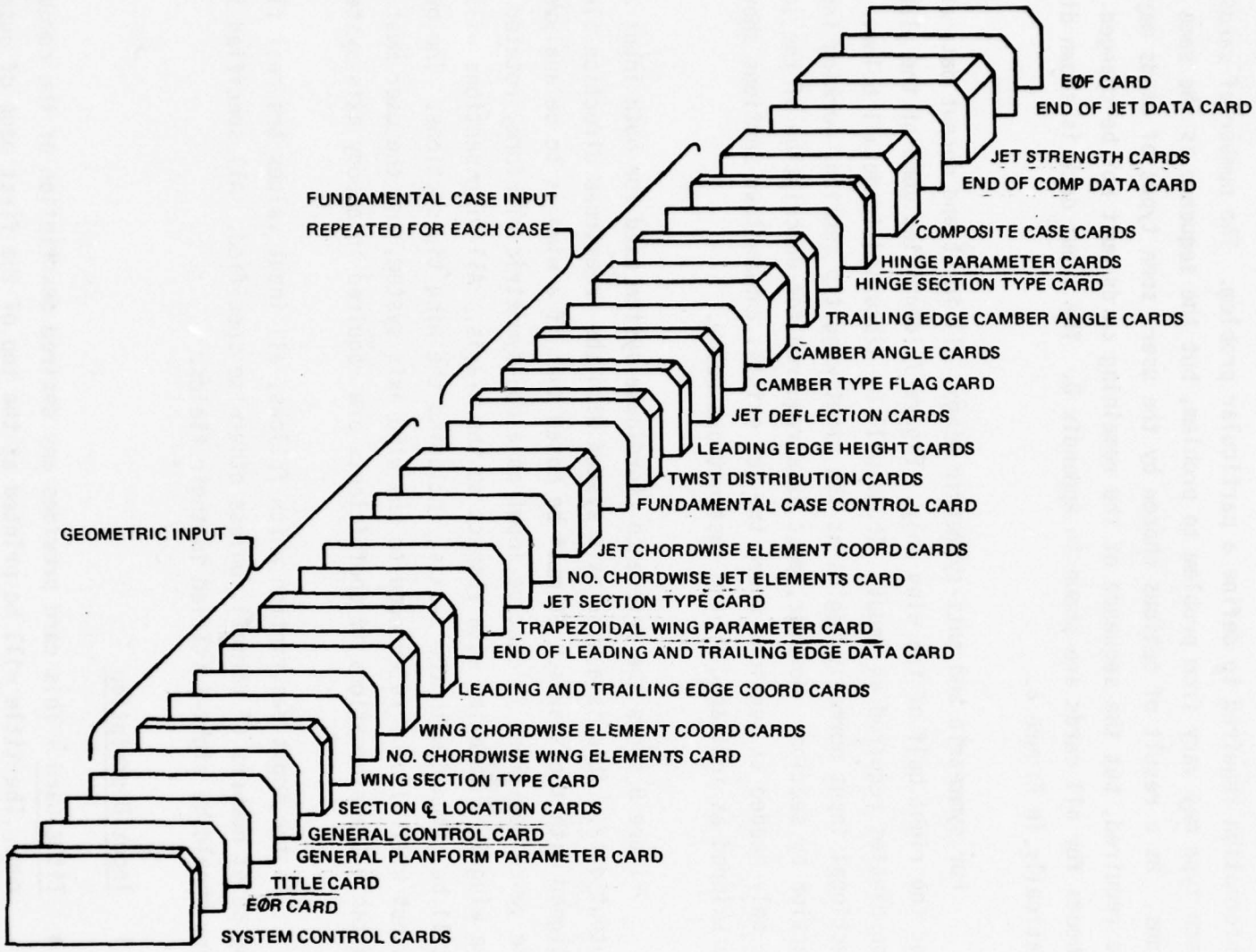


FIGURE 6. ARRANGEMENT OF INPUT CARD DECK

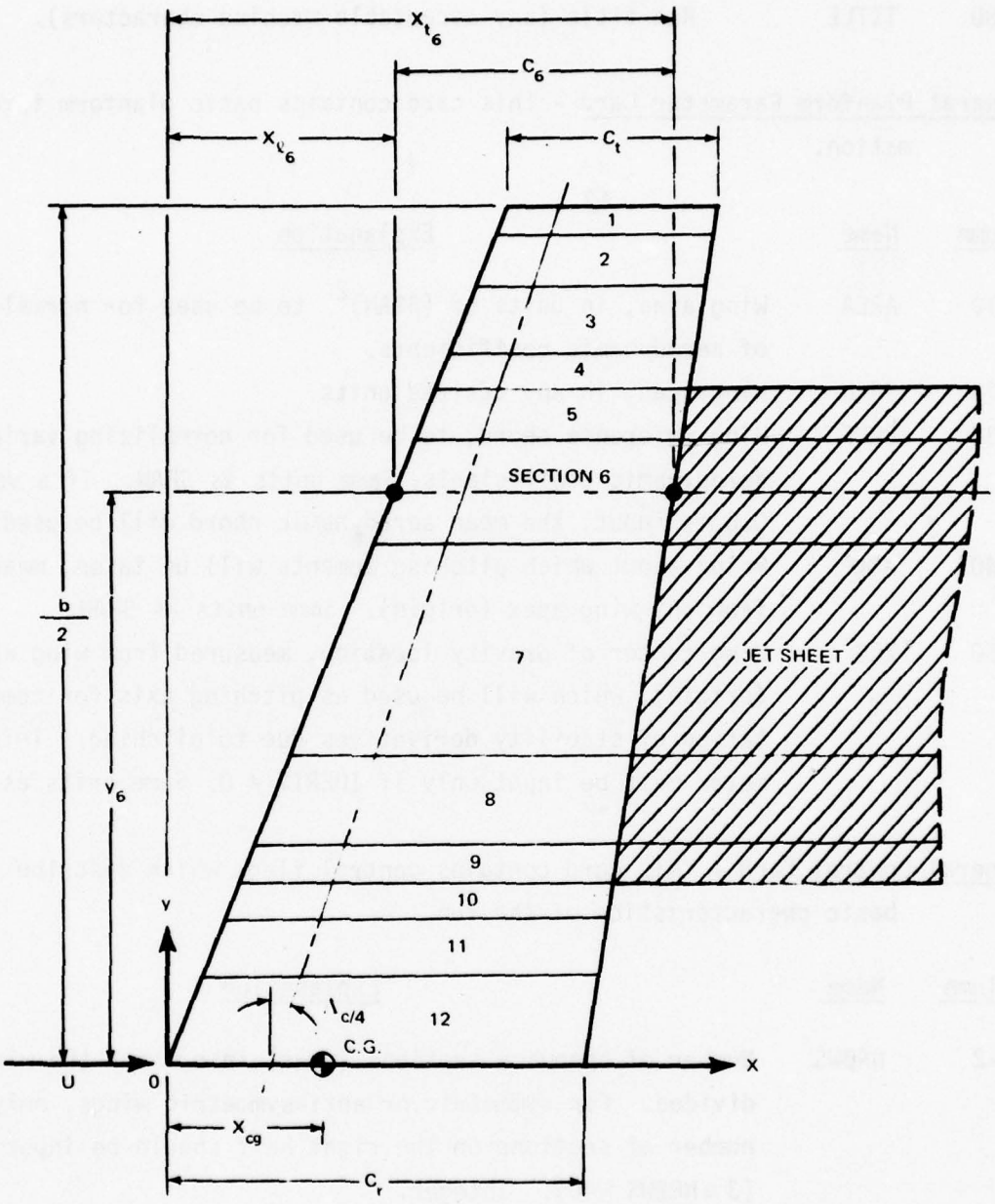


FIGURE 7. PROGRAM PLANFORM DIMENSIONS AND NOTATION

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

- General Planform Parameter Card - This card contains basic planform information.

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. Same units as 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). Same units as 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. Same units as SPAN.

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

Column	Name	Explanation
1-2	NRØWS	Number of spanwise sections (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 ($3 \leq$ NRØWS ≤ 40). Integer.
3-4	NCASES	Total number of fundamental cases. It must be noted that the angle of attack case is always set up automatically as fundamental case number one, and no input data is required for it. Therefore, NCASES must be one more than the number of cases for which input data will be given, to allow for the angle of attack case. ($1 \leq$ NCASES ≤ 10). Integer.

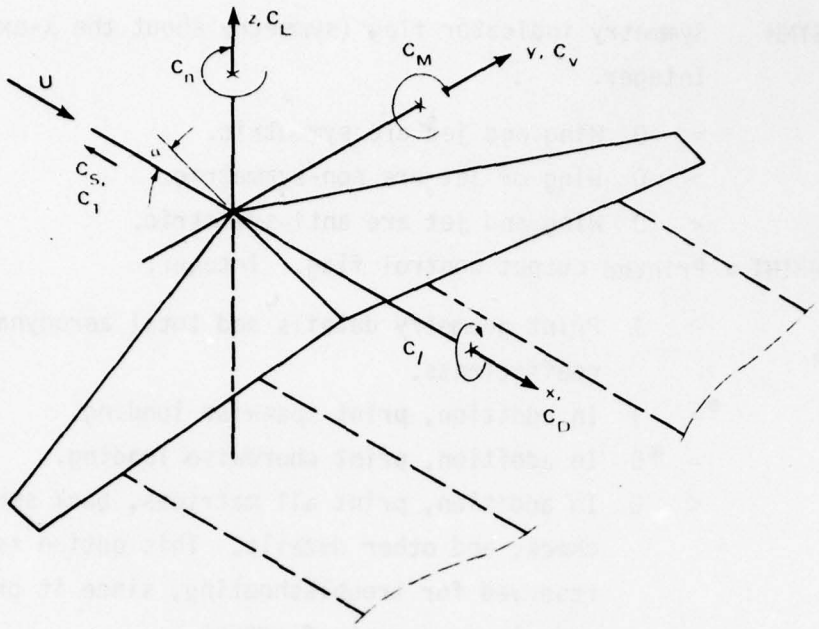


FIGURE 8a. REFERENCE COORDINATE SYSTEM FOR AERODYNAMIC COEFFICIENTS

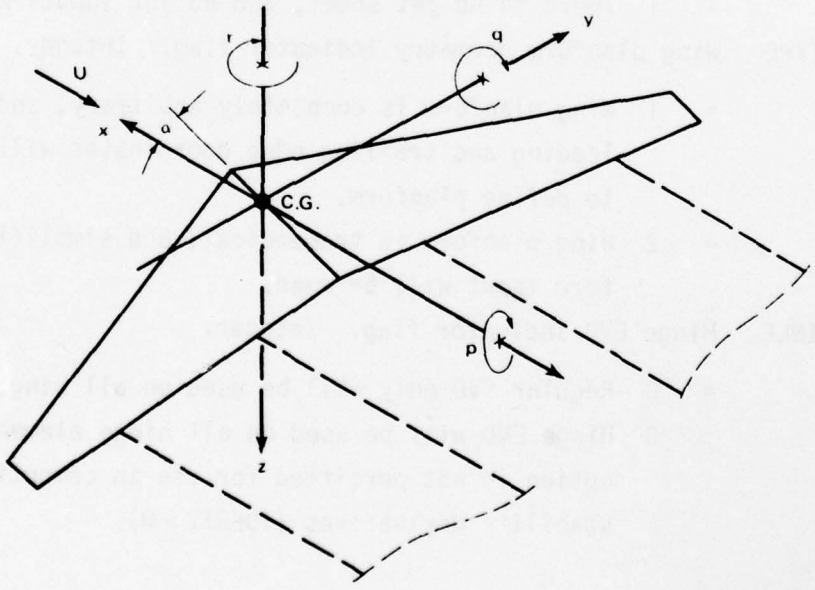


FIGURE 8b. REFERENCE COORDINATE SYSTEM FOR STABILITY DERIVATIVES

FIGURE 8. REFERENCE COORDINATE SYSTEM

Column	Name	Explanation
5-6	ISYMM	Symmetry indicator flag (symmetry about the x-axis). Integer. = 0 Wing and jet are symmetric. > 0 Wing or jet are non-symmetric. < 0 Wing and jet are anti-symmetric.
7-8	IPRINT	Printed output control flag. Integer. > 1 Print geometry details and total aerodynamic coefficients. = 1 In addition, print spanwise loading. = 0 In addition, print chordwise loading. < 0 In addition, print all matrices, back substitution check, and other details. This option is normally reserved for troubleshooting, since it produces a very large amount of output.
10	JETFLG	Jet indicator flag. Integer. = 0 There is a jet sheet. = 1 There is no jet sheet, and no jet inputs will be read.
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.
14	IHINGE	Hinge EVD indicator flag. Integer. = 0 Regular EVD only will be used on all hinge elements. > 0 Hinge EVD will be used on all hinge elements. This option is not permitted for use in computing dynamic stability derivatives (IDERIV > 0).

Column	Name	Explanation
16	IDERIV	Dynamic stability derivative control flag. Integer. = 0 A basic run will be executed, with no stability derivatives computed. > 0 A basic run will be executed, and in addition, a dynamic stability derivative run will be executed. This option requires approximately double the basic computing time.

- 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 five cards (40 sections) allowed.

Column	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). NRØWS values must be input, beginning at the right wing tip and working to the following: (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.

Column	Name	Explanation
1-2, 3-4, etc.	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 locations of all EVD elements, have the same ICTYPE value. NRØWS values must be input. There may not be more than ten different values (i.e., section types). The highest value input is called NWTYPE, and all values less than NWTYPE

Column	Name	Explanation
		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.

Column	Name	Explanation
1-2,3-4, etc. Up to CC10	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 two but not more than twenty chordwise EVD elements must be input on each wing section. ($2 \leq NI \leq 20$). Integer.

- 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 ten sets, three cards per set.

Column	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 for all Leading edge EVD's and the "peak" point for all Regular and Hinge EVD's. 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 600.

- 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 tip and root sections must be input. The leading and trailing edges are interpolated for sections not given by putting 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 ≤NRØWS.

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). Same units as SPAN.
21-30	XTRAIL	Chordwise distance from section trailing edge (at section centerline) to the wing apex (y-axis). Same units as SPAN.
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.

- Trapezoidal Wing Parameters - This card contains planform information for simple trapezoidal wings. It replaces the above coordinate cards including the 9 card when IGTYPE = 2. This type of input may be used only when the wing planform is symmetric.

Card Column	Variable Name	Explanation
1-10	ARATIØ	Wing aspect ratio, $(SPAN^2/AREA)$.
11-20	SWEEP	Sweep angle of wing quarter-chord line, in degrees.
21-30	TR	Wing taper ratio, $\left(CHØRD_{\substack{wing \\ tip}} / CHØRD_{\substack{axis\ of \\ symmetry}}\right)$

- 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 .

Column	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 .

Column	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 jet 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.

Column	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 ($2 \leq NI \leq 10$).

- Fundamental Case Control Card - This card identifies the types of linear geometric variations to be included in each fundamental case. The number of fundamental cases input must be one less than NCASES, to allow for the angle of attack case. A new fundamental case control card is required for each of the (NCASES - 1) input cases. 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.
8	INCAMB	Camber flag. Integer.
10	INBETA	Wing hinge deflection flag. Integer.

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

Column	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. See *Figure 3b* ②.

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

Column	Name	Explanation
1-10,11-20, etc.	HO	Displacement coordinate 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 and angle of attack are accounted for automatically by the program.

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

Column	Name	Explanation
1-10,11-20, etc.	DJ	Jet turning angle, in degrees, relative to the mean line of the trailing edge. Positive deflection is downward. See Figure 3b ⑨.

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

Column	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. NRØWS 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.

Column	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, including the trailing edge). Positive values are in the same sense as positive angles of attack. See Figure 3b ③.

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

Column	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. NRØWS values are required. Required only if INBETA ≠ 0.

Column	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), their type (leading or trailing flap) and in all deflections. There may be as many as NRØWS 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 edge flap) and turning angle at 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.

Column	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.
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. See Figure 3b ④, ⑥, ⑦.

- 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). See Figure 3c.

Column	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).

- Jet Strength Cards - These cards contain the jet strength for all sections which have a jet. NROWSJ values are required. An unlimited number of sets of cards may be input, maximum of five cards per set. Required only if JETFLG = 0 .

Column	Name	Explanation
1-10,11-20, etc.	CMU	Sectional jet momentum strength for each jet row. CMU is defined as $CMU = J/qc(y)$, where J is the sectional jet momentum per unit span 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).
1	9	A "9" must appear in CC1 of the card following all CMU cards to signal the end of CMU input. Required only if JETFLG = 0 .

4.2 Input Restrictions

Most of the limitations imposed on the range of input values are designed to minimize computer storage requirements, but still to provide the capability for adequate treatment of complicated configurations. While all of the input restrictions have been stated or implied in the input description section above, they will be summarized here for convenient reference.

1. $3 \leq NRØWS \leq 40$
2. $1 \leq NCASES \leq 10$
3. IGTYPE = 1 or 2
4. Number of chordwise wing elements on any section $2 \leq NI \leq 20$
5. Number of chordwise jet elements on any section $2 \leq NI \leq 10$
6. Maximum of ten different section types each on wing and jet
7. On wing $0.0 \leq XB < 1.0$
8. On jet $1.0 \leq XB$
9. Only NROWSJ values required for DJ, ACTE and CMU input
10. $0.0 \leq CMU < 800.0$
11. For stability derivative runs, only one set of CMU data may be input

Due to certain curve fitting restrictions in the calculation of induced drag by the momentum method, at least three adjacent sections must be used in defining either a jet sheet or a region on the wing which has no trailing jet sheet. Thus, a jet cannot consist of only one or two sections. Likewise, inboard or outboard of any jet, there must be at least three unblown wing sections. If the three-in-a-row rule is not followed, the program will stop automatically.

5.0 OUTPUT DATA DESCRIPTION

The program output consists entirely of data written to the standard print Unit 6. Four print options, controlled by the input value of the IPRINT flag described above, are available to the user. The normal printout is given for the default value of zero, and includes chordwise and spanwise loading and total wing aerodynamic coefficients. If the details of chordwise or spanwise loading are not required, a large part of the print and a slight amount of computing time may be saved by requesting one of the more restricted forms of output.

If more information is required for program checkout or troubleshooting, the extended print option may be used. Under this option, all of the pertinent data of the problem solution are printed, including downwash coefficients, augmented matrix coefficients, vorticity solution and back substitution check for recovery of the right side matrix. The matrix information is given in a row by row sequence, each row corresponding to one EVD element. For each row the influence coefficients of all EVD elements on the one in question are listed across the page. For example, the downwash influence due to Element 12 on Element 6 would be the second number of the second line listed under matrix Row 6.

The detailed print option, when used in a stability derivative run, will also give for the second run the chordwise loading, spanwise loading of the last fundamental case, and the summary table of total coefficients for all fundamental cases. These data are not intended for direct use, but mainly for indirect checking of the consistency of certain stability derivative results.

It will be helpful for the user to refer to the sample problem output of Appendix A during discussion of the output data given below. All output symbols and labels are listed in the nomenclature section and all angles are measured in degrees.

Even though many significant digits are printed, this is by no means meant to imply that the coefficients are accurate to such a degree. Six or eight significant digits are printed only so that the effects due to small

deflections or planform changes will be visible in the F format. 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.

The composite case linear aerodynamic characteristics are lift, pitching moment, root bending moment, and rolling moment. They are composed of a value at zero angle of attack and a linear variation with angle of attack, and may be calculated as follows:

$$C_L = C_{L_{\alpha=0}} + C_{L_\alpha} \alpha$$

(CCL = CCLO + CCLA * ALPHA)

The nonlinear characteristics include all of the drag contributions, yawing moment and side force, and are composed of a value at zero angle of attack, a square variation with angle of attack, and a "cross-product" linear variation with angle of attack as follows:

$$C_D = C_{D_{\alpha=0}} + C_{D_\alpha} \alpha + C_{D_{\alpha^2}} \alpha^2$$

(CCD = CCDO + CCDX * ALPHA + CCDA2 * ALPHA ** 2)

The proper methods of composition of the dynamic stability derivatives vary greatly, and are shown in the printout for each of the types of derivatives.

As noted in the input data instructions, the reference system of both input and output data is a wind axis system, with the x-axis aligned with the freestream. If it is required to analyze a yawed wing, the wing must be input in the yawed position and considered as non-symmetric. If aerodynamic coefficients are required in a body axis system, they must be transformed by the user. In the linearized analysis, the only contribution to side force should be the leading edge suction, taken normal to the local leading edge.

The program automatically normalizes all the geometry data by dividing by half the input value of the wing span (i.e. SPAN). Thus, the program works

exclusively with a wing which has a span of two units, and all output is for this normalized wing.

5.1 Wing Characteristics and Control Flags

The program title header and the input case title are printed for reference. Next the major wing planform characteristics and input control flags are printed. Since the program will automatically redefine some of the parameters and flags under certain run conditions, both the input and internally used values are printed for reference.

5.2 Fundamental Case Geometry

The page header is printed at the beginning of data for each fundamental case. For each section, the pertinent dimensions are given on the first line. Next the detailed data for the wing are given, with the general information printed on the first line, followed by the chordwise arrangement of all element locations, angles and EVD types. Each number printed across the page corresponds to one EVD element, beginning at the sectional leading edge and working toward the trailing edge. The data given include, for each element, the following:

a. Chordwise location, first normalized by the local chord, then in dimensional units.
b. Chordwise element length.
c. Total incidence angle resulting from all forms of fundamental case deflections.
d. Hinge turning angle of each element.
e. EVD type.

Next, the jet element data is given, if there is a jet at that section. The first line contains data applying to the whole jet section, and is followed by the chordwise arrangement of EVD element data. The element incidence angle is not printed since it has no physical meaning, and the hinge turning angles must be zero for all but the leading jet element. The length of the jet infinity EVD element will always be printed as a large number overflowing the format, since it is theoretically infinite and is automatically assigned the value 10^{10}.

The values of TYPE identify the following EVD types:

10	Regular EVD	
20	Leading Edge EVD	
30	Infinity EVD	
41	Leading edge flap hinge	Hinge EVD
42	Trailing edge flap hinge	
43	Trailing edge - jet-emersion	

For all fundamental cases after the first the printout of element dimensional locations and lengths are skipped, since they are the same for all cases. The normalized chordwise locations are also repetitious, but are printed for reference in each case.

5.3 Sectional Jet Strength

This output is given for reference and includes the c_μ value at every section, whether or not it is a jet section. This printout is skipped if JETFLG $\neq$ 0.

For dynamic stability derivative runs, an extra fundamental case is created automatically by the program, with incidence angles defined according to the quasi-steady induced camber due to a one degree/second pitching rate about the XCG location.

5.4 Fundamental Case Coefficients

After the problem solution is completed, the loading, ΔC_p, on each EVD element is printed for all fundamental cases. These data are given section by section, with the wing element loading first and the jet element loading second. In addition, detailed loading at small intervals are given for all leading edge and hinge EVD elements. This is intended primarily as an aid to the user in plotting the chordwise loading. The hinge detailed print is given only if IHINGE $\neq$ 0, since hinge loadings are otherwise represented by regular EVD elements rather than singular hinge EVD elements.

The spanwise loading data include, for each fundamental case, the lift,

induced drag, pitching moment, and center of lift for each wing section. At the bottom of each column, the values labeled "TOTAL" are the corresponding integrated aerodynamic coefficients for the whole wing. The sectional pitching moment coefficients are taken about the sectional leading edges, while the total coefficients are taken about both the wing apex (origin) and about the input XMC location. Sectional center of pressure and center of lift data are all normalized by the sectional chord. The total values indicate the chordwise centers of pressure and lift, each normalized by either reference chord or wing semispan. The total value of induced drag computed by the momentum method is given under the column headed ALFINF.

For dynamic stability derivative runs, the derivatives due to pitching are derived from the last fundamental case and are printed below the spanwise loading data.

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 include rolling and yawing moments. The most important coefficients are labeled with asterisks.

It should be noted that in order to compute the sectional pitching moment about the leading edge due to the thrust component (in the freestream direction) of jet reaction acting at the trailing edge, the vertical distance between the leading and trailing edges must be known. For wings with flap or slat fundamental cases, the program computes this distance using the hinge points and respective deflection angles. If, however, the same deflections are described by a camber fundamental case, the program has no way of knowing that the leading or trailing edges are displaced, and therefore, the jet thrust pitching moment contribution will always be zero. The user is thus warned to use care when interpreting the pitching moment results for jet-wings where flap or slat deflections are treated as camber alone.

5.5 Composite Case Coefficients

The composite case output is similar to that for fundamental cases, except that all data is given for each composite case before going on to the

next. The first line of data contains the input multiplication factors for the fundamental cases comprising this particular composite case. Then the chordwise loading is printed for each section. The Δc_p values are for zero angle of attack, labeled (A = 0), and for the variation of loading with angle of attack, labeled (A = 1). The fundamental case factors apply only to the loading at zero angle of attack, and the loading variation with angle of attack is taken directly from fundamental case number one.

As noted above, the CP value is infinite at the Leading Edge and Hinge points, and the values printed are the Leading Edge "average value" and the Regular EVD "peak" value underlying the singular Hinge EVD loading, if any. Thus, these points must not be plotted directly by the user. The Jet-Infinity EVD decays slowly from its "peak" value to zero at infinity downstream. The printed values may be used to compute any intermediate loading from the following equations:

Leading Edge Element:

$$\Delta c_{p_{XB}} = \frac{2}{3} \Delta c_{p_i} \left[\left(\frac{XB}{\bar{\delta}_i} \right)^{-\frac{1}{2}} - \left(\frac{XB}{\bar{\delta}_i} \right) \right] + \Delta c_{p_{i+1}} \left(\frac{XB}{\bar{\delta}_i} \right)$$

Hinge Element:

$$\Delta c_{p_{XB}} = -\frac{4_{\beta_i}}{\pi} \left[\log(c|XB-XH|) + \frac{\log\left(c\bar{\delta}_{i-1}\right)(XB-XH)}{\bar{\delta}_{i-1}} \right] + \left[\Delta c_{p_i} + \frac{\left(\Delta c_{p_i} - \Delta c_{p_{i-1}}\right)}{\bar{\delta}_{i-1}} (XB-XH) \right]$$

(for XB < XH)

$$\Delta c_{p_{XB}} = -\left[\frac{4_{\beta_i}}{\pi} \log(c|XB-XH|) - \frac{\log\left(c\bar{\delta}_i\right)(XB-XH)}{\bar{\delta}_i} \right] + \left[\Delta c_{p_i} - \frac{\left(\Delta c_{p_i} - \Delta c_{p_{i+1}}\right)}{\bar{\delta}_i} (XB-XH) \right]$$

(for XB < XH)

Jet-Infinity Element:

$$\Delta c_{p_{XB}} = \Delta c_{p_i} \, d^2 \left| c\,(XB-1) \right|^{-2}$$

where c is the local chord, $\bar{\delta}$ 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, Hinge or Jet-Infinity elements, respectively. XB, XH, and $\bar{\delta}$ are normalized by c.

The spanwise loading data are presented in generally the same way as for each fundamental case, except that there is more information. For the linear aerodynamic coefficients (lift, pitching moment, and center of lift), the sectional and total coefficients are given for both zero angle of attack (labels ending with 0) and their variation with angle of attack (labels ending with A). The first two lines of total lift center data are the chordwise locations normalized by reference chord, and in the third and fourth lines they are normalized by the wing semispan.

For the nonlinear induced drag coefficients, three terms are required: zero angle of attack (labels ending with 0), variation with angle of attack squared (labels ending with A2), and the "cross-product" variation with angle of attack (labels ending with X).

A summary table of all total aerodynamic coefficients is printed, with the data taken from the total data of the spanwise loading. Coefficients at zero angle of attack and those which are associated with α and α^2 are shown.

Finally, a summary table of the variations of total lift, pitching moment (about XMC), rolling moment, induced drag (momentum method) and yawing moment about XMC with angle of attack are given. In the sample case, the small negative drag values are not errors, but are simply an indication of inaccuracy resulting from the very coarse spacing used in the example.

It must be noted that in the above description of composite cases, angle of attack equal to zero refers to the basic configuration of the wing as constructed from the scale factors input for all fundamental cases. Since the α-case is a legitimate fundamental case and a multiplication factor may be input for Case 1, it is possible to have a composite case which is labeled "α=0" by the program, but which actually has an arbitrary constant flat plate

loading superimposed on it. The program does not distinguish an input for fundamental Case 1 from the other cases and, therefore, does not "know" that the wing is already at some angle of attack.

This feature may be used with care to obtain the composite loading at some angle of attack of particular interest, say α=4.3°, but the user should be aware that he has now forced the loading labels to be misleading. A label of α=0° now refers to the loading at α=4.3°, α=10° now means α=14.3°, etc. The user should check the scale factor for fundamental Case 1 printed at the top of the page in order to avoid misinterpretation of the composite case output. If it is 0.0, then all the labels are consistent with the printed data.

All of the output data described in Section 5.5 are repeated for each requested composite case in turn.

5.6 Stability Derivative Output

For most runs the program will repeat the above output for all requested composite cases. It will then attempt to read new CMU data and, if no CMU data is found, it will attempt to begin a completely new run. If no new run data is found the job is terminated. For dynamic stability derivative runs, however, the program will automatically begin a second run in which the derivatives due to rolling and yawing are computed. All previously input fundamental cases are altered to represent the effects induced by a yawing rate of one degree/second. Likewise, the extra fundamental case is altered to represent the effects induced by a rolling rate of one degree/second.

The fundamental case output for the second run consists simply of the rolling and yawing derivatives due to yawing (original fundamental cases) and due to rolling (extra fundamental case).

The output for the composite cases again includes the input fundamental case multiplication factors. As with the basic aerodynamic coefficients, the stability derivatives are generally composed of several angle of attack terms, but some of them are also dependent on the rates of yawing or rolling

themselves. The complete formulas for calculation are given in the printout.

Finally, a table of the variation of those derivatives which depend on angle of attack is given. Since only one CMU case can be treated, the program will attempt to begin a completely new run after the second stability derivative run. The termination message is printed, indicating the current run is completed, and if no further input data is given, the job is terminated.

6.0 OPERATIONAL CONSIDERATIONS

6.1 Program Versions Available

The EVD Jet-Wing Computer Program has been written entirely in Fortran IV, and could run on any large computer system. The program was developed on the IBM 360 and 370 systems at Douglas Aircraft Company and converted for use on CDC 6000 series computers. It is expected that only minor changes would be required for running on any other large-scale computing system.

6.2 Overlay Structure

Because of the large size of the program, the overlay feature of FORTRAN is required for operation on most machines. Shown below in Figure 9 is the overlay structure of the Mark II version. This structure requires approximately 141000 octal words of storage on the CDC 6500 and approximately 235000 bytes of storage on the IBM 360 computer. The overlay control cards for CDC 6000 series computers are included in the program listing, Appendix C.

6.3 Peripheral Storage Devices

The program requires the use of six peripheral storage devices, plus any devices required by the local system. These devices may be tape or disk units, and are described as follows:

Unit	Type	Use
1	Binary	Mass storage of downwash influence coefficients and matrices.
2	Binary	Matrix input to solution routine.
3	Binary	Temporary storage during matrix solution.
4	Binary	Temporary storage during matrix solution.
5	BCD	Standard input unit.
6	BCD	Standard printed output unit.

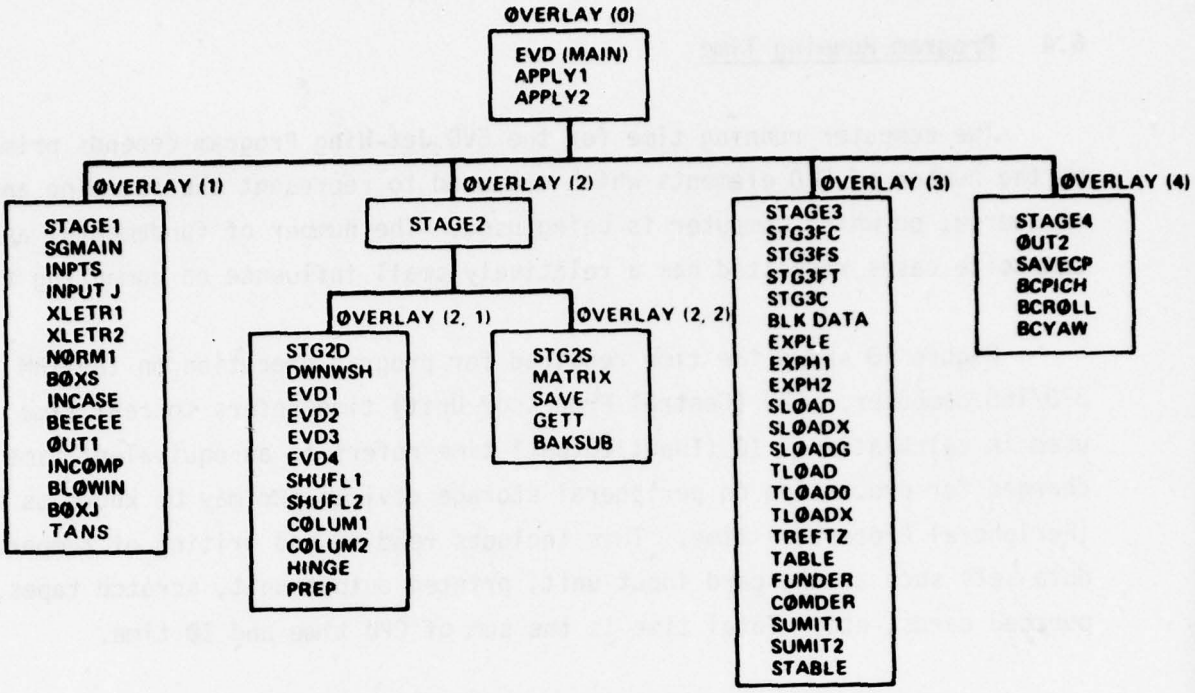


FIGURE 9. PROGRAM OVERLAY STRUCTURE

6.4 Program Running Time

The computer running time for the EVD Jet-Wing Program depends primarily on the number of EVD elements which are used to represent the jet-wing and, of course, on which computer is being used. The number of fundamental and composite cases requested has a relatively small influence on computing time.

Figure 10 shows the time required for program execution on the IBM 370/165 computer. CPU (Central Processor Unit) time refers to real time used in calculation. IØ (Input/Øutput) 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, printed output unit, scratch tapes, punched cards, etc. Total time is the sum of CPU time and IØ 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, IØ time is expected to vary from installation to installation, depending on the facilities available, the demand placed upon them, their cost, etc.

Figure 10 applies to the execution of one value of jet strength (which may be the no-jet case). For each additional jet strength case required in a single run, the total time will be increased by approximately half the total time of the first jet strength case.

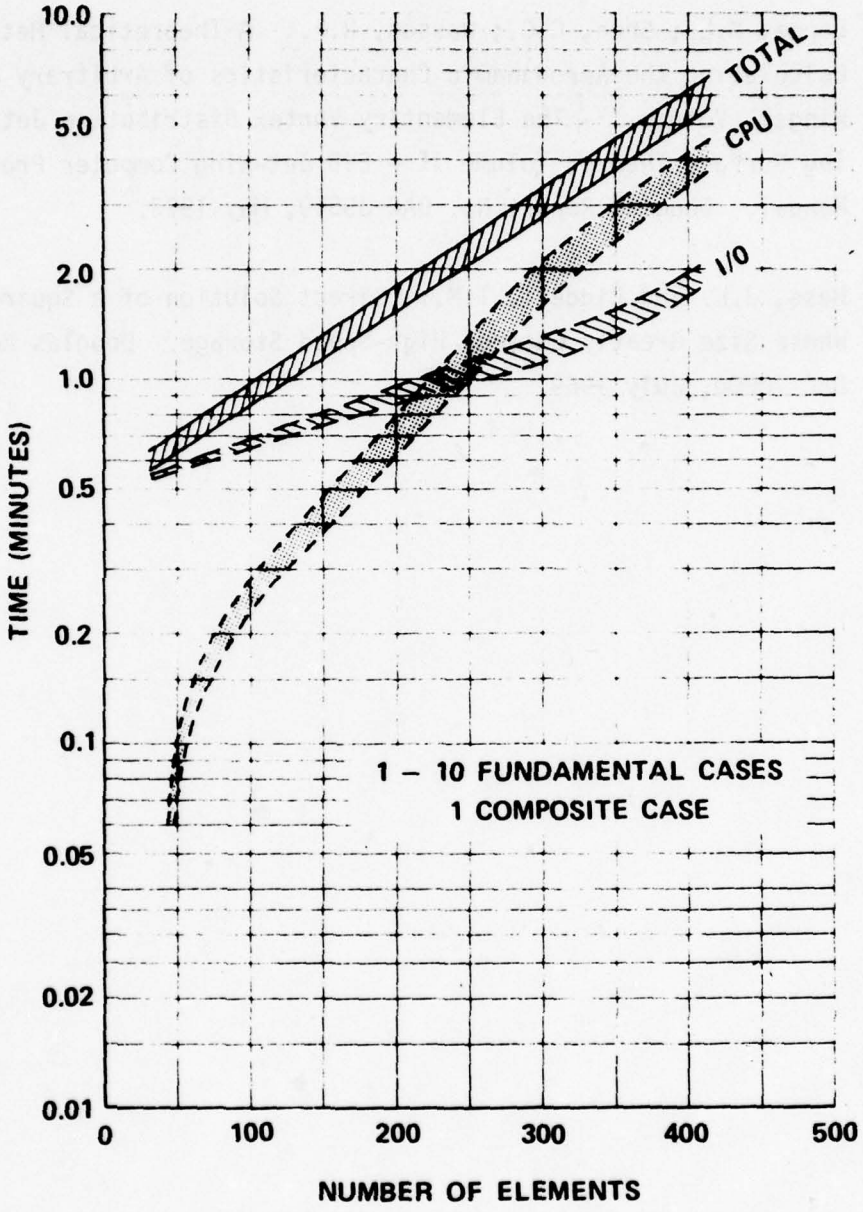


FIGURE 10.

COMPUTER PROGRAM OPERATIONAL TIME REQUIREMENTS

IBM 370/165

REFERENCES

1. Lopez, M.L.; Shen, C.C.; Wasson, N.F.: A Theoretical Method for Calculating the Aerodynamic Characteristics of Arbitrary Jet Flap Wings. Volume I - The Elementary Vortex Distribution Jet-Wing Lifting Surface Theory; Volume II - EVD Jet-Wing Computer Program User's Manual. Douglas Report No. DAC J5519, May 1973.

2. 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.

APPENDIX A - SAMPLE PROBLEM

Both the input card images and the program printout for a small sample case are given below. It must be noted that this case has been prepared only as a brief example of program input and output. The spacing is very rough, both spanwise and chordwise, and should not be used as a model for preparation of other cases. The fundamental and composite case capabilities of the program are far greater than those demonstrated by the sample case. Because of the crude spacing and few elements, it is expected that the results shown will not accurately predict the wing's aerodynamic characteristics.

INPUT CARD IMAGES

```
*** ONR SAMPLE CASE ***  RECTANGULAR WING    CMU = 1
                         WITH STABILITY DERIVATIVES
 4.500      4.500      1.000      0.250      0.250
 4 3 0 0 0 2 1 1
 0.9750     0.88750    0.68750    0.2750
 1 1 2 1
 5 6
 0.000      0.100      0.200      0.500      0.900
 0.000      0.100      0.200      0.500      0.800      0.900
 4.500      0.000      1.000
 1 1 1 1
 4
 1.000      1.100      1.500      3.000
0 0 1 0 0
 1.000      1.000      1.000      1.000
 0 0 0 0 1
 0 0 1 0
 0.9000   0 1.000
 1 0.00   2 10.00 3 10.00
9
 1.000      1.000      1.000      1.000
9
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

SAMPLE PROBLEM OUTPUT

```
**************************************
* EVD  JET - WING  COMPUTER  PROGRAM *
**************************************

*** ONR SAMPLE CASE ***  RECTANGULAR WING    CMU = 1    WITH STABILITY DERIVATIVES

                          USED          INPUT
          AREA =       0.888889      4.500000
          SPAN =       2.000000      4.500000
          CREF =       0.444444      1.000000
           XMC =       0.111111      0.250000
          CMAC =       0.444444      0.999999
       ARATIO =       4.500000      4.500000
           XCG =       0.111111      0.250000

         NROWS =  4          4
        NCASES =  3          3
         ISYMM =  0          0
        IPRINT =  0          0
        JETFLG =  0          0
        IGTYPE =  2          2
        IHINGE =  0          1

NUMBER OF WING ELEMENTS =  21
NUMBER OF JET ELEMENTS  =  16
TOTAL NUMBER OF ELEMENTS =  37

**************************************************************************
* ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  1 *
**************************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
            XB    0.0        0.100000   0.200000   0.500000   0.900000
            XI    0.0        0.044444   0.088889   0.222222   0.400000
           DEL    0.100000   0.100000   0.300000   0.400000   0.100000
           EPS    1.000000   1.000000   1.000000   1.000000   1.000000
          BETA    0.0        0.0        0.0        0.0        0.0
          TYPE    20         10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  1.000000
            XB    1.000000   1.099999   1.500000   3.000000
            XI    0.444444   0.488889   0.666667   1.333333
           DEL    0.099999   0.400001   1.500000**********
          BETA    0.0        0.0        0.0        0.0
          TYPE    10         10         10         30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
            XB    0.0        0.100000   0.200000   0.500000   0.900000
            XI    0.0        0.044444   0.088889   0.222222   0.400000
           DEL    0.100000   0.100000   0.300000   0.400000   0.100000
           EPS    1.000000   1.000000   1.000000   1.000000   1.000000
          BETA    0.0        0.0        0.0        0.0        0.0
          TYPE    20         10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  1.000000
            XB    1.000000   1.099999   1.500000   3.000000
            XI    0.444444   0.488889   0.666667   1.333333
           DEL    0.099999   0.400001   1.500000**********
          BETA    0.0        0.0        0.0        0.0
          TYPE    10         10         10         30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  6     TWIST =  0.0          HL =  0.0          THETA S =  0.0
            XB    0.0        0.100000   0.200000   0.500000   0.800000   0.900000
            XI    0.0        0.044444   0.088889   0.222222   0.355555   0.400000
           DEL    0.100000   0.100000   0.300000   0.300000   0.100000   0.100000
           EPS    1.000000   1.000000   1.000000   1.000000   1.000000   1.000000
          BETA    0.0        0.0        0.0        0.0        0.0        0.0
          TYPE    20         10         10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  1.000000
            XB    1.000000   1.099999   1.500000   3.000000
            XI    0.444444   0.488889   0.666667   1.333333
           DEL    0.099999   0.400001   1.500000**********
          BETA    0.0        0.0        0.0        0.0
          TYPE    10         10         10         30

*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0      XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS    NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
            XB    0.0        0.100000   0.200000   0.500000   0.900000
            XI    0.0        0.044444   0.088889   0.222222   0.400000
           DEL    0.100000   0.100000   0.300000   0.400000   0.100000
           EPS    1.000000   1.000000   1.000000   1.000000   1.000000
          BETA    0.0        0.0        0.0        0.0        0.0
          TYPE    20         10         10         10         10

 JET ELEMENTS    NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  1.000000
            XB    1.000000   1.099999   1.500000   3.000000
            XI    0.444444   0.488889   0.666667   1.333333
           DEL    0.099999   0.400001   1.500000**********
          BETA    0.0        0.0        0.0        0.0
          TYPE    10         10         10         30
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
**********************************************************************
*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  2  *
**********************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  1.000000     ACTE =  0.0          THETA =  1.000000
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    1.000000    0.0         0.0         0.0
        TYPE    43          10          10          30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  1.000000     ACTE =  0.0          THETA =  1.000000
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    1.000000    0.0         0.0         0.0
        TYPE    43          10          10          30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  6     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.800000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  1.000000     ACTE =  0.0          THETA =  1.000000
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    1.000000    0.0         0.0         0.0
        TYPE    43          10          10          30

*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  1.000000     ACTE =  0.0          THETA =  1.000000
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    1.000000    0.0         0.0         0.0
        TYPE    43          10          10          30


**********************************************************************
*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE  3  *
**********************************************************************

*** SECTION  1 ***  Y =  0.975000  DELTA =  0.025000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  0.0
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    0.0         0.0         0.0         0.0
        TYPE    43          10          10          30

*** SECTION  2 ***  Y =  0.887500  DELTA =  0.062500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  0.0
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    0.0         0.0         0.0         0.0
        TYPE    43          10          10          30

*** SECTION  3 ***  Y =  0.687500  DELTA =  0.137500  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  6     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.800000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0         1.000000
        BETA    0.0         0.0         0.0         0.0         0.0         1.000000
        TYPE    20          10          10          10          10          42

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  1.000000
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    0.0         0.0         0.0         0.0
        TYPE    43          10          10          30

*** SECTION  4 ***  Y =  0.275000  DELTA =  0.275000  XLEAD =  0.0       XTRAIL =  0.444444  CHORD =  0.444444  TANLE =  0.0

WING ELEMENTS     NW =  5     TWIST =  0.0          HL =  0.0          THETA S =  0.0
          XB    0.0         0.100000    0.200000    0.500000    0.900000
         EPS    0.0         0.0         0.0         0.0         0.0
        BETA    0.0         0.0         0.0         0.0         0.0
        TYPE    20          10          10          10          10

 JET ELEMENTS     NJ =  4     D =  0.888889     DJ =  0.0          ACTE =  0.0          THETA =  0.0
          XB    1.000000    1.099999    1.500000    3.000000
        BETA    0.0         0.0         0.0         0.0
        TYPE    43          10          10          30
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

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

ROW	CMU
1	1.000000
2	1.000000
3	1.000000
4	1.000000

****
* ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR FUNDAMENTAL CASE 4 *
****

*** SECTION 1 *** Y = 0.975000 DELTA = 0.025000 XLEAD = 0.0 XTRAIL = 0.444444 CHORD = 0.444444 TANLE = 0.0

WING ELEMENTS	NW = 5	TWIST = 0.0		HL = 0.0	THETA S = 0.0	
XB	0.0	0.100000	0.200000	0.500000	0.900000	
EPS	-0.400000	-0.200000	0.199999	0.900000	1.400000	
BETA	0.0	0.0	0.0	0.0	0.0	
TYPE	20	10	10	10	10	

JET ELEMENTS	NJ = 4	D = 0.888889	DJ = 0.0	ACTE = 0.0	THETA = 1.500000
XB	1.000000	1.099999	1.500000	3.000000	
BETA	0.0	0.0	0.0	0.0	
TYPE	43	10	10	30	

*** SECTION 2 *** Y = 0.887500 DELTA = 0.062500 XLEAD = 0.0 XTRAIL = 0.444444 CHORD = 0.444444 TANLE = 0.0

WING ELEMENTS	NW = 5	TWIST = 0.0		HL = 0.0	THETA S = 0.0	
XB	0.0	0.100000	0.200000	0.500000	0.900000	
EPS	-0.400000	-0.200000	0.199999	0.900000	1.400000	
BETA	0.0	0.0	0.0	0.0	0.0	
TYPE	20	10	10	10	10	

JET ELEMENTS	NJ = 4	D = 0.888889	DJ = 0.0	ACTE = 0.0	THETA = 1.500000
XB	1.000000	1.099999	1.500000	3.000000	
BETA	0.0	0.0	0.0	0.0	
TYPE	43	10	10	30	

*** SECTION 3 *** Y = 0.687500 DELTA = 0.137500 XLEAD = 0.0 XTRAIL = 0.444444 CHORD = 0.444444 TANLE = 0.0

WING ELEMENTS	NW = 6	TWIST = 0.0		HL = 0.0	THETA S = 0.0	
XB	0.0	0.100000	0.200000	0.500000	0.800000	0.900000
EPS	-0.400000	-0.200000	0.199999	0.800000	1.199999	1.400000
BETA	0.0	0.0	0.0	0.0	0.0	0.0
TYPE	20	10	10	10	10	42

JET ELEMENTS	NJ = 4	D = 0.888889	DJ = 0.0	ACTE = 0.0	THETA = 1.500000
XB	1.000000	1.099999	1.500000	3.000000	
BETA	0.0	0.0	0.0	0.0	
TYPE	43	10	10	30	

*** SECTION 4 *** Y = 0.275000 DELTA = 0.275000 XLEAD = 0.0 XTRAIL = 0.444444 CHORD = 0.444444 TANLE = 0.0

WING ELEMENTS	NW = 5	TWIST = 0.0		HL = 0.0	THETA S = 0.0	
XB	0.0	0.100000	0.200000	0.500000	0.900000	
EPS	-0.400000	-0.200000	0.199999	0.900000	1.400000	
BETA	0.0	0.0	0.0	0.0	0.0	
TYPE	20	10	10	10	10	

JET ELEMENTS	NJ = 4	D = 0.888889	DJ = 0.0	ACTE = 0.0	THETA = 1.500000
XB	1.000000	1.099999	1.500000	3.000000	
BETA	0.0	0.0	0.0	0.0	
TYPE	43	10	10	30	

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 1 Y = 0.975000 CHORD = 0.444444

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	0.129570	0.015231	0.006761	0.028993	0.0	0.0	0.0	0.0	0.0	0.0
	2	0.100000	0.062480	0.009628	0.004529	0.041840	0.0	0.0	0.0	0.0	0.0	0.0
	3	0.200000	0.036112	0.007270	0.003624	0.042261	0.0	0.0	0.0	0.0	0.0	0.0
	4	0.500000	0.016367	0.006730	0.004280	0.044040	0.0	0.0	0.0	0.0	0.0	0.0
	5	0.900000	0.007476	0.016049	0.005106	0.033779	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	22	1.000000	0.005186	0.058976	0.003649	0.019742	0.0	0.0	0.0	0.0	0.0	0.0
	23	1.099999	0.003300	0.009183	0.002402	0.008359	0.0	0.0	0.0	0.0	0.0	0.0
	24	1.500000	0.001203	0.001247	0.000538	0.002077	0.0	0.0	0.0	0.0	0.0	0.0
	25	3.000000	0.000202	0.000111	0.000026	0.000261	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.168371	0.022600	0.010083	0.047722
2	0.040000	0.127018	0.015845	0.007136	0.039566
3	0.060000	0.097176	0.012793	0.005832	0.038460
4	0.080000	0.077456	0.010931	0.005057	0.039619
5	0.100000	0.062480	0.009628	0.004529	0.041840

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 2 Y = 0.887500 CHORD = 0.444444

WING	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
	6	0.0	0.185800	0.024182	0.010750	0.056734	0.0	0.0	0.0	0.0	0.0	0.0
	7	0.100000	0.107434	0.017094	0.008019	0.068596	0.0	0.0	0.0	0.0	0.0	0.0
	8	0.200000	0.064708	0.013036	0.006499	0.067874	0.0	0.0	0.0	0.0	0.0	0.0
	9	0.500000	0.029236	0.012062	0.007938	0.070135	0.0	0.0	0.0	0.0	0.0	0.0
	10	0.900000	0.012979	0.027342	0.011413	0.051704	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	26	1.000000	0.008876	0.071214	0.008014	0.030824	0.0	0.0	0.0	0.0	0.0	0.0
	27	1.099999	0.005497	0.015626	0.004878	0.014111	0.0	0.0	0.0	0.0	0.0	0.0
	28	1.500000	0.001876	0.001991	0.000959	0.003273	0.0	0.0	0.0	0.0	0.0	0.0
	29	3.000000	0.000287	0.000155	0.000032	0.000369	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.273687	0.036243	0.016196	0.090728
2	0.040000	0.189277	0.025879	0.011673	0.072112
3	0.060000	0.150051	0.021396	0.009764	0.067293
4	0.080000	0.125341	0.018803	0.008694	0.066906
5	0.100000	0.107434	0.017094	0.008019	0.068596

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 3 Y = 0.687500 CHORD = 0.444444

WING	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
	11	0.0	0.230761	0.034686	0.014448	0.091838	0.0	0.0	0.0	0.0	0.0	0.0
	12	0.100000	0.140899	0.025102	0.011127	0.095908	0.0	0.0	0.0	0.0	0.0	0.0
	13	0.200000	0.089584	0.019423	0.009238	0.090418	0.0	0.0	0.0	0.0	0.0	0.0
	14	0.500000	0.044098	0.018778	0.012470	0.089254	0.0	0.0	0.0	0.0	0.0	0.0
	15	0.800000	0.025147	0.027830	0.032386	0.073930	0.0	0.0	0.0	0.0	0.0	0.0
	16	0.900000	0.020025	0.035635	0.085626	0.062237	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	30	1.000000	0.013414	0.076381	0.026293	0.039217	0.0	0.0	0.0	0.0	0.0	0.0
	31	1.099999	0.008238	0.019171	0.011354	0.019388	0.0	0.0	0.0	0.0	0.0	0.0
	32	1.500000	0.002674	0.002923	0.001175	0.004705	0.0	0.0	0.0	0.0	0.0	0.0
	33	3.000000	0.000348	0.000179	0.000040	0.000442	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.341409	0.052102	0.021837	0.143841
2	0.040000	0.238066	0.037353	0.015827	0.110679
3	0.060000	0.190841	0.031040	0.013332	0.099851
4	0.080000	0.161645	0.027436	0.011865	0.096198
5	0.100000	0.140898	0.025102	0.011127	0.095908

* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES *

SECTION 4 Y = 0.275000 CHORD = 0.444444

WING	I	XB	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
	17	0.0	0.261299	0.044340	0.011430	0.121582	0.0	0.0	0.0	0.0	0.0	0.0
	18	0.100000	0.152285	0.032007	0.007930	0.117146	0.0	0.0	0.0	0.0	0.0	0.0
	19	0.200000	0.105206	0.024658	0.005808	0.106311	0.0	0.0	0.0	0.0	0.0	0.0
	20	0.500000	0.054223	0.022370	0.004253	0.101691	0.0	0.0	0.0	0.0	0.0	0.0
	21	0.900000	0.025307	0.036868	0.002913	0.071827	0.0	0.0	0.0	0.0	0.0	0.0
JET												
	34	1.000000	0.017243	0.078280	0.002095	0.044616	0.0	0.0	0.0	0.0	0.0	0.0
	35	1.099999	0.010685	0.020564	0.001440	0.022860	0.0	0.0	0.0	0.0	0.0	0.0
	36	1.500000	0.003524	0.003578	0.000644	0.005991	0.0	0.0	0.0	0.0	0.0	0.0
	37	3.000000	0.000403	0.000220	0.000081	0.000522	0.0	0.0	0.0	0.0	0.0	0.0

DETAILED LEADING EDGE LOADING

1	0.020000	0.387138	0.066587	0.017101	0.188462
2	0.040000	0.270667	0.047717	0.012172	0.142565
3	0.060000	0.217741	0.039630	0.010023	0.126296
4	0.080000	0.185220	0.035007	0.008767	0.119495
5	0.100000	0.162285	0.032007	0.007930	0.117146

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

SPANWISE LOADING FOR FUNDAMENTAL CASE 1

SECTION	Y	LIFT CLG	LIFT CLMU	LIFT CL	*	INDUCED DRAG CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
1	0.975000	0.034284	0.017453	0.051737	*	0.0005984	0.0001523	0.0002930	0.0004577	1.0000000	0.0081921	0.0427911
2	0.887500	0.050186	0.017453	0.073639	*	0.0009806	0.0001523	0.0006025	0.0005304	1.0000000	0.0133976	0.0218746
3	0.687500	0.076014	0.017453	0.093468	*	0.0013267	0.0001523	0.0009294	0.0005496	1.0000000	0.0181988	0.0120300
4	0.275000	0.089566	0.017453	0.107020	*	0.0015632	0.0001523	0.0011917	0.0005239	1.0000000	0.0215896	0.0065474
	TOTAL	0.078903	0.017453	0.096356	*	0.0013771	0.0001523	0.0010010	0.0005285	0.9999996		0.0005033

SECTION	Y	PITCHING MOMENT CMG	CMMU	CMT	CM	* *	LIFT CENTER XCP/C	XCL/C	
1	0.975000	-0.007645	-0.017453	0.017453	-0.007645	* *	0.222985	0.485106	
2	0.887500	-0.013398	-0.017453	0.017453	-0.013388	* *	0.238284	0.418818	
3	0.687500	-0.019396	-0.017453	0.017453	-0.019396	* *	0.255167	0.394250	
4	0.275000	-0.023761	-0.017453	0.017453	-0.023761	* *	0.265285	0.385106	
	TOTAL	-0.020458	-0.017453	0.017453	-0.020458	(APEX)	0.259282	0.393450	(X/CREF)
		-0.000732	-0.013090	0.013090	-0.000732	(XMC)	0.115236	0.174867	(X/B/2)

SPANWISE LOADING FOR FUNDAMENTAL CASE 2

SECTION	Y	LIFT CLG	LIFT CLMU	LIFT CL	*	INDUCED DRAG CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
1	0.975000	0.013256	0.017453	0.030710	*	0.0	0.0001523	0.0000040	0.0001483	1.0000000	0.0044180	0.0254144
2	0.887500	0.021353	0.017453	0.038806	*	0.0	0.0001523	0.0000102	0.0001421	1.0000000	0.0068849	0.0122251
3	0.687500	0.028446	0.017453	0.045899	*	0.0	0.0001523	0.0000210	0.0001313	1.0000000	0.0089218	0.0057449
4	0.275000	0.033527	0.017453	0.050980	*	0.0	0.0001523	0.0000343	0.0001180	1.0000000	0.0103034	0.0022070
	TOTAL	0.029594	0.017453	0.047047	*	0.0	0.0001523	0.0000261	0.0001262	0.9999996		0.0001164

SECTION	Y	PITCHING MOMENT CMG	CMMU	CMT	CM	* *	LIFT CENTER XCP/C	XCL/C	
1	0.975000	-0.007630	-0.017453	0.0	-0.025283	* *	0.590655	0.823298	
2	0.887500	-0.012071	-0.017453	0.0	-0.029524	* *	0.565305	0.760812	
3	0.687500	-0.015144	-0.017453	0.0	-0.032597	* *	0.532393	0.710203	
4	0.275000	-0.017048	-0.017453	0.0	-0.034498	* *	0.508389	0.676695	
	TOTAL	-0.015460	-0.017453	0.0	-0.032893	(APEX)	0.521710	0.699142	(X/CREF)
		-0.008041	-0.013090	0.0	-0.021131	(XMC)	0.231871	0.310730	(X/B/2)

SPANWISE LOADING FOR FUNDAMENTAL CASE 3

SECTION	Y	LIFT CLG	LIFT CLMU	LIFT CL	*	INDUCED DRAG CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
1	0.975000	0.004811	0.0	0.004811	*	0.0	0.0	0.0000008	-0.0000008	1.0000000	0.0013792	0.0050237
2	0.887500	0.009230	0.0	0.009230	*	0.0	0.0	0.0000020	-0.0000020	1.0000000	0.0026257	0.0014837
3	0.687500	0.026510	0.017453	0.043964	*	0.0000977	0.0001523	0.0000036	0.0002463	1.0000000	0.0066333	0.0108795
4	0.275000	0.005419	0.0	0.005419	*	0.0	0.0	0.0000023	-0.0000023	1.0000000	0.0014751	0.0095400
	TOTAL	0.011115	0.004800	0.015915	*	0.0000269	0.0000419	0.0000025	0.0000662	0.9999996		0.0000785

SECTION	Y	PITCHING MOMENT CMG	CMMU	CMT	CM	* *	LIFT CENTER XCP/C	XCL/C	
1	0.975000	-0.002248	0.0	0.0	-0.002248	* *	0.467185	0.467185	
2	0.887500	-0.004608	0.0	0.0	-0.004608	* *	0.499208	0.499208	
3	0.687500	-0.016228	-0.017453	0.001745	-0.031936	* *	0.662069	0.802619	
4	0.275000	-0.001887	0.0	0.0	-0.001887	* *	0.348260	0.348260	
	TOTAL	-0.006189	-0.004800	0.000480	-0.010509	(APEX)	0.556798	0.690460	(X/CREF)
		-0.003410	-0.003600	0.000480	-0.006530	(XMC)	0.247466	0.306871	(X/B/2)

SPANWISE LOADING FOR FUNDAMENTAL CASE 4

SECTION	Y	LIFT CLG	LIFT CLMU	LIFT CL	*	INDUCED DRAG CDG	CDMU	CS	CD	CMU	GAMMA	ALFIN
1	0.975000	0.040381	0.026180	0.066561	*	0.0003055	0.0003427	0.0000147	0.0006335	1.0000000	0.0101972	0.0550330
2	0.887500	0.065122	0.026180	0.091302	*	0.0004685	0.0003427	0.0000552	0.0007550	1.0000000	0.0164325	0.0277257
3	0.687500	0.086505	0.026180	0.112785	*	0.0005723	0.0003427	0.0001472	0.0007678	1.0000000	0.0219236	0.0144727
4	0.275000	0.100914	0.026180	0.127094	*	0.0006315	0.0003427	0.0002580	0.0007162	1.0000000	0.0256589	0.0069844
	TOTAL	0.089478	0.026180	0.115658	*	0.0005785	0.0003427	0.0001901	0.0007311	0.9999996		0.0007183

SECTION	Y	PITCHING MOMENT CMG	CMMU	CMT	CM	* *	LIFT CENTER XCP/C	XCL/C	
1	0.975000	-0.018607	-0.026180	0.0	-0.044787	* *	0.460776	0.672865	
2	0.887500	-0.029265	-0.026180	0.0	-0.055445	* *	0.449387	0.607270	
3	0.687500	-0.037602	-0.026180	0.0	-0.063782	* *	0.434182	0.565522	
4	0.275000	-0.042490	-0.026180	0.0	-0.068670	* *	0.421055	0.540310	
	TOTAL	-0.038299	-0.026180	0.0	-0.064479	(APEX)	0.428023	0.557493	(X/CREF)
		-0.015929	-0.019635	0.0	-0.035564	(XMC)	0.190232	0.247775	(X/B/2)

LIFT COEFFICIENT DERIVATIVE DUE TO PITCHING ABOUT XCG, CLQ = 0.089478
PITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN DUE TO PITCHING ABOUT XCG, CMQ = -0.038299
PITCHING MOMENT COEFF DERIVATIVE ABOUT XMC DUE TO PITCHING ABOUT XCG, CMQMC = -0.015929

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TOTAL AERODYNAMIC COEFFICIENTS

		CASE 1	CASE 2	CASE 3	CASE 4	CASE 5	CASE 6	CASE 7	CASE 8	CASE 9	CASE 10
	CCLG	0.0785028	0.0295941	0.0111152	0.0894784	0.0	0.0	0.0	0.0	0.0	0.0
	CCLJ	0.0174533	0.0174533	0.0047997	0.0261799	0.0	0.0	0.0	0.0	0.0	0.0
**	CCL	0.0663361	0.0470474	0.0159149	0.1156583	0.0	0.0	0.0	0.0	0.0	0.0
	CCDG	0.0013771	0.0	0.0000269	0.0005785	0.0	0.0	0.0	0.0	0.0	0.0
	CCDJ	0.0001523	0.0001523	0.0000419	0.0003427	0.0	0.0	0.0	0.0	0.0	0.0
	CCS	0.0010010	0.0000261	0.0000025	0.0001901	0.0	0.0	0.0	0.0	0.0	0.0
	CCD	0.0005285	0.0001262	0.0000662	0.0007311	0.0	0.0	0.0	0.0	0.0	0.0
**	CDITZ	0.0005033	0.0001164	0.0000785	0.0007183	0.0	0.0	0.0	0.0	0.0	0.0
**	CCJ	0.9999996	0.9999996	0.9999996	0.9999996	0.0	0.0	0.0	0.0	0.0	0.0
	CCMG	-0.0204581	-0.0154396	-0.0061899	-0.0382688	0.0	0.0	0.0	0.0	0.0	0.0
	CCMJ	-0.0174533	-0.0174533	-0.0047997	-0.0261799	0.0	0.0	0.0	0.0	0.0	0.0
	CCMT	0.0174533	0.0	0.0004800	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	CCM	-0.0204581	-0.0328929	-0.0105086	-0.0644787	0.0	0.0	0.0	0.0	0.0	0.0
	CXCP	0.2592817	0.5217098	0.5567981	0.4280229	0.0	0.0	0.0	0.0	0.0	0.0
	CXCL	0.3934504	0.6991421	0.6904603	0.5574928	0.0	0.0	0.0	0.0	0.0	0.0
	CXCPB	0.1152363	0.2318709	0.2474658	0.1902324	0.0	0.0	0.0	0.0	0.0	0.0
	CXCLB	0.1748668	0.3107298	0.3068712	0.2477745	0.0	0.0	0.0	0.0	0.0	0.0
	CCMGMC	-0.0007324	-0.0080410	-0.0034101	-0.0159292	0.0	0.0	0.0	0.0	0.0	0.0
	CCMJMC	-0.0130900	-0.0130900	-0.0035997	-0.0196349	0.0	0.0	0.0	0.0	0.0	0.0
	CCMTMC	0.0130900	0.0	0.0004800	0.0	0.0	0.0	0.0	0.0	0.0	0.0
**	CCMMC	-0.0007324	-0.0211310	-0.0065299	-0.0355642	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
*	CNIMC	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
	CBGR	0.0358229	0.0136640	0.0067121	0.0403301	0.0	0.0	0.0	0.0	0.0	0.0
	CBGL	0.0358229	0.0136640	0.0067121	0.0403301	0.0	0.0	0.0	0.0	0.0	0.0
	CBJR	0.0087266	0.0087266	0.0032998	0.0130900	0.0	0.0	0.0	0.0	0.0	0.0
	CBJL	0.0087266	0.0087266	0.0032998	0.0130900	0.0	0.0	0.0	0.0	0.0	0.0
	CBR	0.0445495	0.0221906	0.0100119	0.0533200	0.0	0.0	0.0	0.0	0.0	0.0
	CBL	0.0445495	0.0221906	0.0100119	0.0533200	0.0	0.0	0.0	0.0	0.0	0.0
	CPMBP	0.4623423	0.4716647	0.4290900	0.4662008	0.0	0.0	0.0	0.0	0.0	0.0
	CPMBL	0.4623423	0.4716647	0.4290900	0.4662008	0.0	0.0	0.0	0.0	0.0	0.0

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

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	10.000000	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 = 0.975000 CHORD = 0.444444

WING					
XB	0.0	0.100000	0.200000	0.500000	0.900000
CP(A=0)	0.219927	0.141564	0.108941	0.110103	0.211553
CP(A=1)	0.129570	0.062480	0.036112	0.016367	0.007476
JET					
XB	1.000000	1.099999	1.500000	3.000000	
CP(A=0)	0.626244	0.115851	0.018450	0.001371	
CP(A=1)	0.005186	0.003300	0.001203	0.000202	

SECTION 2 Y = 0.887500 CHORD = 0.444444

WING					
XB	0.0	0.100000	0.200000	0.500000	0.900000
CP(A=0)	0.349325	0.251132	0.195345	0.200600	0.387549
CP(A=1)	0.185800	0.107434	0.064708	0.029236	0.012979
JET					
XB	1.000000	1.099999	1.500000	3.000000	
CP(A=0)	0.792272	0.205033	0.029491	0.001867	
CP(A=1)	0.008878	0.005497	0.001876	0.000287	

SECTION 3 Y = 0.687500 CHORD = 0.444444

WING						
XB	0.0	0.100000	0.200000	0.500000	0.800000	0.900000
CP(A=0)	0.491340	0.362289	0.287105	0.312486	0.602161	1.212608
CP(A=1)	0.230761	0.140898	0.089584	0.044098	0.025147	0.020025
JET						
XB	1.000000	1.099999	1.500000	3.000000		
CP(A=0)	1.026743	0.305252	0.040979	0.002192		
CP(A=1)	0.013414	0.008238	0.002674	0.000348		

SECTION 4 Y = 0.275000 CHORD = 0.444444

WING					
XB	0.0	0.100000	0.200000	0.500000	0.900000
CP(A=0)	0.557696	0.399370	0.304666	0.266234	0.397805
CP(A=1)	0.261299	0.162285	0.105206	0.054223	0.025307
JET					
XB	1.000000	1.099999	1.500000	3.000000	
CP(A=0)	0.303744	0.220041	0.042222	0.003008	
CP(A=1)	0.017243	0.010685	0.003524	0.000403	

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	10.000000	10.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

SECTION	Y	LIFT CLGO / CLGA	LIFT CLMUO / CLMUA	LIFT CLO / CLA	* *	PITCHING MOMENT CMGO / CMGA	PITCHING MOMENT CMMUO / CMMUA	PITCHING MOMENT CMTO / CMTA	PITCHING MOMENT CMO / CMA	* *	LIFT CENTER XCPO/C / XCPA/C	LIFT CENTER XCLO/C / XCLA/C
1	0.975000	0.180674	0.174533	0.355206	* *	-0.100776	-0.174533	0.0	-0.275309	* *	0.557779	0.775067
		0.034284	0.017453	0.051737	* *	-0.007645	-0.017453	0.017453	-0.007645	* *	0.222985	0.485106
2	0.887500	0.305825	0.174533	0.480358	* *	-0.166784	-0.174533	0.0	-0.341317	* *	0.545357	0.710547
		0.056186	0.017453	0.073639	* *	-0.013388	-0.017453	0.017453	-0.013388	* *	0.238284	0.418818
3	0.687500	0.529560	0.349066	0.878626	* *	-0.313718	-0.349066	0.017453	-0.645331	* *	0.592413	0.754342
		0.076014	0.017453	0.093468	* *	-0.019396	-0.017453	0.017453	-0.019396	* *	0.255167	0.394250
4	0.275000	0.389460	0.174533	0.563993	* *	-0.189319	-0.174533	0.0	-0.363852	* *	0.486108	0.645136
		0.083566	0.017453	0.107020	* *	-0.023761	-0.017453	0.017453	-0.023761	* *	0.265285	0.385106
	TOTAL	0.407093	0.222529	0.629623	* *	-0.216285	-0.222529	0.004800	-0.434014	(APEX)	0.531290	0.696948
		0.078903	0.017453	0.096356	* *	-0.020458	-0.017453	0.017453	-0.020458	(APEX)	0.259282	0.393450
					* *	-0.114511	-0.166837	0.004800	-0.276609	(XMC)	0.236129	0.309754
					* *	-0.000732	-0.013090	0.013090	-0.000732	(XMC)	0.115236	0.174867

SECTION	Y	INDUCED DRAG CDGO / CDGX / CDGA2	CDMUO / CDMUX / CDMUA2	CSO / CSX / CSA2	CDO / CDX / CDA2	GAMMAO / GAMMA2	ALFINO / ALFINA	CTO	CMU
1	0.975000	0.0	0.0152309	0.0008442	0.0143867	0.0579720	0.3043813	0.9856133	1.0000000
		0.0031533	0.0030462	0.0009947	0.0052048				
		0.0005984	0.0001523	0.0002930	0.0004577	0.0081921	0.0427911		
2	0.887500	0.0	0.0152309	0.0021298	0.0131011	0.0951066	0.1370882	0.9868990	1.0000000
		0.0053377	0.0030462	0.0022656	0.0061182				
		0.0009806	0.0001523	0.0006025	0.0005304	0.0133976	0.0218746		
3	0.687500	0.0195420	0.0609235	0.0042135	0.0762519	0.1555502	0.1662443	0.9237481	1.0000000
		0.0095344	0.0060923	0.0039578	0.0116690				
		0.0013267	0.0001523	0.0009294	0.0005496	0.0181988	0.0120300		
4	0.275000	0.0	0.0152309	0.0054284	0.0098025	0.1177849	0.1174700	0.9901975	1.0000000
		0.0067974	0.0030462	0.0050868	0.0047568				
		0.0015632	0.0001523	0.0011917	0.0005239	0.0215896	0.0065474		
	TOTAL	0.0053741	0.0277963	0.0044528	0.0287176		0.3451221	0.9712820	0.9999996
		0.0071854	0.0038839	0.0042190	0.0068502		0.0101239		
		0.0013771	0.0001523	0.0010010	0.0005285		0.0005033		

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TOTAL AERODYNAMIC COEFFICIENTS

		ALPHA=0	ALPHA	ALPHA**2
	CCLG	0.407093	0.078903	
	CCLJ	0.222529	0.017453	
**	CCL	0.629623	0.096356	
	CCDG	0.0053741	0.0071854	0.0013771
	CCDJ	0.0277963	0.0038839	0.0001523
	CCS	0.0044526	0.0042190	0.0010010
	CCD	0.0287176	0.0068502	0.0005285
**	CDITZ	0.0451221	0.0101239	0.0005033
**	CCJ	0.999996		
	CCMG	-0.216285	-0.020458	
	CCMJ	-0.222529	-0.017453	
	CCMT	0.004800	0.017453	
	CCM	-0.434014	-0.020458	
	CXCP	0.531290	0.259282	
	CXCL	0.696948	0.393450	
	CXCPB	0.236129	0.115236	
	CXCLB	0.309754	0.174867	
	CCMGMC	-0.114511	-0.000732	
	CCMJMC	-0.166897	-0.013090	
	CCMTMC	0.004800	0.013090	
**	CCMMC	-0.276609	-0.000732	
	CLLG	0.0	0.0	
	CLLJ	0.0	0.0	
*	CLL	0.0	0.0	
	CNJ	0.0		
*	CNTMC	0.0	0.0	0.0
*	CCY	0.0	0.0	0.0
	CBGR	0.201761	0.035823	
	CBGL	0.201761	0.035823	
	CBJR	0.120264	0.008727	
	CBJL	0.120264	0.008727	
	CBR	0.322025	0.044549	
	CBL	0.322025	0.044549	
	CPMBR	0.511457	0.462342	
	CPMBL	0.511457	0.462342	

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TABULATED TOTAL COEFFICIENTS FOR COMPOSITE CASE 1

ALPHA		CCL	CCL**2	CCM(MC)	CLL		CDITZ	CCT		CNT	CN	CCY
-10.000000	*	-0.3335302	0.1115147	-0.2692851	0.0	*	-0.0057847	1.0057840	*	0.0	0.0	0.0
-9.000000	*	-0.2375821	0.0564452	-0.2700175	0.0	*	-0.0052239	1.0052233	*	0.0	0.0	0.0
-8.000000	*	-0.1412280	0.0199448	-0.2707499	0.0	*	-0.0036565	1.0036554	*	0.0	0.0	0.0
-7.000000	*	-0.0448599	0.0020133	-0.2714822	0.0	*	-0.0010824	1.0010815	*	0.0	0.0	0.0
-6.000000	*	0.0514652	0.0026509	-0.2722145	0.0	*	0.0024982	0.9975014	*	0.0	0.0	0.0
-5.000000	*	0.1478423	0.0218573	-0.2729469	0.0	*	0.0070856	0.9929140	*	0.0	0.0	0.0
-4.000000	*	0.2441584	0.0596328	-0.2736793	0.0	*	0.0126796	0.9873200	*	0.0	0.0	0.0
-3.000000	*	0.3405545	0.1159773	-0.2744116	0.0	*	0.0192803	0.9807193	*	0.0	0.0	0.0
-2.000000	*	0.4369106	0.1908608	-0.2751440	0.0	*	0.0268876	0.9731120	*	0.0	0.0	0.0
-1.000000	*	0.5332667	0.2843733	-0.2758763	0.0	*	0.0355015	0.9644981	*	0.0	0.0	0.0
0.0	*	0.6296228	0.3964249	-0.2766087	0.0	*	0.0451221	0.9548775	*	0.0	0.0	0.0
1.000000	*	0.7259789	0.5270452	-0.2773410	0.0	*	0.0557494	0.9442502	*	0.0	0.0	0.0
2.000000	*	0.8223349	0.6762347	-0.2780734	0.0	*	0.0673832	0.9326164	*	0.0	0.0	0.0
3.000000	*	0.9186910	0.8439032	-0.2788057	0.0	*	0.0800238	0.9199758	*	0.0	0.0	0.0
4.000000	*	1.0150471	1.0303202	-0.2795381	0.0	*	0.0936710	0.9063286	*	0.0	0.0	0.0
5.000000	*	1.1114025	1.2352152	-0.2802705	0.0	*	0.1083248	0.8916748	*	0.0	0.0	0.0
6.000000	*	1.2077589	1.4586811	-0.2810028	0.0	*	0.1239854	0.8760143	*	0.0	0.0	0.0
7.000000	*	1.3041153	1.7007160	-0.2817352	0.0	*	0.1406525	0.8593472	*	0.0	0.0	0.0
8.000000	*	1.4004707	1.9613180	-0.2824675	0.0	*	0.1583263	0.8416733	*	0.0	0.0	0.0
9.000000	*	1.4968271	2.2404909	-0.2831998	0.0	*	0.1770068	0.8229929	*	0.0	0.0	0.0
10.000000	*	1.5931835	2.5382329	-0.2839322	0.0	*	0.1966938	0.8033058	*	0.0	0.0	0.0
11.000000	*	1.6895390	2.8545418	-0.2846646	0.0	*	0.2173876	0.7826120	*	0.0	0.0	0.0
12.000000	*	1.7858953	3.1894217	-0.2853969	0.0	*	0.2390880	0.7609116	*	0.0	0.0	0.0
13.000000	*	1.8822508	3.5428677	-0.2861293	0.0	*	0.2617950	0.7382046	*	0.0	0.0	0.0
14.000000	*	1.9786072	3.9148855	-0.2868617	0.0	*	0.2855088	0.7144909	*	0.0	0.0	0.0
15.000000	*	2.0749636	4.3054733	-0.2875940	0.0	*	0.3102291	0.6897705	*	0.0	0.0	0.0
16.000000	*	2.1713200	4.7146301	-0.2883264	0.0	*	0.3359562	0.6640435	*	0.0	0.0	0.0
17.000000	*	2.2676754	5.1423512	-0.2890587	0.0	*	0.3626897	0.6373099	*	0.0	0.0	0.0
18.000000	*	2.3640318	5.5886459	-0.2897910	0.0	*	0.3904300	0.6095696	*	0.0	0.0	0.0
19.000000	*	2.4603882	6.0535097	-0.2905234	0.0	*	0.4191770	0.5808226	*	0.0	0.0	0.0
20.000000	*	2.5567436	6.5369377	-0.2912558	0.0	*	0.4489306	0.5510691	*	0.0	0.0	0.0
21.000000	*	2.6531000	7.0389395	-0.2919881	0.0	*	0.4796908	0.5203089	*	0.0	0.0	0.0
22.000000	*	2.7494564	7.5595102	-0.2927205	0.0	*	0.5114577	0.4885419	*	0.0	0.0	0.0
23.000000	*	2.8458119	8.0986443	-0.2934529	0.0	*	0.5442312	0.4557684	*	0.0	0.0	0.0
24.000000	*	2.9421682	8.6563530	-0.2941852	0.0	*	0.5780115	0.4219882	*	0.0	0.0	0.0
25.000000	*	3.0385246	9.2326317	-0.2949175	0.0	*	0.6127983	0.3872014	*	0.0	0.0	0.0
26.000000	*	3.1348801	9.8274727	-0.2956499	0.0	*	0.6485918	0.3514079	*	0.0	0.0	0.0
27.000000	*	3.2312365	10.4408884	-0.2963822	0.0	*	0.6853919	0.3146077	*	0.0	0.0	0.0
28.000000	*	3.3275923	11.0729741	-0.2971146	0.0	*	0.7231987	0.2768009	*	0.0	0.0	0.0
29.000000	*	3.4239483	11.7236211	-0.2978470	0.0	*	0.7620121	0.2379875	*	0.0	0.0	0.0
30.000000	*	3.5203047	12.3929447	-0.2985793	0.0	*	0.8018323	0.1981674	*	0.0	0.0	0.0

****
* SECOND RUN FOR STABILITY DERIVATIVE CASE *
****

****
* STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE 1 *
****

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING, CLLR = 0.0001168

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS

CN(R) = CNR*R + CNR2*R**2

WHERE CNR = 0.000001049
CNR2 = 0.0

SIDE FORCE COEFFICIENT DUE TO YAWING, CY(R) MAY BE CALCULATED AS FOLLOWS

CY(R) = CYR*R + CYR2*R**2

WHERE CYR = 0.0
CYR2 = 0.0

****
* STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE 2 *
****

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING, CLLR = 0.0000436

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS

CN(R) = CNR*R + CNR2*R**2

WHERE CNR = 0.000000034
CNR2 = 0.0

SIDE FORCE COEFFICIENT DUE TO YAWING, CY(R) MAY BE CALCULATED AS FOLLOWS

CY(R) = CYR*R + CYR2*R**2

WHERE CYR = 0.0
CYR2 = 0.0

****
* STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE 3 *
****

ROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO YAWING, CLLR = 0.0000344

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS

CN(R) = CNR*R + CNR2*R**2

WHERE CNR = -0.000000100
CNR2 = 0.0

SIDE FORCE COEFFICIENT DUE TO YAWING, CY(R) MAY BE CALCULATED AS FOLLOWS

CY(P) = CYR*R + CYR2*R**2

WHERE CYR = 0.0
CYP2 = 0.0

****
* STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE 4 *
****

ROLLING MOMENT COEFF DERIVATIVE DUE TO ROLLING, CLLP = -0.0066949

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS

CN(P) = CNP2*P**2

WHERE CNP2 = 0.0

SIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P) MAY BE CALCULATED AS FOLLOWS

CY(P) = CYP2*P**2

WHERE CYP2 = 0.0

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

****
* STABILITY DERIVATIVE DATA 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	10.000000	10.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

LIFT COEFFICIENT DERIVATIVE DUE TO PITCHING ABOUT XCG, CLQ = 0.089478
PITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN DUE TO PITCHING ABOUT XCG, CMQ = -0.038299
PITCHING MOMENT COEFF DERIVATIVE ABOUT XMC DUE TO PITCHING ABOUT XCG, CMQMC = -0.015929

ROLLING MOMENT COEFF DERIVATIVE DUE TO ROLLING, CLLP = -0.0066949

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS
CN(P) = CNP*P + CNP2*P**2
WHERE CNP = CNPO + CNPA*ALPHA
CNPO = -0.0003531
CNPA = -0.0000601
CNP2 = 0.0

SIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P) MAY BE CALCULATED AS FOLLOWS
CY(P) = CYP*P + CYP2*P**2
WHERE CYP = CYPO + CYPA*ALPHA
CYPO = 0.0
CYPA = 0.0
CYP2 = 0.0

ROLLING MOMENT COEFF DERIVATIVE DUE TO YAWING ABOUT XCG, CLLR MAY BE CALCULATED AS FOLLOWS
CLLR = CLLRO + CLLRA*ALPHA
WHERE CLLRO = 0.0007805
CLLRA = 0.0001168

YAWING MOMENT COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS
CN(R) = CNR*R + CNR2*R**2
WHERE CNR = CNRO + CNRA*ALPHA + CNRA2*ALPHA**2
CNRO = -0.0000130
CNRA = -0.0000035
CNRA2 = 0.0000010
AND CNR2 = CNR2O + CNR2A*ALPHA + CNR2A2*ALPHA**2
CNR2O = 0.0
CNR2A = 0.0
CNR2A2 = 0.0

SIDE FORCE COEFFICIENT ABOUT XMC DUE TO YAWING ABOUT XCG, CY(R) MAY BE CALCULATED AS FOLLOWS
CY(R) = CYR*R + CYR2*R**2
WHERE CYR = CYRO + CYRA*ALPHA + CYRA2*ALPHA**2
CYRO = 0.0
CYRA = 0.0
CYRA2 = 0.0
AND CYR2 = CYR2O + CYR2A*ALPHA + CYR2A2*ALPHA**2
CYR2O = 0.0
CYR2A = 0.0
CYR2A2 = 0.0

****
* VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK *
****

ALPHA	CNP	CNP2	CYP	CYP2	CLLR	CNR	CNR2
-10.000000	0.0002479	0.0	0.0	0.0	-0.0003880	0.0001273	0.0
-9.000000	0.0001876	0.0	0.0	0.0	-0.0002712	0.0001038	0.0
-8.000000	0.0001277	0.0	0.0	0.0	-0.0001543	0.0000824	0.0
-7.000000	0.0000676	0.0	0.0	0.0	-0.0000375	0.0000632	0.0
-6.000000	0.0000075	0.0	0.0	0.0	0.0000794	0.0000460	0.0
-5.000000	-0.0000526	0.0	0.0	0.0	0.0001962	0.0000309	0.0
-4.000000	-0.0001127	0.0	0.0	0.0	0.0003131	0.0000179	0.0
-3.000000	-0.0001728	0.0	0.0	0.0	0.0004299	0.0000070	0.0
-2.000000	-0.0002329	0.0	0.0	0.0	0.0005468	-0.0000017	0.0
-1.000000	-0.0002930	0.0	0.0	0.0	0.0006636	-0.0000084	0.0
0.0	-0.0003531	0.0	0.0	0.0	0.0007805	-0.0000130	0.0
1.000000	-0.0004132	0.0	0.0	0.0	0.0008973	-0.0000155	0.0
2.000000	-0.0004733	0.0	0.0	0.0	0.0010142	-0.0000159	0.0
3.000000	-0.0005334	0.0	0.0	0.0	0.0011310	-0.0000142	0.0
4.000000	-0.0005935	0.0	0.0	0.0	0.0012479	-0.0000104	0.0
5.000000	-0.0006536	0.0	0.0	0.0	0.0013647	-0.0000045	0.0
6.000000	-0.0007137	0.0	0.0	0.0	0.0014815	0.0000035	0.0
7.000000	-0.0007738	0.0	0.0	0.0	0.0015984	0.0000136	0.0
8.000000	-0.0008339	0.0	0.0	0.0	0.0017152	0.0000258	0.0
9.000000	-0.0008940	0.0	0.0	0.0	0.0018321	0.0000401	0.0
10.000000	-0.0009541	0.0	0.0	0.0	0.0019489	0.0000565	0.0
11.000000	-0.0010142	0.0	0.0	0.0	0.0020658	0.0000750	0.0
12.000000	-0.0010743	0.0	0.0	0.0	0.0021826	0.0000956	0.0
13.000000	-0.0011344	0.0	0.0	0.0	0.0022995	0.0001182	0.0
14.000000	-0.0011945	0.0	0.0	0.0	0.0024163	0.0001430	0.0
15.000000	-0.0012546	0.0	0.0	0.0	0.0025332	0.0001699	0.0
16.000000	-0.0013147	0.0	0.0	0.0	0.0026500	0.0001989	0.0
17.000000	-0.0013748	0.0	0.0	0.0	0.0027669	0.0002299	0.0
18.000000	-0.0014349	0.0	0.0	0.0	0.0028837	0.0002631	0.0
19.000000	-0.0014950	0.0	0.0	0.0	0.0030006	0.0002984	0.0
20.000000	-0.0015551	0.0	0.0	0.0	0.0031174	0.0003358	0.0
21.000000	-0.0016152	0.0	0.0	0.0	0.0032343	0.0003752	0.0
22.000000	-0.0016753	0.0	0.0	0.0	0.0033511	0.0004168	0.0
23.000000	-0.0017354	0.0	0.0	0.0	0.0034680	0.0004605	0.0
24.000000	-0.0017955	0.0	0.0	0.0	0.0035848	0.0005062	0.0
25.000000	-0.0018556	0.0	0.0	0.0	0.0037017	0.0005541	0.0
26.000000	-0.0019157	0.0	0.0	0.0	0.0038185	0.0006040	0.0
27.000000	-0.0019758	0.0	0.0	0.0	0.0039354	0.0006561	0.0
28.000000	-0.0020359	0.0	0.0	0.0	0.0040522	0.0007102	0.0
29.000000	-0.0020960	0.0	0.0	0.0	0.0041690	0.0007665	0.0
30.000000	-0.0021561	0.0	0.0	0.0	0.0042859	0.0008248	0.0

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

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

APPENDIX B - 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
THE ISYM FLAG INDICATED AN ANTI-SYMETRIC CASE. HOWEVER, IT WILL BE TREATED AS SYMETRIC.	APPLY2
THE IGTYPE FLAG HAS THE VALUE OF** ONLY THE VALUES 1 OR 2 ARE ACCEPTABLE THIS CASE HAS BEEN TERMINATED	STAGE1
NROWS =***	STAGE1
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
J ROW CONTINUITY RULE FAILURE	INPUTJ
AN INCONSISTENCY HAS BEEN FOUND IN THE SECTIONAL LEADING AND TRAILING EDGE INPUT	XLETR1
PLEASE CHECK YOUR SECTION LOCATION (Y) INPUT	ROXS
**** IS TOO MANY ELEMENTS	ROXS
A ZERO VALUE OF CMU HAS BEEN INPUT. THIS CMU CASE HAS BEEN IGNORED.	ROXJ
FUNDAMENTAL GEOMETRIC CASE*** AN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT DATA FOR WING ROW***, RUN TYPE***	BEECEE
AN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN FOUND. IT WILL BE IGNORED.	INCOMP
MORE THAN 25 COMPOSITE CASES HAVE BEEN INPUT. SUBSEQUENT INPUTS WILL BE IGNORED.	INCOMP
NO END OF FILE HAS BEEN READ DURING COMPOSITE CASE INPUT	INCOMP
MATRIX DOES NOT HAVE ENOUGH CORE TO WORK THIS CASE HAS BEEN TERMINATED	STG2S

THIS PAGE IS BEST QUALITY PRACTICABLE FROM COPY FURNISHED TO DDC

```
*DECK MAIN
      OVERLAY(OVLY,2,2)
      PROGRAM EVDI(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1           TAPE1,TAPE2,TAPE3,TAPE4)
C
C*****************************************************************
C*****************************************************************
C
C
C     THE DEVELOPMENT OF THE ELEMENTARY VORTEX DISTRIBUTION (EVDI) JET-WING
C     COMPUTER PROGRAM FOR INTERMEDIATE CHARACTERISTICS OF
C     ARBITRARY JET FLAPPED WINGS WAS PREPARED UNDER THE SPONSORSHIP OF THE
C     DOUGLAS AIRCRAFT COMPANY GROUP - AERODYNAMICS RESEARCH.
C     THE INDEPENDENT DEVELOPMENT OF THE SPONSORSHIP OF
C     DOUGLAS, THE DOUGLAS EVDI JET INVELOPMENT PROGRAM OF MCDONNELL
C     THE SUBJECT OF EXTENSIONS AND IMPROVEMENTS. SURFACE THEORY HAS BEEN
C     IT COMPLETED UNDER OFFICE OF NAVAL RESEARCH CONTRACT NO. N6-ONR-271-025
C     COMPUTER PROGRAM WITH ENGINEERS, WORK LEADING TO THE PRESENT
C     DEVELOPED BY THE DOUGLAS AIRCRAFT CO., SEVERAL FEATURES ORIGINALLY
C     UNDER THE SPONSORSHIP OF THIS CONTRACT.
C
C     IN SUMMARY THE EVD JET WING COMPUTER PROGRAM WILL PROVIDE,
C     AND ARBITRARY PLANFORMS OF WING
C        1.  SPANWISE AND CHORDWISE LOADING
C        2.  SPANWISE VARIATION OF INDUCED DRAG
C        3.  A CAPABILITY TO INVESTIGATE THE EFFECTS OF -
C            A.  AFT SPAN FLAPS
C            B.  FULL SPAN FLAPS
C            C.  ROLLING, YAWING, PITCHING AND SIDESLIP
C        4.  TOTAL LIFT AND INDUCED DRAG (TREFFTZ PLANE METHOD),
C            PITCHING, YAWING AND ROLLING MOMENTS, ETC.
C
C     COMPLETE DOCUMENTATION OF THE EVD JET WING LIFTING SURFACE THEORY
C     AND ASSOCIATED COMPUTER PROGRAM ARE CONTAINED IN DOUGLAS REPORT
C
C          DSR13 -- A THEORETICAL METHOD FOR CALCULATING THE
C          AERODYNAMIC CHARACTERISTICS OF ARBITRARY JET FLAP WINGS
C
C                  VOLUME I
C              THE ELEMENTARY VORTEX DISTRIBUTION JET-WING
C                    LIFTING SURFACE THEORY
C
C                  VOLUME II
C              EVD JET-WING COMPUTER PROGRAM USERS MANUAL
C
C*****************************************************************
      COMMON/MATHEV/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IMINGE
      COMMON/MARK/NPONJS,NWT,NJT,NWEY,NWING(4),NJET(4),IW(4,1),IJ(4,1)
      COMMON/ZLIKE/ TITLE(80)
      COMMON/SPOTIT/ ARSPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG
      COMMON/INDATA/ SPAN,CHORD,NOALFA,LOGIC,IB
      COMMON/GEOM1/Y(140),CHORD(140),CELTA(140),XLE(140),ZLE(140),
     1 DIH(140),KK(140),TYPE(140),NC(140),NT(140),IGT,IHI
      COMMON/GEOM2/ZETA(140),TANLE(140),TANTE(140)
      COMMON/GEOM3/ XMR(140),BETA(140),ROT(140),ACE(140),
      COMMON/CASEX/POSITC(10),BETA2(10),XHF(140),NET,NHT
      COMMON/CASEY/CMUP(40),CMUPP(40)
      COMMON/CONDOR/ZETOT(741),NCC
      COMMON/DERIV/UC(40),CLO,CMOMC
      COMMON/ZENOTT/INDEX(100)
      DATA CHECK/1040
C
      CALL OPENMS(1,INDEX,2000,0)
C
C     READ THE TITLE FOR THIS CASE
   10 READ(5,1) TITLE
      IF(EOF(5)) 1000,20
C
C     READ GENERAL GEOMETRY CONTROL DATA
   20 READ(5,2) NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IMINGE,INDEX
      READ(5,3) NPONJS,NWT,NJT,NWEY,NWING,NJET,IW,IJ
    1 FORMAT(10A8)
    2 FORMAT(8I10)
    3 FORMAT(40I2)
      NC = NCASES
      ISY = ISYMM
      IPR = IPRINT
      JET = JETFLG
      IHI = IMINGE
C
C     FIND OUT WHICH TYPE OF RUN IS REQUIRED
      IF(IMINGE .NE. 0) GO TO 70
C
C     A REGULAR RUN WILL BE EXECUTED
      CALL APPLY1(
      GO TO ( 70 , 80 , 110 , 130 ), IB
C
C     A STABILITY DERIVATIVE RUN WILL BE EXECUTED
   70 CALL APPLY2(
      IF(IB .EQ. 2) GO TO 130
C
C*****************************************************************
C
C     PRINT COMPLETION MESSAGE FOR THIS RUN AND GO BACK TO BEGIN A NEW RUN
   80 WRITE(6, 90 )
   90 FORMAT(1H0///,' 33X,16(5H*****), 3H***/ 33X,
     2       32H* THE PROGRAM HAS REACHED NORMAL TERMINATION */
     3       32H* AT THE END OF THE NEXT CARD IS A TITLE OR AN OLD END OF CMU CARD
     4       32H* IF IT IS A TITLE, GO TO 10
     5       32H* IF IT IS AN OLD END OF CMU CARD GO TO 10
     6       32H* GO TO 10                                  */
     7       33X,16(5H*****), 3H***)
C
C     PRINT COMPLETION MESSAGE AND STOP EXECUTION
  110 WRITE(6, 120 )
  120 STOP
C
C*****************************************************************
C     A FATAL ERROR HAS OCCURED. PRINT FINAL MESSAGE AND STOP EXECUTION.
  130 WRITE(6,140)
  140 FORMAT(1H1///,' 33X,16(5H*****)/
     1       33X,1H*,14X,'PROGRAM HAS REACHED ABNORMAL TERMINATION */
     2       33X,16(5H*****))
      STOP
      END
```

```
*DECK APPLY1
      SUBROUTINE APPLY1
C
C     THIS SUBROUTINE CONTROLS ALL ASPECTS OF CALCULATION OF REGULAR CASES
C
      COMMON/MATHEV/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IMINGE
      COMMON/SPOTIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IB
C     DECIDE WHETHER OR NOT THERE IS AN ALPHA CASE
      IB = 1
      LOGIC = 1
      IF(ISYMM .LT. 0) NOALFA = 0
C
C     INITIALIZE AND INCREMENT THE CMU CASE CONTROL COUNTER
   10 NEWCMU = NEWCMU + 1
   20 NEWCMU = NEWCMU + 1
C
C     EXECUTE THE PROBLEM FORMATION STAGE
   30 CALL OVERLAY(4HOVLY,2,1)
      GO TO ( 40 , 40 , 70 , 80 ), IR
C
C     EXECUTE THE PROBLEM SOLUTION STAGE
   40 CALL OVERLAY(4HOVLY,2,2)
      IF(IR .EQ. 2) GO TO 80
C
C     EXECUTE THE AERODYNAMIC PARAMETER STAGE
   60 CALL OVERLAY(4HOVLY,3,0)
C
C     THE PROGRAM HAS BEEN EXECUTED SUCCESSFULLY
C     GO BACK AND DO A NEW CMU CASE
      IF(JETFLG .NE. 0) GO TO 20
      GO TO 80
C
C*****************************************************************
C     THIS RUN HAS BEEN COMPLETED. RETURN TO START A NEW RUN.
   70 IR = 2
      GO TO 90
C     THIS RUN HAS BEEN COMPLETED. NO FURTHER RUNS FOLLOW.
   80 IR = 3
      RETURN
C     A FATAL ERROR HAS OCCURED. RETURN AND QUIT.
   90 IR = 4
      RETURN
      END
```

C1

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK APPLY2
      SUBROUTINE APPLY2                                                 APL20010
C                                                                       APL20020
C  THIS SUBROUTINE CONTROLS ALL ASPECTS OF CALCULATION OF               APL20030
C  STABILITY DERIVATIVES                                                APL20040
C                                                                       APL20050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETELG,IGTYPE,IHINGE            APL20060
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALEA,LOGIC,IR                      APL20070
C                                                                       APL20080
C  CHECK ON STATUS OF CONTROL FLAGS                                     APL20090
   10 IHINGE = 0                                                        APL20100
      NOALEA =1                                                         APL20110
      NEWCMU = 1                                                        APL20120
      IF(ISYMM .GE. 0) GO TO 30                                         APL20130
      ISYMM = 0                                                         APL20140
      WRITE(6, 20 )                                                     APL20150
   20 FORMAT(1H0///14X,61HTHE ISYMM FLAG INDICATED AN ANTI-SYMETRIC.    APL20160
     1    48H CASE.  HOWEVER, IT WILL BE TREATED AS SYMETRIC.)          APL20170
C                                                                       APL20180
C  EXECUTE THE FIRST RUN                                                APL20190
C                                                                       APL20200
C  FORMULATE THE PROBLEM AS USUAL                                       APL20210
   30 CALL OVERLAY(4HOVLY,1,-1)                                         APL20220
      GO TO ( 40 , 110 , 120 , 110 ), IR                                APL20230
C                                                                       APL20240
C  ADD THE EXTRA FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING       APL20250
   40 LOGIC = 1                                                         APL20260
      CALL OVERLAY(4HOVLY,4,-1)                                         APL20270
C                                                                       APL20280
C  EXECUTE THE PROBLEM SOLUTION STAGE AS USUAL FOR THE FIRST RUN        APL20290
      LOGIC = 1                                                         APL20300
   50 CALL OVERLAY(4HOVLY,2,-1)                                         APL20310
      IF(IR .EQ. 3) GO TO 110                                           APL20320
C                                                                       APL20330
C  EXECUTE THE AERODYNAMIC STAGE FOR THE FIRST RUN                      APL20340
C  FUNDAMENTAL CASES                                                    APL20350
   60 LOGIC = 2                                                         APL20360
      CALL OVERLAY(4HOVLY,3,-1)                                         APL20370
C                                                                       APL20380
C  THE FIRST RUN HAS BEEN COMPLETED                                     APL20390
C  EXECUTE THE SECOND RUN                                               APL20400
C                                                                       APL20410
      WRITE(6, 70 )                                                     APL20420
   70 FORMAT(1H1////////////// 37X,11(4H****),2H* /                     APL20430
     1       37X,6HR  SECOND RUN FOR STABILITY DERIVATIVE CASE  * /     APL20440
     2       37X,11(4H****),2H**)                                       APL20450
C  IF THIS IS A SYMETRIC WING, SWITCH IT TO ANTI-SYMETRIC FOR RUN 2     APL20460
   80 IF(ISYMM .EQ. 0) ISYMM = -1                                       APL20470
C                                                                       APL20480
C  STORE THE FIRST RUN SOLUTION ON UNIT 1, DEFINE THE FUNDAMENTAL CASES APL20490
C  FOR YAWING AND ROLLING RATES, AND PRINT THE NEW FUND CASE GEOMETRY.   APL20500
   90 LOGIC = 2                                                         APL20510
      CALL OVERLAY(4HOVLY,4,-1)                                         APL20520
C                                                                       APL20530
C  SET UP AND SOLVE THE MATRIX SYSTEM FOR THE SECOND RUN                APL20540
      LOGIC = 2                                                         APL20550
      CALL OVERLAY(4HOVLY,2,-1)                                         APL20560
      IF(IR .EQ. 3) GO TO 110                                           APL20570
C                                                                       APL20580
C  CALCULATE AND PRINT THE DERIVATIVES FOR ALL FUNDAMENTAL              APL20590
C  AND COMPOSITE CASES                                                  APL20600
      LOGIC = 3                                                         APL20610
      IF(IPRINT .GE. 0) IPRINT = 2                                      APL20615
      CALL OVERLAY(4HOVLY,3,-1)                                         APL20620
C                                                                       APL20630
C**********************************************************************APL20640
C  THIS IS THE END OF THE LINE                                          APL20650
  100 IR = 1                                                            APL20660
      RETURN                                                            APL20670
C                                                                       APL20680
C  A FATAL ERROR HAS OCCURED.  RETURN ABNORMALLY TO MAIN.               APL20690
  110 IR = 2                                                            APL20700
      RETURN                                                            APL20710
C**********************************************************************APL20720
      END                                                               APL20730
```

```
*DECK STAGE1
      OVERLAY(OVLY,1,-1)                                                STG10010
      PROGRAM STAGE1                                                    STG10020
C                                                                       STG10030
C  THIS SUBROUTINE READS THE GENERAL GEOMETRY PARAMETERS AND FLAGS, AND STG10040
C  CONTROLS THE CALLING OF THE SPECIALIZED GEOMETRY SUBROUTINES         STG10050
C                                                                       STG10060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETELG,IGTYPE,IHINGE            STG10070
      COMMON/MARK/NJONS,NEQNSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) STG10080
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALEA,LOGIC,IR                      STG10090
C                                                                       STG10100
C  CHECK WHETHER THIS IS THE FIRST CMU CASE                             STG10110
      IF(NEWCMU .GT. 1) GO TO 50                                        STG10120
      IF((NROWS .GT. 40) .OR. (NROWS .LT. 3)) GO TO 80                  STG10130
C                                                                       STG10140
C  SECTIONAL INPUT                                                      STG10150
   10 IF((IGTYPE .EQ. 1).OR.(IGTYPE .EQ. 2))CALL SGMAIN(NOALEA,IR)      STG10160
      GO TO ( 20 , 40 , 100 ), IR                                       STG10170
C                                                                       STG10180
C  USER INPUT                                                           STG10190
C                                                                       STG10200
C  PRINT ERROR MESSAGE BECAUSE IGTYPE HAS THE WRONG VALUE               STG10210
   20 WRITE(6, 30 ) IGTYPE                                              STG10220
   30 FORMAT(1H1////44X,22HTHE IGTYPE FLAG HAS THE VALUE OF,I2/         STG10230
     1       44X,37HONLY THE VALUES 1 OR 2 ARE ACCEPTABLE//             STG10240
     2       44X,29HTHIS CASE HAS BEEN TERMINATED)                      STG10250
      GO TO 100                                                         STG10260
C                                                                       STG10270
C  READ THE COMPOSITE CASE REQUIREMENTS                                 STG10280
   40 CALL INCOMP(NCASES,IR)                                            STG10290
      IF(IR .EQ. 2) GO TO 100                                           STG10300
C                                                                       STG10310
C  READ THE CMU DATA                                                    STG10320
   50 CALL BLOWIN(JETELG,IR)                                            STG10330
      GO TO ( 60 , 110 , 120 ), IR                                      STG10340
   60 CALL BOX1(NEWMAX,IR)                                              STG10350
      IF(IR .EQ. 2) GO TO 50                                            STG10360
C                                                                       STG10370
C  RETURN NORMALLY TO THE CONTROL PROGRAM                               STG10380
   70 IR = 1                                                            STG10390
      GO TO 130                                                         STG10400
C                                                                       STG10410
C  PRINT ERROR MESSAGE BECAUSE THE NROWS VALUE IS UNACCEPTABLE          STG10420
   80 WRITE(6, 90 ) NROWS                                               STG10430
   90 FORMAT(1H1/55X,7HNROWS =,I3)                                      STG10440
C                                                                       STG10450
C**********************************************************************STG10460
C  A FATAL ERROR HAS OCCURED.  RETURN ABNORMALLY TO MAIN.               STG10470
  100 IR = 4                                                            STG10480
      GO TO 130                                                         STG10490
C                                                                       STG10500
C  THIS RUN HAS BEEN COMPLETED.  THANK GOD FOR SMALL BLESSINGS.         STG10510
C  RETURN TO MAIN AND BEGIN A COMPLETELY NEW RUN                        STG10520
  110 IR = 2                                                            STG10530
      GO TO 130                                                         STG10540
C                                                                       STG10550
C  RETURN TO MAIN AND STOP THE EXECUTION                                STG10560
  120 IR = 3                                                            STG10570
  130 CONTINUE                                                          STG10580
C**********************************************************************STG10590
      END                                                               STG10600
```

```
*DECK SGMAIN
      SUBROUTINE SGMAIN(NOALEA,IR)                                      SGMN0010
C                                                                       SGMN0020
C  THIS SUBROUTINE CONTROLS ALL GEOMETRY CALCULATIONS FOR THE           SGMN0030
C  SECTIONAL GEOMETRY METHOD                                            SGMN0040
C                                                                       SGMN0050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETELG,IGTYPE,IHINGE            SGMN0060
C                                                                       SGMN0070
C  READ THE WING PLANFORM GEOMETRY DATA                                 SGMN0080
   10 CALL INPTS(IR)                                                    SGMN0090
      IF(IR .EQ. 2) GO TO 100                                           SGMN0100
      IF(IGTYPE .EQ. 1) CALL XLETR1(IR)                                 SGMN0110
      IF(IR .EQ. 2) GO TO 100                                           SGMN0120
      IF(IGTYPE .EQ. 2) CALL XLETR2                                     SGMN0130
C                                                                       SGMN0140
C  NORMALIZE THE WING PLANFORM GEOMETRY DATA                            SGMN0150
   20 CALL NORM1                                                        SGMN0160
C                                                                       SGMN0170
C  READ THE JET SHEET GEOMETRY DATA                                     SGMN0180
   30 CALL INPUTJ(IR)                                                   SGMN0190
      IF(IR .EQ. 2) GO TO 100                                           SGMN0200
C                                                                       SGMN0210
C  CONSTRUCT THE EVD ELEMENTS                                           SGMN0220
   40 CALL BOXS(IR)                                                     SGMN0230
      IF(IR .EQ. 2) GO TO 100                                           SGMN0240
C                                                                       SGMN0250
C  CONSTRUCT THE SET OF FUNDAMENTAL GEOMETRIC CASES                     SGMN0260
C                                                                       SGMN0270
      DO 90 N = 1,NCASES                                                SGMN0280
      LCASE = N                                                         SGMN0290
C                                                                       SGMN0300
C  READ THE GEOMETRY FOR THIS CASE                                      SGMN0310
   50 CALL INCASE(LCASE,NOALEA)                                         SGMN0320
C                                                                       SGMN0330
C  CONSTRUCT THE CASE DATA                                              SGMN0340
   60 CALL RECCFE(LCASE,NOALEA,IR)                                      SGMN0350
      IF(IR .EQ. 2) GO TO 100                                           SGMN0360
C                                                                       SGMN0370
C  PRINT THE GEOMETRY AND CONSTRUCTED CASE DATA IF REQUIRED             SGMN0380
      IF(LCASE .EQ. 1) WRITE(6, 70 )                                    SGMN0390
   70 FORMAT(1H1)                                                       SGMN0400
   80 CALL OUT1(LCASE)                                                  SGMN0410
   90 CONTINUE                                                          SGMN0420
      IR = 2                                                            SGMN0430
      RETURN                                                            SGMN0440
C                                                                       SGMN0450
C  AN ERROR HAS OCCURED.  RETURN ABNORMALLY TO STAGE1.                  SGMN0460
C                                                                       SGMN0470
  100 IR = 3                                                            SGMN0480
      RETURN                                                            SGMN0490
      END                                                               SGMN0500
```

C2

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

C3

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK INPUTS                                                          INPS0010
      SUBROUTINE INPTS(IR)                                            INPS0020
C                                                                     INPS0030
C  THIS SUBROUTINE READS THE WING GEOMETRY DATA                       INPS0040
C  FOR THE SECTIONAL GEOMETRY METHOD                                  INPS0050
C                                                                     INPS0060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) INPS0070
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(6CC),XI(6CC),DEL(6CC),INPS0080
     1                 D(4C),KK(6CC),ITYPE(6CC)                     INPS0090
      COMMON/GEOM2/XLEAD(4C),XTRAIL(4C),TANLE(4C),TANTE(4C)           INPS0100
      COMMON/SG1/XW(2C,1C),XBJ(2C,1C),ICTYPE(4C),IJTYPE(4C),          INPS0110
     1           NWTYPE,NJTYPE                                      INPS0120
      DIMENSION NI(1C)                                                INPS0130
C                                                                     INPS0140
C  READ THE SECTIONAL PLANFORM DATA                                   INPS0150
      IC NWTYPE = C                                                   INPS0160
         READ(5, 2C ) (Y(K),K=1,NROWS)                                INPS0170
   2C FORMAT(8E1C.6)                                                  INPS0180
         READ(5, 3C ) (ICTYPE(K),K=1,NROWS)                           INPS0190
   3C FORMAT(4C12)                                                    INPS0200
         DO 4C K = 1,NROWS                                            INPS0210
         IF(ICTYPE(K) .GT. NWTYPE) NWTYPE = ICTYPE(K)                 INPS0220
   4C CONTINUE                                                        INPS0230
         IF(NWTYPE .GT. 12) GO TO 9C                                  INPS0240
         READ(5, 3C ) (NI(N),N=1,NWTYPE)                              INPS0250
C                                                                     INPS0260
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE                 INPS0270
         DO 5C N = 1,NWTYPE                                           INPS0280
         NIN = NI(N)                                                  INPS0290
         IF((NIN .LT. 1) .OR. (NIN .GT. 2C)) GO TO 1CC                INPS0300
         READ(5, 2C ) (XW(L,N),L=1,NIN)                               INPS0310
   5C CONTINUE                                                        INPS0320
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW              INPS0330
         DO 7C  K = 1,NROWS                                           INPS0340
         ICK = ICTYPE(K)                                              INPS0350
   6C NW(K) = NI(ICK)                                                 INPS0360
   7C CONTINUE                                                        INPS0370
         IR = 1                                                       INPS0380
         RETURN                                                       INPS0390
C                                                                     INPS0400
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.                  INPS0410
   9C WRITE(6, 9C ) NWTYPE                                            INPS0420
   9C FORMAT(1H1/45X,26HNUMBER OF WING ROW TYPES =,I3)                INPS0430
         IR = 2                                                       INPS0440
         RETURN                                                       INPS0450
  1CC WRITE(6, 11C ) NIN,N                                            INPS0460
  11C FORMAT(1H1/38X,I3,38H WING ELEMENTS PRESCRIBED FOR ROW TYPE,I3) INPS0470
         IR = 2                                                       INPS0480
         RETURN                                                       INPS0490
         END
```

```
*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,ININGE          INPJ0070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) INPJ0080
      COMMON/SG1/XW(2C,1C),XBJ(2C,1C),ICTYPE(4C),IJTYPE(4C),          INPJ0090
     1           NWTYPE,NJTYPE                                      INPJ0100
      DIMENSION NI(1C)                                                INPJ0110
C                                                                     INPJ0120
C  READ THE TYPE OF DIVISION FOR EACH ROW                             INPJ0130
      IC NJTYPE = C                                                   INPJ0140
         NROWSJ = C                                                   INPJ0150
         IF(JETFLG .NE. 1) GO TO 11C                                  INPJ0160
         READ(5, 2C ) (IJTYPE(K),K=1,NROWS)                           INPJ0170
   2C FORMAT(4C12)                                                    INPJ0180
         DO 3C K = 1,NROWS                                            INPJ0190
         IF(IJTYPE(K) .GT. NJTYPE) NJTYPE = IJTYPE(K)                 INPJ0200
         IF(IJTYPE(K) .NE. 0) NROWSJ = NROWSJ + 1                     INPJ0210
   3C CONTINUE                                                        INPJ0220
         IF(NJTYPE .GT. 1C) GO TO 13C                                 INPJ0230
C  READ THE NUMBER OF CHORDWISE DIVISIONS IN EACH ROW TYPE            INPJ0240
         READ(5, 4C ) (NI(N),N=1,NJTYPE)                              INPJ0250
   4C FORMAT(1C12)                                                    INPJ0260
C  READ THE CHORDWISE DIVISION DATA FOR EACH ROW TYPE                 INPJ0270
         DO 6C N = 1,NJTYPE                                           INPJ0280
         NIN = NI(N)                                                  INPJ0290
         IF((NIN .LT. 1) .OR. (NIN .GT. 2C)) GO TO 15C                INPJ0300
         READ(5, 5C ) (XBJ(L,N),L=1,NIN)                              INPJ0310
   5C FORMAT(8E1C.4)                                                  INPJ0320
   6C CONTINUE                                                        INPJ0330
C                                                                     INPJ0340
C  DEFINE THE NUMBER OF CHORDWISE DIVISIONS FOR EACH ROW              INPJ0350
         DO 8C K = 1,NROWS                                            INPJ0360
         NJ(K) = C                                                    INPJ0370
         IF(IJTYPE(K) .EQ. 0) GO TO 8C                                INPJ0380
         IJK = IJTYPE(K)                                              INPJ0390
   7C NJ(K) = NI(IJK)                                                 INPJ0400
   8C 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 1CC 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 9C                                 INPJ0490
         IF(ICOUNT .LT. 3)GO TO 17C                                   INPJ0500
         ICOUNT = 1                                                   INPJ0510
         IF(NJ(K) .EQ. 0)ITEST = 0                                    INPJ0520
         IF(NJ(K) .GT. 0)ITEST = 1                                    INPJ0530
         GO TO 1CC                                                    INPJ0540
   9C ICOUNT = ICOUNT + 1                                             INPJ0550
  1CC CONTINUE                                                        INPJ0560
         IR = 1                                                       INPJ0570
         RETURN                                                       INPJ0580
C                                                                     INPJ0590
C  THERE IS NO JET FOR THIS RUN                                       INPJ0600
  11C DO 12C K = 1,NROWS                                              INPJ0610
         IJTYPE(K) = 0                                                INPJ0620
         NJ(K) = 0                                                    INPJ0630
  12C CONTINUE                                                        INPJ0640
         IR = 1                                                       INPJ0650
         RETURN                                                       INPJ0660
C                                                                     INPJ0670
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.                  INPJ0680
  13C WRITE(6, 14C ) NJTYPE                                           INPJ0690
  14C FORMAT(1H1/25HNUMBER OF JET ROW TYPES =,I3)                     INPJ0700
         IR = 2                                                       INPJ0710
         RETURN                                                       INPJ0720
  15C WRITE(6, 16C ) NIN,N                                            INPJ0730
  16C FORMAT(1H1/38X,I3,37H JET ELEMENTS PRESCRIBED FOR ROW TYPE,I3)  INPJ0740
         IR = 2                                                       INPJ0750
         RETURN                                                       INPJ0760
  17C WRITE(6, 18C )                                                  INPJ0770
  18C FORMAT(1H1,29H3 ROW CONTINUITY RULE FAILURE)                    INPJ0780
         IR = 2                                                       INPJ0790
         RETURN                                                       INPJ0800
         END
```

```
*DECK XLETRJ                                                          XLTJC010
      SUBROUTINE XLETR1(IR)                                           XLTJC020
C                                                                     XLTJC030
C  THIS SUBROUTINE READS THE LEADING AND TRAILING EDGE COORDINATES    XLTJC040
C  AT SPANWISE STATIONS CONNECTED BY STRAIGHT LEADING AND TRAILING EDGESXLTJC050
C  AND INTERPOLATES TO GET THE COORDINATES AT INTERMEDIATE SECTIONS    XLTJC060
C                                                                     XLTJC070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) XLTJC080
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(6CC),XI(6CC),DEL(6CC),XLTJC090
     1                 D(4C),KK(6CC),ITYPE(6CC)                     XLTJC100
      COMMON/GEOM2/XLEAD(4C),XTRAIL(4C),TANLE(4C),TANTE(4C)           XLTJC110
      DIMENSION YP(4C),XLE(4C),XTR(4C)                                XLTJC120
C                                                                     XLTJC130
C  READ XLEAD AND XTRAIL                                              XLTJC140
      NX = C                                                          XLTJC150
      DO 3C N = 1,NROWS                                               XLTJC160
      READ(5, 1C )YP(N),XLE(N),XTR(N)                                 XLTJC170
   1C FORMAT(3E1C.6)                                                  XLTJC180
   2C IF(YP(N) .GT. 1.1) GO TO 4C                                     XLTJC190
      NX = NX + 1                                                     XLTJC200
   3C CONTINUE                                                        XLTJC210
C  CHECK WHETHER THE Y VALUES ARE REALISTIC                           XLTJC220
   4C IF(ABS(YP(1)-Y(1)) .GT. 0.CCC1)GO TO 11C                        XLTJC230
      IF(ABS(YP(NX)-Y(NROWS)) .GT. 0.CCC1)GO TO 11C                   XLTJC240
C                                                                     XLTJC250
C  READ THE EXTRA 9 CARD IF NROWS CARDS HAVE BEEN INPUT               XLTJC260
      IF(NX .EQ. NROWS) READ(5, 1C ) EXTRA9                           XLTJC270
C                                                                     XLTJC280
C  INTERPOLATE FOR XLEAD AND XTRAIL AT THE INTERMEDIATE SECTIONS      XLTJC290
      K = 1                                                           XLTJC300
      DO 1CC N = 1,NX                                                 XLTJC310
   5C K = K + 1                                                       XLTJC320
      IF(K .GT. NROWS) GO TO 11C                                      XLTJC330
      IF(ABS(YP(N)-Y(K)) .GT. 0.CCC1)GO TO 7C                         XLTJC340
C  XLE AND XTR WERE INPUT FOR ROW K                                   XLTJC350
   6C XLEAD(K) = XLE(N)                                               XLTJC360
      XTRAIL(K) = XTR(N)                                              XLTJC370
      GO TO 1CC                                                       XLTJC380
C  XLE AND XTR MUST BE INTERPOLATED FOR ROW K                         XLTJC390
   7C NM1 = N - 1                                                     XLTJC400
   8C YRATIO = (Y(K)-YP(N)) / (YP(NM1)-YP(N))                         XLTJC410
      XLEAD(K) = XLE(N) + YRATIO * (XLE(NM1)-XLE(N))                  XLTJC420
   9C XTRAIL(K) = XTR(N) + YRATIO * (XTR(NM1)-XTR(N))                 XLTJC430
      GO TO 5C                                                        XLTJC440
  1CC CONTINUE                                                        XLTJC450
      IR = 1                                                          XLTJC460
      RETURN                                                          XLTJC470
C                                                                     XLTJC480
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND RETURN.                XLTJC490
  11C WRITE(6, 12C )                                                  XLTJC500
  12C FORMAT(1H1////2CX,33HAN INCONSISTENCY HAS BEEN FOUND IN THE,    XLTJC510
     1       42H SECTIONAL LEADING AND TRAILING EDGE INPUT)           XLTJC520
      IR = 2                                                          XLTJC530
      RETURN                                                          XLTJC540
      END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK XLETR2
      SUBROUTINE XLETR2                                                 XLT20C10
C                                                                       XLT20020
C  THIS SUBROUTINE READS THE FUNDAMENTAL PLANFORM PARAMETERS FOR A      XLT20030
C  TRAPEZOIDAL WING, AND CALCULATES THE LEADING AND TRAILING EDGE       XLT20040
C  COORDINATES AT EACH Y STATION.  NOTE THAT THE PLANFORM OUTLINE       XLT20050
C  MUST BE SYMETRIC.                                                    XLT20060
C                                                                       XLT20070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) XLT20C80
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     XLT20090
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XR(600),XI(600),DEL(600),  XLT20100
     1            D(40),KK(600),ITYPE(600)                              XLT20110
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             XLT20120
C                                                                       XLT20130
C  READ THE FUNDAMENTAL PLANFORM PARAMETERS                             XLT20140
      READ(5, 10 ) ARATIO,SWEEP,TR                                      XLT20150
   10 FORMAT(4F10.6)                                                    XLT20160
C                                                                       XLT20170
C  COMPUTE THE GENERAL PLANFORM CHARACTERISTICS                         XLT20180
      B2 = SPAN / 2.00                                                  XLT20190
      SW = SWEEP / 57.295779                                            XLT20200
   20 CROOT = 2.0 * SPAN / ((1.0+TR)*ARATIO)                            XLT20210
      AREA = ((1.0+TR) * CROOT * B2                                     XLT20220
      XLB2 = 0.250 * ((1.0-TR) * CROOT + B2 * TAN(SW)                   XLT20230
   30 CMAC = 2.0 * CROOT * (1.0 + TR + TR*TR) / (3.0*(1.0+TR))          XLT20240
      IF(CREF .EQ. 0.0) CREF = CMAC                                     XLT20250
      CBAR=AREA/SPAN                                                    XLT20260
C                                                                       XLT20270
C  COMPUTE THE LEADING AND TRAILING EDGE COORDINATES                    XLT20280
   40 DO 60 K = 1,NROWS                                                 XLT20290
      YBAR = Y(K)                                                       XLT20300
      IF(YBAR .LT. 0.0) YBAR = -YBAR                                    XLT20310
      XLEAD(K) = XLB2 * YBAR                                            XLT20320
   50 C = CROOT * ((1.0-(1.0-TR)*YBAR)                                  XLT20330
      XTRAIL(K) = XLEAD(K) + C                                          XLT20340
   60 CONTINUE                                                          XLT20350
      RETURN                                                            XLT20360
      END                                                               XLT20370


*DECK NORM1
      SUBROUTINE NORM1                                                  NORMC010
C                                                                       NORMC020
C  THIS SUBROUTINE NORMALIZES ALL WING PLANFORM GEOMETRY BY SPAN/2      NORMC030
C                                                                       NORMC040
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) NORMC050
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     NORMC060
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XR(600),XI(600),DEL(600),  NORMC070
     1            D(40),KK(600),ITYPE(600)                              NORMC080
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             NORMC090
   10 B2 = SPAN / 2.00                                                  NORM0100
      AREA = AREA / B2**2                                               NORM0110
      CREF = CREF / B2                                                  NORM0120
   20 XMC = XMC / B2                                                    NORM0130
      XCG = XCG / B2                                                    NORM0140
      DO 40 K = 1,NROWS                                                 NORM0150
   30 XLEAD(K) = XLEAD(K) / B2                                          NORM0160
      XTRAIL(K) = XTRAIL(K) / B2                                        NORM0170
   40 CONTINUE                                                          NORM0180
      SPAN = 2.00                                                       NORM0190
      ARATIO = SPAN * SPAN / AREA                                       NORM0200
      RETURN                                                            NORM0210
      END                                                               NORM0220


*DECK BOXS
      SUBROUTINE BOXS(IR)                                               BOXS0010
C                                                                       BOXS0020
C  THIS SUBROUTINE CONSTRUCTS THE GEOMETRIC PARAMETERS FOR ALL THE      BOXS0030
C  EVD ELEMENTS ON THE WING AND JET                                     BOXS0040
C                                                                       BOXS0050
      COMMON/MATHEW/NCASES,ISYPM,IPRINT,JETFLG,IGTYPE,IMINGE           BOXS0060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BOXS0070
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     BOXS0080
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XR(600),XI(600),DEL(600),  BOXS0090
     1            D(40),KK(600),ITYPE(600)                              BOXS0100
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             BOXS0110
      COMMON/SGL/XBW(20,10),XBJ(20,10),ICTYPE(40),IJTYPE(40),           BOXS0120
     1           NWTYPE,NJTYPE                                          BOXS0130
C                                                                       BOXS0140
C  CONSTRUCT THE ELEMENTS ON THE WING                                   BOXS0150
C                                                                       BOXS0160
C  COMPUTE SECTIONAL DATA                                               BOXS0170
   10 CHORD(1) = XTRAIL(1) - XLEAD(1)                                   BOXS0180
      DELTA(1) = 1.00 - Y(1)                                            BOXS0190
      CMAC = CHORD(1)**2 * DELTA(1)                                     BOXS0200
      DO 30 K = 2,NROWS                                                 BOXS0210
   20 CHORD(K) = XTRAIL(K) - XLEAD(K)                                   BOXS0220
      DELTA(K) = Y(K-1) - Y(K) - DELTA(K-1)                             BOXS0230
      IF(DELTA(K) .LT. 0.0) GO TO 190                                   BOXS0240
      CMAC = CMAC + CHORD(K)**2 * DELTA(K)                              BOXS0250
   30 CONTINUE                                                          BOXS0260
C  CHECK THE VALIDITY OF THE SECTIONAL ALLIGNMENT                       BOXS0261
      TD = Y(NROWS) - DELTA(NROWS)                                      BOXS0262
      IF((ISYMM .GE. 0) .AND. (ABS(TD) .GT. 0.00011)) GO TO 190         BOXS0263
      IF((ISYMM .EQ. 1) .AND. (ABS(TD+1.0) .GT. 0.0001)) GO TO 190      BOXS0264
      DSUM = DELTA(1)                                                   BOXS0265
      DO 35 K = 2,NROWS                                                 BOXS0266
      YL = Y(K) + DELTA(K)                                              BOXS0267
      YR = Y(K-1) - DELTA(K-1)                                          BOXS0268
      IF(ABS(YR-YL) .GT. 0.CC1) GO TO 190                               BOXS0269
      DSUM = DSUM + DELTA(K)                                            BOXS0270
   35 CONTINUE                                                          BOXS0271
      IF(ABS(DSUM-0.50) .GT. C.0001) GO TO 190                          BOXS0272
      CMAC = 2.0 * CMAC / AREA                                          BOXS0278
      IF(ISYMM .LT. 1) CMAC = 2.0 * CMAC                                BOXS0280
      IF(CREF .LT. 0.0001) CREF = CMAC                                  BOXS0290
      CALL TANS(TANLE,XLEAD,Y,NROWS)                                    BOXS0295
C                                                                       BOXS0300
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION            BOXS0310
      I = 0                                                             BOXS0320
      DO 90 K = 1,NROWS                                                 BOXS0330
C  COMPUTE X-COORDINATES                                                BOXS0340
      NWK = NW(K)                                                       BOXS0350
      DO 50 L = 1,NWK                                                   BOXS0360
      I = I + 1                                                         BOXS0370
      ICK = ICTYPE(K)                                                   BOXS0380
   40 XB(I) = XBW(L,ICK)                                                BOXS0390
   50 CONTINUE                                                          BOXS0400
C  COMPUTE ALL OTHER PARAMETERS                                         BOXS0410
      I = I - NWK                                                       BOXS0420
      IW(K) = I + 1                                                     BOXS0430
      DO 80 L = 1,NWK                                                   BOXS0440
      I = I + 1                                                         BOXS0450
   60 KK(I) = K                                                         BOXS0460
      DEL(I) = XB(I+1) - XB(I)                                          BOXS0470
   70 XI(I) = XLEAD(K) + XB(I) * CHORD(K)                               BOXS0480
      ITYPE(I) = 10                                                     BOXS0490
   80 CONTINUE                                                          BOXS0500
C  REDEFINE THE LAST DEL IN THIS SECTION, AND DEFINE THE L.E. EVD TYPE  BOXS0510
      DEL(I) = 1.00 - XB(I)                                             BOXS0520
      IWK = IW(K)                                                       BOXS0530
      ITYPE(IWK) = 20                                                   BOXS0540
   90 CONTINUE                                                          BOXS0550
      NWT = I                                                           BOXS0560
C                                                                       BOXS0570
C  CONSTRUCT THE ELEMENTS ON THE JET SHEET                              BOXS0580
C                                                                       BOXS0590
C  COMPUTE ALL CHORDWISE ELEMENT PARAMETERS FOR EACH SECTION            BOXS0600
      IF(JETFLG .NE. 0) GO TO 180                                       BOXS0610
      DO 170 K = 1,NROWS                                                BOXS0620
C  COMPUTE X-COORDINATES                                                BOXS0630
      IJ(K) = 0                                                         BOXS0640
  100 NJK = NJ(K)                                                       BOXS0650
      IF(NJK .EQ. 0) GO TO 170                                          BOXS0660
      DO 120 L = 1,NJK                                                  BOXS0670
      I = I + 1                                                         BOXS0680
      IJK = IJTYPE(K)                                                   BOXS0690
  110 XB(I) = XBJ(L,IJK)                                                BOXS0700
  120 CONTINUE                                                          BOXS0710
C  COMPUTE ALL OTHER PARAMETERS                                         BOXS0720
      I = I - NJK                                                       BOXS0730
  130 IJ(K) = I + 1                                                     BOXS0740
      DO 160 L = 1,NJK                                                  BOXS0750
      I = I + 1                                                         BOXS0760
  140 KK(I) = K                                                         BOXS0770
      DEL(I) = XB(I+1) - XB(I)                                          BOXS0780
  150 XI(I) = XLEAD(K) + XB(I) * CHORD(K)                               BOXS0790
      ITYPE(I) = 10                                                     BOXS0800
  160 CONTINUE                                                          BOXS0810
C  REDEFINE THE LAST DEL AND EVD TYPE AND THE D VALUE FOR THIS SECTION  BOXS0820
      DEL(I) = 1.0E10                                                   BOXS0830
      ITYPE(I) = 30                                                     BOXS0840
      D(K) = XI(I) - XTRAIL(K)                                          BOXS0850
  170 CONTINUE                                                          BOXS0860
  180 NMAX = I                                                          BOXS0870
      IF(NMAX .GT. 600) GO TO 210                                       BOXS0880
      NJT = NMAX - NWT                                                  BOXS0890
      IR = 1                                                            BOXS0900
      RETURN                                                            BOXS0910
C                                                                       BOXS0920
C  AN ERROR HAS OCCURED.   PRINT A MESSAGE AND QUIT.                    BOXS0930
  190 WRITE(6, 200 )                                                    BOXS0940
  200 FORMAT(1H1/38X,44HPLEASE CHECK YOUR SECTION LOCATION (Y) INPUT)   BOXS0950
      IR = 2                                                            BOXS0960
      RETURN                                                            BOXS0970
  210 WRITE(6, 220 ) NMAX                                               BOXS0980
  220 FORMAT(1H1/49X,I4,21H IS TOO MANY ELEMENTS)                       BOXS0990
      IR = 2                                                            BOXS1000
      RETURN                                                            BOXS1010
      END                                                               BOXS1020


*DECK TANS
      SUBROUTINE TANS(TAN,X,Y,NROWS)                                    TANS0010
C                                                                       TANS0020
C  THIS SUBROUTINE COMPUTES THE TANGENT OF THE LEADING OR TRAILING EDGE TANS0030
C  SWEEP ANGLE  AT THE CENTERLINE OF EACH SECTION.  IT IS ACCURATE FOR  TANS0040
C  SECTIONS WITH STRAIGHT EDGES IN GROUPS OF THREE OR MORE.             TANS0050
C  IT IS ONLY APPROXIMATE FOR CURVED EDGES.                             TANS0060
C  IT MAY RESULT IN ERRORS FOR SECTIONS ADJACENT TO WING BREAKS,        TANS0070
C  IF STRAIGHT EDGES ARE IN ADJACENT GROUPS OF ONLY ONE OR TWO.         TANS0080
C                                                                       TANS0090
      DIMENSION TAN(40),X(40),Y(40),S(40)                               TANS0100
      SLOP(XR,XL,YR,YL) = (XR-XL) / (YR-YL)                             TANS0110
C                                                                       TANS0120
      DO 20 K = 1,NROWS                                                 TANS0130
      KR = K-1                                                          TANS0140
      KL = K                                                            TANS0150
      IF(K .GT. 1) GO TO 10                                             TANS0160
      KR = 1                                                            TANS0170
      KL = 2                                                            TANS0180
   10 S(K) = SLOP(X(KR),X(KL),Y(KR),Y(KL))                              TANS0190
   20 CONTINUE                                                          TANS0200
      DO 50 K = 1,NROWS                                                 TANS0210
      IF(K .LT. 3) GO TO 30                                             TANS0220
      IF(K .EQ. NROWS) GO TO 30                                         TANS0230
      IF(K .EQ. (NROWS-1)) GO TO 40                                     TANS0240
C  CHECK WHETHER THE RIGHT OR LEFT SIDES ARE STRAIGHT                   TANS0250
      IF(ABS(S(K) - S(K-1)) .LT. C.001) GO TO 30                        TANS0260
      IF(ABS(S(K+1) - S(K+2)) .LT. 2.001) GO TO 40                      TANS0270
C  NEITHER SIDE IS CONCLUSIVELY STRAIGHT - CHECK FURTHER LEFT AND RIGHT TANS0280
      IF(K .EQ. 3) GO TO 40                                             TANS0290
      IF(K .EQ. (NROWS-2)) GO TO 30                                     TANS0300
      IF(ABS(S(K-1) - S(K-2)) .LT. 0.001) GO TO 40                      TANS0310
      IF(ABS(S(K+2) - S(K+3)) .LT. 0.001) GO TO 30                      TANS0320
```

```
C  THE TRUE SHAPE CANNOT BE DETERMINED - GIVE UP AND TAKE THE AVERAGE   TANSC330
      TAN(K) = (S(K) + S(K+1)) / 2.00                                   TANSC340
      GO TO 50                                                          TANSC350
C  THE RIGHT EDGE IS STRAIGHT                                           TANSC360
   30 TAN(K) = S(K)                                                     TANSC370
      GO TO 50                                                          TANSC380
C  THE LEFT EDGE IS STRAIGHT                                            TANSC390
   40 TAN(K) = S(K+1)                                                   TANSC400
   50 CONTINUE                                                          TANSC410
      RETURN                                                            TANSC420
      END                                                               TANSC430


*DECK INCASE
      SUBROUTINE INCASE(LCASE,NOALFA)                                   INCSC010
C                                                                       INCSC020
C  THIS SUBROUTINE READS THE FUNDAMENTAL GEOMETRIC CASE DATA            INCSC030
C                                                                       INCSC040
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) INCSC050
      COMMON/ECASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTR                  INCSC060
      COMMON/ECASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     INCSC070
     1              XHR(4,40),RET(4,40),FST(4,40),ICT(40),IHT(40),NCT,NHT INCSC080
      DIMENSION NI(10),DUMMY(40)                                        INCSC090
C                                                                       INCSC100
C  INITIALIZE SECTIONAL DATA                                            INCSC110
      DO 3C K = 1,NROWS                                                 INCSC120
   10 TST(K,LCASE) = 0.00                                               INCSC130
      HL(K,LCASE) = 0.00                                                INCSC140
      DJ(K) = 0.00                                                      INCSC150
      ACTE(K) = 0.00                                                    INCSC160
      ICT(K) = 0                                                        INCSC170
      IHT(K) = 0                                                        INCSC180
C  INITIALIZE THE CAMBER ANGLES                                         INCSC190
      NWK = NW(K)                                                       INCSC200
      DO 2C L = 1,NWK                                                   INCSC210
      AC(L,K) = 0.00                                                    INCSC220
   20 CONTINUE                                                          INCSC230
   30 CONTINUE                                                          INCSC240
C  INITIALIZE THE HINGE DATA                                            INCSC250
      DO 5C N = 1,NROWS                                                 INCSC260
      DO 4C L = 1,4                                                     INCSC270
      XHR(L,N) = 0.00                                                   INCSC280
      RET(L,N) = 0.00                                                   INCSC290
   40 CONTINUE                                                          INCSC300
   50 CONTINUE                                                          INCSC310
C                                                                       INCSC320
      IF(LCASE .EQ. 1) .AND. (NOALFA .GT. 0)) RETURN                    INCSC330
C                                                                       INCSC340
C  READ FUNDAMENTAL CASE CONTROL FLAGS                                  INCSC350
      READ(5, 6C ) INPUTT,INPUTH,INPUTD,INPUTC,INPUTR                   INCSC360
   6C FORMAT(512)                                                       INCSC370
C                                                                       INCSC380
C  READ SECTIONAL TWIST, HEIGHT AND JET DEFLECTION DATA                 INCSC390
      IF(INPUTT .NE. 0) READ(5, 7C ) (TST(K,LCASE),K=1,NROWS)           INCSC400
   7C FORMAT(8E10.4)                                                    INCSC410
      IF(INPUTH .NE. 0) READ(5, 7C ) (HL(K,LCASE),K=1,NROWS)            INCSC420
      IF(INPUTD .EQ. 0) GO TO 9C                                        INCSC430
      READ(5, 7C ) (DUMMY(K),K=1,NROWSJ)                                INCSC440
C  DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE DJ ARRAY                 INCSC450
      KP = 0                                                            INCSC460
      DO 8C K = 1,NROWS                                                 INCSC470
      IF(NJ(K) .EQ. 0) GO TO 8C                                         INCSC480
      KP = KP + 1                                                       INCSC490
      DJ(K) = DUMMY(KP)                                                 INCSC500
   8C CONTINUE                                                          INCSC510
C                                                                       INCSC520
C  READ THE CAMBER ANGLES, IN DEGREES                                   INCSC530
   9C IF(INPUTC .EQ. 0) GO TO 16C                                       INCSC540
      READ(5, 1CC ) ((ICT(K),K=1,NROWS)                                 INCSC550
  1CC FORMAT(4012)                                                      INCSC560
      NCT = 0                                                           INCSC570
      DO 11C K = 1,NROWS                                                INCSC580
      IF(ICT(K) .EQ. 0) GO TO 11C                                       INCSC590
      IF(ICT(K) .GT. NCT) NCT = ICT(K)                                  INCSC600
      ICK = ICT(K)                                                      INCSC610
      NI(ICK) = NW(K)                                                   INCSC620
  11C CONTINUE                                                          INCSC630
      DO 13C N = 1,NCT                                                  INCSC640
      NIN = NI(N)                                                       INCSC650
      READ(5, 7C ) (AC(L,N),L=1,NIN)                                    INCSC660
  13C CONTINUE                                                          INCSC670
      IF(NROWSJ .GT. 0) READ(5, 7C ) (DUMMY(K),K=1,NROWSJ)              INCSC680
C  DISTRIBUTE THE DUMMY VALUES PROPERLY IN THE ACTE ARRAY               INCSC690
      KP = 0                                                            INCSC700
      DO 15C K = 1,NROWS                                                INCSC710
      IF(NJ(K) .EQ. 0) GO TO 15C                                        INCSC720
      KP = KP + 1                                                       INCSC730
      ACTE(K) = DUMMY(KP)                                               INCSC740
  15C CONTINUE                                                          INCSC750
C                                                                       INCSC760
C  READ THE HINGE LOCATION, TYPE AND TURNING ANGLE DATA                 INCSC770
  16C IF(INPUTR .EQ. 0) GO TO 21C                                       INCSC780
  17C READ(5, 1CC ) (IHT(K),K=1,NROWS)                                 INCSC790
      NHT = 0                                                           INCSC800
      DO 18C K = 1,NROWS                                                INCSC810
      IF(IHT(K) .GT. NHT) NHT = IHT(K)                                  INCSC820
  18C CONTINUE                                                          INCSC830
      DO 2CC N = 1,NHT                                                  INCSC840
      READ(5, 19C ) (XHR(L,N),FST(L,N),RET(L,N),L=1,4)                  INCSC850
  19C FORMAT(4(F10.4,F8.4))                                             INCSC860
  2CC CONTINUE                                                          INCSC870
  21C RETURN                                                            INCSC880
      END                                                               INCSC890


*DECK RECCFE
      SUBROUTINE RECCFE(LCASE,NOALFA,IR)                                RECFC010
C                                                                       RECFC020
C  THIS SUBROUTINE CONSTRUCTS THE BOUNDARY CONDITION ARRAYS FOR THE     RECFC030
C  FUNDAMENTAL GEOMETRIC CASES                                          RECFC040
C                                                                       RECFC050
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) RECFC060
      COMMON/GROM1/XY(40),CHORD(40),DELTA(40),YA(40),XT(40),DEL(40),    RECFC070
     1             DI(40),KK(40),ITYPE(40)                              RECFC080
      COMMON/ECASE1/INPUTT,INPUTH,INPUTD,INPUTC,INPUTR                  RECFC090
      COMMON/ECASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),     RECFC100
     1              XHR(4,40),RET(4,40),FST(4,40),ICT(40),IHT(40),NCT,NHT RECFC110
      COMMON/ECASE3/EPS(40,10),BETA(40,10),THETA(40,10),THS(40,10)      RECFC120
      DIMENSION NH(40)                                                  RECFC130
C                                                                       RECFC140
C  INITIALIZE THE BOUNDARY CONDITION ANGLES                             RECFC150
      DO 2C I = 1,NMAX                                                  RECFC160
   10 EPS(I,LCASE) = 0.00                                               RECFC170
      BETA(I,LCASE) = 0.00                                              RECFC180
   20 CONTINUE                                                          RECFC190
      DO 3C K = 1,NROWS                                                 RECFC200
      THETA(K,LCASE) = 0.00                                             RECFC210
      THS(K,LCASE) = 0.00                                               RECFC220
   30 CONTINUE                                                          RECFC230
C                                                                       RECFC240
C  DEFINE THE ANGLES FOR THE ANGLE-OF-ATTACK FUNDAMENTAL CASE           RECFC250
      IF((LCASE .GT. 1) .OR. (NOALFA .EQ. 0)) GO TO 60                  RECFC260
      DO 40 I = 1,NWT                                                   RECFC270
      EPS(I,1) = 1.00                                                   RECFC280
   40 CONTINUE                                                          RECFC290
      DO 50 K = 1,NROWS                                                 RECFC300
      IF(NJ(K) .GT. 0) THETA(K,1) = 1.000                               RECFC310
   50 CONTINUE                                                          RECFC320
      IR = 1                                                            RECFC330
      RETURN                                                            RECFC340
C                                                                       RECFC350
C  DEFINE THE ANGLES FOR ALL REMAINING FUNDAMENTAL CASES                RECFC360
C                                                                       RECFC370
C  CAMBER CONTRIBUTION                                                  RECFC380
   60 IF(INPUTC .EQ. 0) GO TO 110                                       RECFC390
      I = 0                                                             RECFC400
      DO 100 K = 1,NROWS                                                RECFC410
      IF(NJ(K) .EQ. 0) GO TO 70                                         RECFC420
      THETA(K,LCASE) = ACTE(K)                                          RECFC430
   70 NWK = NW(K)                                                       RECFC440
      IF(ICT(K) .EQ. 0) GO TO 90                                        RECFC450
      DO 80 L = 1,NWK                                                   RECFC460
      I = I + 1                                                         RECFC470
      ICK = ICT(K)                                                      RECFC480
      EPS(I,LCASE) = AC(L,ICK)                                          RECFC490
   80 CONTINUE                                                          RECFC500
      GO TO 100                                                         RECFC510
   90 I = I + NWK                                                       RECFC520
  100 CONTINUE                                                          RECFC530
C                                                                       RECFC540
C  TWIST CONTRIBUTION                                                   RECFC550
  110 IF(INPUTT .EQ. 0) GO TO 160                                       RECFC560
      I = 0                                                             RECFC570
      DO 150 K = 1,NROWS                                                RECFC580
      IF(NJ(K) .EQ. 0) GO TO 120                                        RECFC590
      THETA(K,LCASE) = THETA(K,LCASE) + TST(K,LCASE)                    RECFC600
  120 NWK = NW(K)                                                       RECFC610
      DO 140 L = 1,NWK                                                  RECFC620
      I = I + 1                                                         RECFC630
  130 EPS(I,LCASE) = EPS(I,LCASE) + TST(K,LCASE)                        RECFC640
  140 CONTINUE                                                          RECFC650
  150 CONTINUE                                                          RECFC660
C                                                                       RECFC670
C  FLAP AND SLAT DEFLECTION CONTRIBUTION                                RECFC680
C                                                                       RECFC690
  160 IF(INPUTA .EQ. 0) GO TO 320                                       RECFC700
C                                                                       RECFC710
C  SUM UP THE TOTAL SLAT ANGLE AND FIND THE NUMBER OF HINGES ON EACH ROW RECFC720
      DO 190 K = 1,NROWS                                                RECFC730
      NH(K) = 0                                                         RECFC740
      IF(IHT(K) .EQ. 0) GO TO 190                                       RECFC750
      DO 180 L = 1,4                                                    RECFC760
      N = IHT(K)                                                        RECFC770
      IF(XHR(L,N) .LT. 0.001) GO TO 190                                 RECFC780
      NH(K) = NH(K) + 1                                                 RECFC790
      IF(FST(L,N) .GT. 0) THS(K,LCASE) = THS(K,LCASE) + RET(L,N)        RECFC800
  180 CONTINUE                                                          RECFC810
  190 CONTINUE                                                          RECFC820
C                                                                       RECFC830
C  COMPUTE INCIDENCE OF EACH ELEMENT AND FIND TURNING ANGLE AND EVD TYPE RECFC840
C  FOR EACH HINGE ELEMENT                                               RECFC850
      I = 0                                                             RECFC860
      DO 310 K = 1,NROWS                                                RECFC870
      NWK = NW(K)                                                       RECFC880
      N = IHT(K)                                                        RECFC890
  200 IF(N .EQ. 0) GO TO 300                                            RECFC900
      LSTART = 1                                                        RECFC910
      A = 0.0                                                           RECFC920
      NHK = NH(K)                                                       RECFC930
  210 IF(NHK .EQ. 0) GO TO 260                                          RECFC940
C  CYCLE THE HINGE POINTS IN CHORDWISE ORDER                            RECFC950
      DO 220 LH = 1,NHK                                                 RECFC960
C  CYCLE THE VORTEX POINTS IN CHORDWISE ORDER, LOOKING FOR NEXT HINGE   RECFC970
      DO 250 L = LSTART,NWK                                             RECFC980
      I = I + 1                                                         RECFC990
C  CHECK ON RELATIVE LOCATION OF VORTEX POINT AND HINGE POINT           RECFD000
  220 XDIFF = XHR(LH,N) - XT(I)                                         RECFD010
      IF(ABS(XDIFF) .LT. 0.001) GO TO 230                               RECFD020
      IF(XDIFF .GT. 0.001) GO TO 240                                    RECFD030
      IF(XDIFF .LT. -0.001) GO TO 300                                   RECFD040
C  THE ITH VORTEX POINT IS A HINGE POINT                                RECFD050
  230 A = A + RET(LH,N)                                                 RECFD060
      BETA(I,LCASE) = RET(LH,N)                                         RECFD070
```

C5

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

AD-A058 821 DOUGLAS AIRCRAFT CO LONG BEACH CALIF F/G 20/4
A THEORETICAL METHOD FOR CALCULATING THE AERODYNAMIC CHARACTERI--ETC(U)
1977 M L LOPEZ, C SHEN, N F WASSON N00014-71-C-0250
MDC-J5519-02-VOL-2 NL

UNCLASSIFIED
2 OF 2
AD
A058 821

END
DATE
FILMED
11-78
DDC

```
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B                    AEFE1080
      ITYPE(I) = 42                                                     AEFE1090
      IF(IFS(LH,N) .GT. 2) ITYPE(I) = 41                                AEFE1100
      GO TO 240                                                         AEFE1110
C  THE ITH VORTEX POINT IS NOT A HINGE POINT                            AEFE1120
  230 EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B                    AEFE1130
  250 CONTINUE                                                          AEFE1140
      LSTART = I - IW(K) + 1                                            AEFE1150
  270 CONTINUE                                                          AEFE1160
C  DEFINE THE INCIDENCE ANGLE FOR REMAINING POINTS BEHIND THE LAST HINGEAEFE1170
      IF(LSTART .EQ. NWK) GO TO 290                                     AEFE1180
      LSTART = LSTART + 1                                               AEFE1190
      DO 280 I = LSTART,NWK                                             AEFE1200
      I = I + 1                                                         AEFE1210
      EPS(I,LCASE) = EPS(I,LCASE) - THS(K,LCASE) + B                    AEFE1220
  280 CONTINUE                                                          AEFE1230
C  COMPUTE THE EFFECT OF THE HINGES ON THE JET ANGLE                    AEFE1240
  290 IF(NJK(K) .GT. 0) THETA(K,LCASE) = THETA(K,LCASE)-THS(K,LCASE)+B  AEFE1250
      GO TO 310                                                         AEFE1260
  300 I = I + NWK                                                       AEFE1270
  310 CONTINUE                                                          AEFE1280
C                                                                       AEFE1290
C  JET DEFLECTION CONTRIBUTION                                          AEFE1300
C                                                                       AEFE1310
  320 IF(INPUTD .EQ. 2) GO TO 350                                       AEFE1320
      DO 340 K = 1,NBDYS                                                AEFE1330
      IF(NJK(K) .EQ. 0) GO TO 340                                       AEFE1340
      I = IJ(K)                                                         AEFE1350
  330 BETA(I,LCASE) = DJ(K)                                             AEFE1360
      IF(ABS(DJ(K)) .GT. .0001) ITYPE(I) = 43                           AEFE1370
      THETA(K,LCASE) = THETA(K,LCASE) + DJ(K)                           AEFE1380
  340 CONTINUE                                                          AEFE1390
  350 IP = 1                                                            AEFE1400
      RETURN                                                            AEFE1410
C                                                                       AEFE1420
C  AN ERROR HAS OCCURED.  PRINT A MESSAGE AND QUIT.                     AEFE1430
C                                                                       AEFE1440
  360 WRITE(6, 370 ) LCASE,K,N                                          AEFE1450
  370 FORMAT(1H1///45X,24HFUNDAMENTAL GEOMETRIC CASE,I3/                AEFE1460
     1       18X,50HAN INCONSISTENCY HAS BEEN FOUND IN THE HINGE INPUT, AEFE1470
     2       18H DATA FOR WING BOX,I3,1OH, BOW TYPE,I3)                 AEFE1480
      IP = 2                                                            AEFE1490
      RETURN                                                            AEFE1500
      END                                                               AEFE1510
```

```
*DECK OUTI
      SUBROUTINE OUTI(LCASE)                                            OUTI0010
C                                                                       OUTI0020
C  THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE       OUTI0030
C  SECTIONAL METHOD INPUT                                               OUTI0040
C                                                                       OUTI0050
      COMMON/MATHEM/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           OUTI0060
      COMMON/MARK/IBOWS,IBOWS1,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) OUTI0070
      COMMON /ALPHA/ TITLE(18)                                          OUTI0080
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     OUTI0090
      COMMON/GEOM1/XV(40),CHORD(40),DELTA(40),XR(600),XI(600),DEL(600),OUTI0100
     1             Q(40),KK(40),ITYPE(600)                              OUTI0110
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)             OUTI0120
      COMMON/ECASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),BC(20,40),     OUTI0130
     1              XHR(4,40),BETA(600),IFS(4,40),ICT(40),IHT(40),NCT,NHTOUTI0140
      COMMON/ECASE3/EPS(600,10),BETA(600,10),THETA(40,10),THS(40,10)    OUTI0150
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NBO,NC,ISY,IPR,JET,IGT,IHT    OUTI0160
C                                                                       OUTI0170
C  PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS                    OUTI0180
      IF(LCASE .GT. 1) GO TO 40                                         OUTI0190
   10 WRITE(6, 20 ) TITLE                                               OUTI0200
   20 FORMAT(1H1,39X,18(5H****)//                                       OUTI0210
     1  42X,4CIM  FV3  JET - WING  COMPUTER  PROGRAM  */                OUTI0220
     2  42X,18(4H****)///20X,9A10)                                      OUTI0230
      CMA = CMAC = SPA / 2.0                                            OUTI0240
   30 WRITE(6, 40 ) ARE,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,  OUTI0250
     1              ARATIO,XCG,XC                                      OUTI0260
   40 FORMAT(1HC///54X,4HINPUT //54X,5HINPUT /                          OUTI0270
     1       41X,6HAREA =,2E15.6 / 41X,6HSPAN =,2E15.6 /                OUTI0280
     2       41X,6HCREF =,2E15.6 / 42X,5HXMC =,2E15.6 /                 OUTI0290
     3       41X,6HCMAC =,2E15.6 / 39X,8HARATIO =,2E15.6 /              OUTI0300
     4       42X,5HXCG =,2E15.6)                                        OUTI0310
      WRITE(6, 50 ) NBOWS,NBOWS,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JETOUTI0320
     1              IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX                 OUTI0330
   50 FORMAT(1HC/ 48X,7HNBOWS =,I3,7X,I3 / 47X,8HNCASES =,I3,7X,I3 /    OUTI0340
     1            48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /   OUTI0350
     2            47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /  OUTI0360
     3            47X,8HIHINGE =,I3,7X,I3 ///                          OUTI0370
     4            43X,25HNUMBER OF WING ELEMENTS =,I4 /                OUTI0380
     5            43X,25HNUMBER OF JET ELEMENTS  =,I4 /                OUTI0390
     6            42X,26HTOTAL NUMBER OF ELEMENTS =,I4)                OUTI0400
      DO 60 J = 1                                                       OUTI0410
      JJ = NWT                                                          OUTI0420
C                                                                       OUTI0430
C  PRINT FUNDAMENTAL CASE HEADER                                        OUTI0440
      WRITE(6, 70 ) LCASE                                               OUTI0450
   70 FORMAT(1H1,23X,1HF,19(4H****)/                                    OUTI0460
     1       24X,54H  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,OUTI0470
     2       17H FUNDAMENTAL CASE,I3,3H  F/24X,1HF,19(4H****))         OUTI0480
      ILINES = 3                                                        OUTI0490
      DO 260 K = 1,NBOWS                                                OUTI0500
C                                                                       OUTI0510
C  PRINT SECTIONAL DATA                                                 OUTI0520
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)OUTI0530
```

```
   80 FORMAT(1HO,11H** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,   OUTI0540
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6, OUTI0550
     2 2X,7HTANLE =,F10.6)                                              OUTI0560
C                                                                       OUTI0570
C  PRINT CHORDWISE DATA ON WING                                         OUTI0580
      NWK = NW(K)                                                       OUTI0590
      WRITE(6, 90 ) NWK,TST(K,LCASE),HL(K,LCASE),THS(K,LCASE)           OUTI0600
   90 FORMAT(1HO,21HWING ELEMENTS   NW =,I3,5X, 7HTWIST =,F10.6,5X,     OUTI0610
     1       4HHL =,F10.4,5X, 8HTHETA S =,F10.6)                        OUTI0620
      WRITE(6, 100 ) (XR(JJ+L),L=1,NWK)                                 OUTI0630
  100 FORMAT(1H ,14X,2HXR,10F11.6 / 17X,10F11.6)                        OUTI0640
      IF(LCASE .GT. 1) GO TO 130                                        OUTI0650
      WRITE(6, 110 ) (XI(JJ+L),L=1,NWK)                                 OUTI0660
  110 FORMAT(1H ,14X, 2HXI,10F11.6/17X,10F11.6)                         OUTI0670
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NWK)                                OUTI0680
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)                         OUTI0690
  130 IF(ICT(K) .EQ. 0) GO TO 150                                       OUTI0700
      ICK = ICT(K)                                                      OUTI0710
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)                                OUTI0720
  140 FORMAT(1H ,10X,4HCAMB,10F11.6/17X,10F11.6)                        OUTI0730
  150 WRITE(6, 160 ) (EPS(JJ+L,LCASE),L=1,NWK)                          OUTI0740
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)                         OUTI0750
      WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NWK)                         OUTI0760
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)                        OUTI0770
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NWK)                              OUTI0780
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,4X)/17X,10(3X,I2,4X))              OUTI0790
      J = J + NWK                                                       OUTI0800
      IL = 1                                                            OUTI0810
      IF(NWK .GT. 9) IL = 2                                             OUTI0820
      ILINES = ILINES + 4 + 4*IL                                        OUTI0830
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL                           OUTI0840
C                                                                       OUTI0850
C  PRINT CHORDWISE DATA ON JET                                          OUTI0860
      NJK = NJ(K)                                                       OUTI0870
      IF(NJK .GT. 0) GO TO 200                                          OUTI0880
      WRITE(6, 190 )                                                    OUTI0890
  190 FORMAT(1H ,8X,19HTHIS ROW HAS NO JET)                             OUTI0900
      ILINES = ILINES + 1                                               OUTI0910
      GO TO 230                                                         OUTI0920
  200 WRITE(6, 210 ) NJK,DJ(K),DJ(K),ACTE(K),THETA(K,LCASE)            OUTI0930
  210 FORMAT(1HO,1X,22HJET ELEMENTS   NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,F10.6,5X,4HACTE =,F10.6,5X,7HTHETA =,F10.6)                          OUTI0940
      WRITE(6, 100 ) (XR(JJ+L),L=1,NJK)                                 OUTI0950
      IF(LCASE .GT. 1) GO TO 225                                        OUTI0960
      WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)                                 OUTI0970
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)                                OUTI0980
  220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)                         OUTI0990
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)                              OUTI1000
      JJ = JJ + NJK                                                     OUTI1010
      IL = 1                                                            OUTI1020
      IF(NJK .EQ. 10) IL = 2                                            OUTI1030
      ILINES = ILINES + 3 + IL                                          OUTI1040
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL                           OUTI1050
  230 IF(K .EQ. NBOWS) GO TO 260                                        OUTI1060
      NWK1 = NW(K+1)                                                    OUTI1070
      IL = 1                                                            OUTI1080
      IF(NWK1 .GT. 9) IL = 2                                            OUTI1090
      NEXT = 4 + 4*IL                                                   OUTI1100
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL                               OUTI1110
      NJK1 = NJ(K+1)                                                    OUTI1120
      IL = 1                                                            OUTI1130
      IF(NJK1 .EQ. 10) IL = 2                                           OUTI1140
      NEXT = NEXT + 1                                                   OUTI1150
      IF(NJK1 .EQ. 0) GO TO 240                                         OUTI1160
      NEXT = NEXT + 1 + 3*IL                                            OUTI1170
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL                               OUTI1180
  240 IF(55-ILINES) .GE. NEXT) GO TO 260                                OUTI1190
      WRITE(6, 250 )                                                    OUTI1200
  250 FORMAT(1H1)                                                       OUTI1210
      ILINES = 1                                                        OUTI1220
  260 CONTINUE                                                          OUTI1230
      RETURN                                                            OUTI1240
      END                                                               OUTI1250
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK INCOMP                                                            INC00010
      SUBROUTINE INCOMP(NCASES,IR)                                      INC00020
C                                                                       INC00030
C  THIS SUBROUTINE READS IN THE COMPOSITE CASE REQUIREMENTS             INC00040
C  WHICH DEFINE THE FUNDAMENTAL CASES AND THEIR DEFLECTION MAGNITUDE    INC00050
C  FOR SUPERPOSITION IN UP TO 24 COMBINATIONS                           INC00060
C                                                                       INC00070
      COMMON/COMPOS/FACTOR(10,24),NCC                                   INC00080
      DIMENSION FUNNY(10),NOE(10)                                       INC00090
C                                                                       INC00100
C  INITIALIZE THE ARRAY OF FUNDAMENTAL CASE DEFLECTIONS                 INC00110
      DO 20 M = 1,24                                                    INC00120
      DO 20 N = 1,10                                                    INC00130
      FACTOR(N,M) = 0.00                                                INC00140
   10 CONTINUE                                                          INC00150
   20 CONTINUE                                                          INC00160
C                                                                       INC00170
C  READ THE COMPOSITE CASE DATA, CONSISTING OF FUNDAMENTAL CASE         INC00180
C  DEFLECTIONS, IN DEGREES                                              INC00190
      NCC = 1                                                           INC00200
   30 NCC = NCC + 1                                                     INC00210
      READ(5, 40 ) (NOE(N),FUNNY(N),N=1,10)                            INC00220
   40 FORMAT(10(I2,F4.4))                                               INC00230
      IF(EOF(5) .NE. 0) GO TO 130                                       INC00240
C  CHECK THE VALIDITY OF THE DATA                                       INC00250
   50 IF(NOE(1) .GT. 12) GO TO 100                                      INC00260
      IF(NCC .GT. 24) GO TO 110                                         INC00270
      DO 90 N = 1,10                                                    INC00280
      IF(NOE(N) .GT. NCASES) GO TO 70                                   INC00290
      IF(NOE(N) .LT. 1) GO TO 90                                        INC00300
C  THE DATA IS OK.  DEFINE FACTOR.                                      INC00310
      NON = NOE(N)                                                      INC00320
   60 FACTOR(NON,NCC) = FUNNY(N)                                        INC00330
      GO TO 90                                                          INC00340
   70 WRITE(6, 80 )                                                     INC00350
   80 FORMAT(1H0,22X,25HAN INCORRECT COMPOSITE CASE INPUT VALUE HAS BEEN INC00360
     1 FOUND.  IT WILL BE IGNORED.)                                     INC00370
   90 CONTINUE                                                          INC00380
      GO TO 30                                                          INC00390
C  THE END OF THE INPUT DATA HAS BEEN REACHED                           INC00400
  100 NCC = NCC - 1                                                     INC00410
      IF(NCC .GT. 24) NCC = 24                                          INC00420
      IR = 1                                                            INC00430
      RETURN                                                            INC00440
C  TOO MANY COMPOSITE CASES HAVE BEEN REQUESTED.  READ ON UNTIL AN END  INC00450
C  CARD IS FOUND.                                                       INC00460
  110 WRITE(6, 120 )                                                    INC00470
  120 FORMAT(1H0,20X,47HMORE THAN 24 COMPOSITE CASES HAVE BEEN INPUT.    INC00480
     1       34HSUBSEQUENT INPUTS WILL BE IGNORED.)                     INC00490
      GO TO 30                                                          INC00500
C                                                                       INC00510
C  AN END OF FILE HAS BEEN READ.  PRINT A MESSAGE AND QUIT.             INC00520
  130 WRITE(6, 140 )                                                    INC00530
  140 FORMAT(1H1/// 31X, 35HAN END OF FILE HAS BEEN READ DURING,        INC00540
     1       21H COMPOSITE CASE INPUT)                                  INC00550
      IR = 2                                                            INC00560
      RETURN                                                            INC00570
      END                                                               INC00580
```

```
*DECK BLOWIN                                                            BLOW0010
      SUBROUTINE BLOWIN(JETFLG,IR)                                      BLOW0020
C                                                                       BLOW0030
C  THIS SUBROUTINE READS THE SECTIONAL JET BLOWING RATES                BLOW0040
C  CMU(K) = J / (Q * CHORD(K))                                          BLOW0050
C                                                                       BLOW0060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BLOW0070
      COMMON/JCASE/CMU(40),CMUP(40)                                     BLOW0080
      DIMENSION DCMU(40)                                                BLOW0090
C                                                                       BLOW0100
      IF(JETFLG .NE. 0) GO TO 30                                        BLOW0110
C  READ THE CMU DATA ONLY FOR THOSE SECTIONS WHICH HAVE A JET           BLOW0120
      READ(5, 10 ) (DCMU(K),K=1,NROWSJ)                                 BLOW0130
   10 FORMAT(8F10.5)                                                    BLOW0140
      IF(EOF(5) .NE. 0) GO TO 40                                        BLOW0150
   20 IF(DCMU(1) .LT. 0.0) GO TO 30                                     BLOW0160
      IR = 2                                                            BLOW0170
      RETURN                                                            BLOW0180
C                                                                       BLOW0190
C  REARRANGE THE DATA INTO THE PROPER SEQUENCE                          BLOW0200
   30 KP = 0                                                            BLOW0210
      DO 50 K = 1,NROWS                                                 BLOW0220
   40 CMU(K) = 0.0                                                      BLOW0230
      IF(NJ(K) .EQ. 0) GO TO 50                                         BLOW0240
      KP = KP + 1                                                       BLOW0250
      CMU(K) = DCMU(KP)                                                 BLOW0260
   50 CONTINUE                                                          BLOW0270
      IR = 1                                                            BLOW0280
      RETURN                                                            BLOW0290
C                                                                       BLOW0300
C  AN END OF FILE HAS BEEN READ.  THIS RUN IS COMPLETELY FINISHED.      BLOW0310
   60 WRITE(6, 70 )                                                     BLOW0320
   70 FORMAT(1H1///41X,37HNO MORE CMU CASES HAVE BEEN REQUESTED)        BLOW0330
      IR = 3                                                            BLOW0340
      RETURN                                                            BLOW0350
      END                                                               BLOW0360
```

```
*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/X(40),CHORD(40),DELTA(40),XW(40,8),XI(40,1,DEL(40),  BOXJ0070
     1                 2(40),KK(40),ITYPE(40)                           BOXJ0080
      COMMON/JCASE/CMU(40),CMUP(40)                                     BOXJ0090
C                                                                       BOXJ0100
C  COMPUTE THE NEW CMUP AND SAVE THE OLD VALUES AS CMUPP                BOXJ0110
   10 NEWMAX = NMAX                                                     BOXJ0120
      ICOUNT = 0                                                        BOXJ0130
      DO 30 K = 1,NROWS                                                 BOXJ0140
      CMUPP(K) = CMUP(K)                                                BOXJ0150
      IF(NJ(K) .EQ. 0) GO TO 30                                         BOXJ0160
      IF(CMU(K) .LT. 0.0001) GO TO 20                                   BOXJ0170
      CMUP(K) = 2.00 / (CHORD(K)*CMU(K))                                BOXJ0180
      GO TO 30                                                          BOXJ0190
   20 ICOUNT = ICOUNT + 1                                               BOXJ0200
      CMUP(K) = 0.00                                                    BOXJ0210
   30 CONTINUE                                                          BOXJ0220
      WRITE(6, 40 ) (K,CMU(K),K=1,NROWS)                                BOXJ0230
   40 FORMAT(1H1,42X,1(/4H****)/ 41X,                                   BOXJ0240
     1       40H*  SECTIONAL JET BLOWING COEFFICIENTS  */41X,1(4H****)// BOXJ0250
     2       53X,3HROW,5X,3HCMU,4X/(53X,I2,F12.6))                      BOXJ0260
      IF(ICOUNT .EQ. 0) GO TO 50                                        BOXJ0270
      IF(ICOUNT .LT. NROWSJ) GO TO 60                                   BOXJ0280
      NEWMAX = NWT                                                      BOXJ0290
   50 IR = 1                                                            BOXJ0300
      RETURN                                                            BOXJ0310
C  AN ERROR HAS OCCURED.  PRINT A MESSAGE AND TRY AGAIN.                BOXJ0320
   60 WRITE(6, 70 )                                                     BOXJ0330
   70 FORMAT(1H0,  43X,35HA ZERO VALUE OF CMU HAS BEEN INPUT..          BOXJ0340
     1       33H  THIS CMU CASE HAS BEEN IGNORED.)                      BOXJ0350
      IR = 2                                                            BOXJ0360
      RETURN                                                            BOXJ0370
      END                                                               BOXJ0380
```

```
*DECK STAGE2                                                            STG20010
      OVERLAY(OVLY,2,0)                                                 STG20020
      PROGRAM STAGE2                                                    STG20030
C                                                                       STG20040
C  THIS PROGRAM CONTROLS THE FORMATION AND SOLUTION OF                  STG20050
C  THE SYSTEM OF LINEAR EQUATIONS                                       STG20060
C                                                                       STG20070
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALEA,LOGIC,IR                      STG20080
C                                                                       STG20090
C  FORM THE SYSTEM OF LINEAR EQUATIONS                                  STG20100
   10 CALL OVERLAY(4HOVLY,2,1)                                          STG20110
C                                                                       STG20120
C  SOLVE THE SYSTEM OF LINEAR EQUATIONS                                 STG20130
   20 CALL OVERLAY(4HOVLY,2,2)                                          STG20140
      IF(IR .EQ. 2) GO TO 30                                            STG20150
C                                                                       STG20160
C  THE SOLUTION HAS NOW BEEN COMPLETED.  RETURN NORMALLY TO MAIN.       STG20170
C  *** HALLELUJIAH ***                                                  STG20180
      IR = 1                                                            STG20190
      GO TO 40                                                          STG20200
C                                                                       STG20210
C**********************************************************************STG20220
C  A FATAL ERROR HAS OCCURED.  RETURN ABNORMALLY TO MAIN.               STG20230
   30 IR = 2                                                            STG20240
   40 CONTINUE                                                          STG20250
C**********************************************************************STG20260
C                                                                       STG20270
      END                                                               STG20280
```

C7

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK ST020                                                          ST2D0010
      OVERLAY(OVLY,2,1)                                              ST2D0020
      PROGRAM ST020                                                  ST2D0030
C                                                                    ST2D0040
C  THIS PROGRAM CONTROLS THE CALCULATION OF ALL EVD DOWNWASH         ST2D0050
C  INFLUENCE COEFFICIENTS AND THE FORMATION OF THE LEFT AND RIGHT SIDE ST2D0060
C  MATRICES.                                                         ST2D0070
C                                                                    ST2D0080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE         ST2D0090
      COMMON/MARK/NROWS,NPOWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) ST2D0100
      COMMON/SCRTCH/ NEWMAX,NEWCMU,NCALFA,LOGIC,TM                   ST2D0110
      DIMENSION W(810)                                               ST2D0120
C                                                                    ST2D0130
C  IF THIS IS A NEW CMU CASE, AUGMENT THE EXISTING DOWNWASH MATRIX ROWS ST2D0140
C  ON THE JET                                                        ST2D0150
      ISIZE = NEWMAX                                                 ST2D0160
      IF(NEWCMU .EQ. 1) GO TO 10                                     ST2D0170
      IF(NEWMAX .GT. NWT) CALL SHUFL2(W,ISIZE,NEWMAX)                ST2D0180
      GO TO 30                                                       ST2D0190
C                                                                    ST2D0200
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENTS AT ALL CONTROL POINTS ST2D0210
C  DUE TO ALL TRIANGULAR, LEADING EDGE AND FAR-JET EVD ELEMENTS      ST2D0220
   10 ISIZE = NMAX                                                   ST2D0230
      IF((LOGIC .EQ. 2) .AND. (ISYMM .GT. 0)) GO TO 30               ST2D0240
      CALL DWNWSH(W,ISIZE)                                           ST2D0250
C                                                                    ST2D0260
C  AUGMENT THE MATRIX ROWS FOR CONTROL POINTS ON THE JET.            ST2D0270
C  NOTE THAT THIS MUST BE DONE EVEN THOUGH CMU MAY BE 0.0,           ST2D0280
C  IN ORDER TO PREPARE FOR FUTURE NONZERO CMU CASES.                 ST2D0290
   20 IF(NMAX .GT. NWT) CALL SHUFL1(W,ISIZE)                         ST2D0300
C                                                                    ST2D0310
C  DEVELOP THE RIGHT SIDE COLUMN MATRIX                              ST2D0320
   30 ISIZE = NEWMAX                                                 ST2D0330
      DO 80 N = 1,NCASES                                             ST2D0340
      LCASE = N                                                      ST2D0350
C                                                                    ST2D0360
C  DEFINE THE LCASE COLUMN, NOT INCLUDING THE INFLUENCE OF ANY HINGES ST2D0370
   40 CALL COLUM1(LCASE)                                             ST2D0380
C                                                                    ST2D0390
C  COMPUTE THE HINGE DOWNWASH INFLUENCE FACTORS                      ST2D0400
      IF((LCASE .EQ. 1) .OR. (IHINGE .EQ. 0)) GO TO 80               ST2D0410
      DO 50 I = 1,NEWMAX                                             ST2D0420
      W(I) = 0.00                                                    ST2D0430
   50 CONTINUE                                                       ST2D0440
   60 CALL HINGE(W,ISIZE,NEWMAX,LCASE)                               ST2D0450
C                                                                    ST2D0460
C  MODIFY THE LCASE COLUMN TO INCLUDE THE INFLUENCE OF ALL HINGES    ST2D0470
   70 CALL COLUM2(W,ISIZE,NEWMAX,LCASE)                              ST2D0480
   80 CONTINUE                                                       ST2D0490
C                                                                    ST2D0500
C  THE MATRIX DEVELOPMENT IS NOW COMPLETE.                           ST2D0510
C  PUT THE MATRIX SYSTEM IN THE PROPER FORM FOR SOLUTION.            ST2D0520
   90 ISIZE = NEWMAX + NCASES                                        ST2D0530
  100 CALL PREP(W,ISIZE,NEWMAX)                                      ST2D0540
      CONTINUE                                                       ST2D0550
      END                                                            ST2D0560
```

```
*DECK DWNWSH                                                         DWNW0010
      SUBROUTINE DWNWSH(W,ISIZE)                                     DWNW0020
C                                                                    DWNW0030
C  THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENT MATRIX. DWNW0040
C  THE MATRIX IS STORED ON THE DIRECT ACCESS SCRATCH FILE.           DWNW0050
C                                                                    DWNW0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE         DWNW0070
      COMMON/MARK/NROWS,NPOWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) DWNW0080
      COMMON/GEOM1/YY(43),CHORD(40),DELTA(40),XR(600),XT(600),DEL(600), DWNW0090
     1    D(43),KK(600),ITYPE(600)                                   DWNW0100
      DIMENSION W(ISIZE)                                             DWNW0110
C                                                                    DWNW0120
C  COMPUTE ALL THE DOWNWASH COEFFICIENTS                             DWNW0130
C                                                                    DWNW0140
      IWRITE = 0                                                     DWNW0150
      IF(IPRINT .LT. 0) WRITE(6, 10 )                                DWNW0160
   10 FORMAT(1H1,38X,44HWING - DUE - TO - WING - AND - JET  DOWNWASH/) DWNW0170
C                                                                    DWNW0180
C  CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET             DWNW0190
      DO 190 I = 1,NMAX                                              DWNW0200
      IWRITE = IWRITE + 1                                            DWNW0210
      KI = KK(I)                                                     DWNW0220
C                                                                    DWNW0230
C  CYCLE THE VORTEX POINTS ON THE WING AND JET                       DWNW0240
      DO 150 J = 1,NMAX                                              DWNW0250
C  COMPUTE THE GENERAL GEOMETRIC PARAMETERS                          DWNW0260
      KJ = KK(J)                                                     DWNW0270
   20 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XT(J)                      DWNW0280
      YY = Y(KI) - Y(KJ)                                             DWNW0290
      IT = ITYPE(J)/10                                               DWNW0300
C                                                                    DWNW0310
C  DECIDE WHICH EVD TYPE TO USE.  ONLY THE TRIANGULAR PART OF HINGES DWNW0320
C  WILL BE CONSIDERED AT THIS TIME.                                  DWNW0330
      GO TO ( 50, 80, 110, 120, 50 ), IT                             DWNW0340
   30 WRITE(6, 40 ) J,IT                                             DWNW0350
   40 FORMAT(1H ,35X,21H** WARNING ** ELEMENT,I4,10H HAS AN ITYPE VALUE, DWNW0360
     1      3H OF,I3/39X,42HAN EQUIVALENT TRIANGULAR DOWNWASH WAS USED) DWNW0370
C                                                                    DWNW0380
C  REGULAR TRIANGULAR EVD  (INCLUDING TRIANGULAR PART OF HINGE EVD)  DWNW0390
   50 D1 = DEL(J-1) * CHORD(KJ)                                      DWNW0400
      INI = IW(KJ) + NW(KJ) - 1                                      DWNW0410
      IF( J .EQ. IJ(KJ)) D1 = DEL(IN1) * CHORD(KJ)                   DWNW0420
      D2 = DEL(J) * CHORD(KJ)                                        DWNW0430
   60 W(I) = EVD1(X,YY,D1,D2,DELTA(KJ))                              DWNW0440
C  SUPERIMPOSE THE DOWNWASH FROM THE LEFT SIDE OF THE WING IF THIS IS A DWNW0450
C  SYMETRIC OR ANTI-SYMETRIC CASE                                    DWNW0460
      IF(ISYMM .GT. 0) GO TO 150                                     DWNW0470
      YY = Y(KI) + Y(KJ)                                             DWNW0480
   70 WDUMMY = EVD1(X,YY,D1,D2,DELTA(KJ))                            DWNW0490
      IF(ISYMM .LT. 0)WDUMMY = -WDUMMY                               DWNW0500
      GO TO 140                                                      DWNW0510
C                                                                    DWNW0520
C  LEADING EDGE EVD                                                  DWNW0530
```

```
   80 D2 = DEL(J) * CHORD(KJ)                                        DWNW0540
   90 W(I) = EVD2(X,YY,D2,DELTA(KJ))                                 DWNW0550
C  SUPERIMPOSE DOWNWASH                                              DWNW0560
      IF(ISYMM .GT. 0) GO TO 150                                     DWNW0570
      YY = Y(KI) + Y(KJ)                                             DWNW0580
  100 WDUMMY = EVD2(X,YY,D2,DELTA(KJ))                              DWNW0590
      IF(ISYMM .LT. 0) WDUMMY = -WDUMMY                              DWNW0600
      GO TO 140                                                      DWNW0610
C                                                                    DWNW0620
C  FAR - JET EVD                                                     DWNW0630
  110 D1 = DEL(J-1) * CHORD(KJ)                                      DWNW0640
  120 W(I) = EVD3(X,YY,D1,D(KJ),DELTA(KJ))                           DWNW0650
C  SUPERIMPOSE DOWNWASH                                              DWNW0660
      IF(ISYMM .GT. 0) GO TO 150                                     DWNW0670
      YY = Y(KI) + Y(KJ)                                             DWNW0680
  130 WDUMMY = EVD3(X,YY,D1,D(KJ),DELTA(KJ))                         DWNW0690
      IF(ISYMM .LT. 0) WDUMMY = -WDUMMY                              DWNW0700
  140 W(I) = W(I) + WDUMMY                                           DWNW0710
  150 CONTINUE                                                       DWNW0720
C                                                                    DWNW0730
C  STORE THE DOWNWASH AT CONTROL POINT I ON THE DIRECT ACCESS UNIT   DWNW0740
C                                                                    DWNW0750
  160 CALL WRITMS(1,W,ISIZE,IWRITE)                                  DWNW0760
      IF(IPRINT .GE. 0) GO TO 190                                    DWNW0770
      IF(I .EQ. NWT+1) WRITE(6, 170 )                                DWNW0780
  170 FORMAT(1H1,38X,43HJET - DUE - TO - WING - AND - JET  DOWNWASH/) DWNW0790
      WRITE(6, 180 ) I,W                                             DWNW0800
  180 FORMAT(1H ,55X,13HMATRIX ROW,I4,6G(/1X,10E13.5))               DWNW0810
  190 CONTINUE                                                       DWNW0820
      RETURN                                                         DWNW0830
C                                                                    DWNW0840
C  THE DIRECT ACCESS UNIT NOW CONTAINS THE FOLLOWING -               DWNW0850
C      WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS  (NWT RECORDS) DWNW0860
C      JET -DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS  (NJT RECORDS) DWNW0870
      END                                                            DWNW0880
```

```
*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 90                       EVD10150
   10 XPD = X + D1                                                   EVD10160
      XMD = X - D2                                                   EVD10170
   20 RDP = R(X,YPD)                                                 EVD10180
      R1D = R(XPD,YMD)                                               EVD10190
      R2P = R(XMD,YMD)                                               EVD10200
   30 RDPP = R(X,YPD)                                                EVD10210
      R1PP = R(XPD,YPD)                                              EVD10220
      R2PP = R(XMD,YPD)                                              EVD10230
C                                                                    EVD10240
C  CALCULATE THE DOWNWASH                                            EVD10250
   40 PART2 = (XPD/D1) * ((R1D-RDP)/YMD - (R1PP-RDPP)/YPD)           EVD10260
      PART3 = (XMD/D2) * ((R2P-R3D)/YMD - (R2PP-RDPP)/YPD)           EVD10270
      YRATIO = (YPD+RDPP) / (YMD+RDP)                                EVD10280
   50 PART4 = (XPD/D1) * ALOG((YMD+R1D)/(YPD+R1PP)) + YRATIO         EVD10290
      PART5 = (XMD/D2) * ALOG((YMD+R2D)/(YPD+R2PP)) + YRATIO         EVD10300
   60 PART6 = YMD * ALOG((XPD+R1D)/(X+RDP))                          EVD10310
      PART7 = YPD * ALOG((XPD+R1PP)/(X+RDPP))                        EVD10320
   70 PART8 = YMD * ALOG((XMD+R2D)/(X+RDP))                          EVD10330
      PART9 = YPD * ALOG((XMD+R2PP)/(X+RDPP))                        EVD10340
   80 EVD1 = -(PART1 + (PART2 + PART3) - 2.0*(PART4 + PART5)         EVD10350
     1    -((PART6 - PART7)/D1 + (PART8 - PART9)/D2)) / 25.13274     EVD10360
      RETURN                                                         EVD10370
   90 EVD1 = -PART1 / 12.56673                                       EVD10380
  100 RETURN                                                         EVD10390
      END                                                            EVD10400
```

C8

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK EVD2
      FUNCTION EVD2(X,Y,DEL,DELTA)                                      EVD20010
C                                                                       EVD20020
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y               EVD20030
C  DUE TO A LEADING EDGE EVD ELEMENT WITH UNIT AVERAGE VORTICITY,       EVD20040
C  LOCATED AT THE ORIGIN 0,0                                            EVD20050
C                                                                       EVD20060
      DIMENSION S1(9),FACTOR(9)                                         EVD20070
C                                                                       EVD20080
      R(A,B) = SQRT(A**A + B**B)                                        EVD20090
      G(A) = 1.00/SQRT(A) - A                                           EVD20100
      S(A) = ABS(A) / A                                                 EVD20110
C                                                                       EVD20120
      DATA S1/-0.9681602,-0.8360311,-0.6133714,-0.3242534,0.0,          EVD20130
     1       0.3242534,0.6133714,0.8360311,0.9681602/                   EVD20140
      DATA FACTOR/0.0812744,0.1804882,0.2604127,0.3123471,0.3302394,    EVD20150
     1            0.3123471,0.2604127,0.1804882,0.0812744/              EVD20160
C                                                                       EVD20170
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS                           EVD20180
   10 XR = X / DEL                                                      EVD20190
      YR = Y / DEL                                                      EVD20200
      DR = DELTA / DEL                                                  EVD20210
      YPD = YR + DR                                                     EVD20220
      YMD = YR - DR                                                     EVD20230
   20 IF(XR .GT. 100.0) GO TO 280                                       EVD20240
      XM1 = XR - 1.0E                                                   EVD20250
   30 RCP = R(XR,YPD)                                                   EVD20260
   40 RCPP = R(XR,YMD)                                                  EVD20270
C                                                                       EVD20280
C  CALCULATE RK(XR)                                                     EVD20290
      IF(ABS(XR) .LT. 1.0E-04) GO TO 100                                EVD20300
      IF(ABS(XM1) .LT. 1.0E-04) GO TO 85                                EVD20310
   50 PART = ALOG(ABS(YM1/YP1))                                         EVD20320
      PART1 = XR * PART + 1.00                                          EVD20330
      IF(XR .GT. 0.00) GO TO 70                                         EVD20340
   60 SQX = SQRT(-XR)                                                   EVD20350
      RK = -2.00 / SQX * ATAN(1.00/SQX) + PART1                         EVD20360
      GO TO 90                                                          EVD20370
   70 SQX = SQRT(XR)                                                    EVD20380
      RK = -ALOG(ABS((1.00-SQX) /(1.00+SQX))) / SQX + PART1             EVD20390
      GO TO 90                                                          EVD20400
   85 RK = 2.396294                                                     EVD20410
C                                                                       EVD20420
C  CALCULATE P(XR)                                                      EVD20430
   90 PART2 = RCP/YMD - RCPP/YPD                                        EVD20440
      P = PART2 - RK                                                    EVD20450
      GO TO 110                                                         EVD20460
  100 P = 0.00                                                          EVD20470
C                                                                       EVD20480
C  CALCULATE E(YR) BY GAUSSIAN INTEGRATION.                             EVD20490
  110 IF(XR .GT. 0.01 .AND. XR .LT. 1.00) GO TO 150                     EVD20500
C  XR IS NOT WITHIN THE X DIMENSIONS OF THE ELEMENT.                    EVD20510
      E = 0.00                                                          EVD20520
      DO 140 N = 1,9                                                    EVD20530
  120 SR = (S1(N)+1.00) / 2.00                                          EVD20540
      XMS = XR - SR                                                     EVD20550
      GS = G(SR)                                                        EVD20560
  130 PART3 = (GS*(R(XMS,YMD)-RCP))/YMD - (GS*(R(XMS,YPD)-RCPP))/YPD    EVD20570
      E = E + FACTOR(N) * PART3 / XMS                                   EVD20580
  140 CONTINUE                                                          EVD20590
      E = 0.50 * E                                                      EVD20600
      GO TO 230                                                         EVD20610
C                                                                       EVD20620
C  XR IS WITHIN THE X DIMENSIONS OF THE ELEMENT.  CALCULATE EC.         EVD20630
  150 EC = 0.00                                                         EVD20640
      GPX = 0.00                                                        EVD20650
      GPPX = 0.00                                                       EVD20660
      IF(XR .LT. 1.0E-04) GO TO 170                                     EVD20670
      GX = G(XR)                                                        EVD20680
      GPX = GX - (ABS(YMD) - RCP)                                       EVD20690
      GPPX = GX + (ABS(YPD) - RCPP)                                     EVD20700
      IF(XM1 .GT. -1.0E-04) GO TO 170                                   EVD20710
  160 EC = -(GPX/YMD - GPPX/YPD) + PART                                 EVD20720
C  CALCULATE E1 BY GAUSSIAN INTEGRATION.                                EVD20730
  170 E1 = 0.00                                                         EVD20740
      DO 200 N = 1,9                                                    EVD20750
  180 SR = (S1(N)+1.00) / 2.00                                          EVD20760
      XMS = XR - SR                                                     EVD20770
  190 IF(ABS(XMS) .LT. 1.0E-04) GO TO 220                               EVD20780
      GS = G(SR)                                                        EVD20790
  200 PART4 = (GS*(R(XMS,YMD)-RCP) - GPX) / XMS                         EVD20800
  210 PART5 = (GS*(R(XMS,YPD)-RCPP) - GPPX) / XMS                       EVD20810
      PART6 = PART4/YMD - PART5/YPD                                     EVD20820
      GO TO 240                                                         EVD20830
  220 PART4 = S(YMD) - S(YPD) - PART2                                   EVD20840
      PART5 = 1.00 + 1.00 / (SQRT(XR))**3                               EVD20850
  230 PART6 = PART4 + PART5                                             EVD20860
  240 E1 = E1 + FACTOR(N) * PART6                                       EVD20870
  250 CONTINUE                                                          EVD20880
      E = E1 * 0.50 + E1                                                EVD20890
C                                                                       EVD20900
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT.                        EVD20910
  270 EVD2 = -(1.00 * (1.00/YMD - 1.00/YPD) + E) / 12.56637             EVD20920
      RETURN                                                            EVD20930
  280 EVD2 = -(1.00/YMD - 1.00/YPD) / 6.283185                          EVD20940
      RETURN                                                            EVD20950
      END                                                               EVD20960
```

```
*DECK EVD3
      FUNCTION EVD3(X,Y,DEL,D,DELTA)                                    EVD30010
C                                                                       EVD30020
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y DUE TO A      EVD30030
C  FAR-JET EVD ELEMENT WITH UNIT PEAK VORTICITY, LOCATED AT THE ORIGIN  EVD30040
C                                                                       EVD30050
      R(A,B) = SQRT(A**A + B**B)                                        EVD30060
C                                                                       EVD30070
C  CALCULATE THE BASIC GEOMETRICAL PARAMETERS                           EVD30080
      IF(Y .LT. 0.0) Y = -Y                                             EVD30090
      YMD = Y - DELTA                                                   EVD30100
      YPD = Y + DELTA                                                   EVD30110
   10 XPD = X + D                                                       EVD30120
   25 PART1 = (DEL/2.00) + D) * (1.00/YMD - 1.00/YPD)                   EVD30130
C  CHECK ON INFINITY                                                    EVD30140
      IF((XPD/YMD)**2 .GT. 1.0E06) GO TO 160                            EVD30150
      XPD1 = X + DEL                                                    EVD30160
      XD = X/DEL                                                        EVD30170
      RCP = R(X,YMD)                                                    EVD30180
      RCPP = R(X,YPD)                                                   EVD30190
   30 R1P = R(XPD1,YMD)                                                 EVD30200
      R1PP = R(XPD1,YPD)                                                EVD30210
      RDP = R(XPD,YMD)                                                  EVD30220
      RDPP = R(XPD,YPD)                                                 EVD30230
C                                                                       EVD30240
C  CALCULATE E0                                                         EVD30250
   40 PART2 = RDP/YMD - RDPP/YPD                                        EVD30260
      PART3 = 0.5E 00 * (XD+1.00) * ((R1P-RDP)/YMD - (R1PP-RDPP)/YPD)   EVD30270
   50 PART4 = ALOG(ABS(((YMD+R1P)/(YPD+R1PP)) * ((YPD+RDPP)/(YMD+RDP))))EVD30280
      PART5 = YMD/DEL * ALOG((XPD1+R1P)/(X+RCP))                        EVD30290
   60 PART6 = YPD/DEL * ALOG((XPD1+R1PP)/(X+RDPP))                      EVD30300
   75 E0 = PART1 - 0.50*PART2 + PART3 - (XD+1.00)*PART4                 EVD30310
     1      - 0.50*(PART5-PART6)                                       EVD30320
C                                                                       EVD30330
C  CALCULATE E1                                                         EVD30340
      IF(ABS(XPD/D) .LT. 1.0E-02) GO TO 130                             EVD30350
C  X IS NOT NEAR -D                                                     EVD30360
      Q = D/XPD                                                         EVD30370
   80 PART1 = -D * Q * (1.00/YMD - 1.00/YPD)                            EVD30380
      PART2 = D * PART2                                                 EVD30390
   90 PART3 = D * Q * ALOG(ABS((YMD+RCP)/(YPD+RDPP)))                   EVD30400
  100 PART4 = YMD/RDP*ALOG(ABS((-XPD*RDP+((RCP+RDP)/D))/(RDP-XPD)))    EVD30410
  110 PART5 = YPD/RDPP*ALOG(ABS((-XPD*RDPP+((RDPP+RDP)/D))/(RDPP-XPD)))EVD30420
  120 E1 = PART1 + PART2 - PART3 + Q * D * (PART4 - PART5)              EVD30430
      GO TO 150                                                         EVD30440
C  X IS NEAR -D                                                         EVD30450
  130 X = -D                                                            EVD30460
      RCP = R(X,YMD)                                                    EVD30470
      RDPP = R(X,YPD)                                                   EVD30480
      PART2 = RDP/YMD - RDPP/YPD                                        EVD30490
      PART3 = (X/YMD)**2 + ALOG(ABS((YMD+RCP)/X))                       EVD30500
      PART4 = (X/YPD)**2 + ALOG(ABS((YPD+RDPP)/X))                      EVD30510
  140 E1 = -0.5E 00*PART2 - 0.5E 00*(PART3 - PART4)                     EVD30520
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT                        EVD30530
C                                                                       EVD30540
  150 EVD3 = -(E0 + E1) / 12.56637                                      EVD30550
      RETURN                                                            EVD30560
  160 EVD3 = -PART1 / 6.283185                                          EVD30570
      RETURN                                                            EVD30580
      END                                                               EVD30590
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK EVD4
      FUNCTION EVD4(X,Y,D1,D2,DELTA)                                    EVD40010
C                                                                       EVD40020
C  THIS FUNCTION CALCULATES THE DOWNWASH AT ANY POINT X,Y               EVD40030
C  DUE TO A HINGE EVD ELEMENT WITH ONE RADIAN TURNING ANGLE             EVD40040
C  LOCATED AT THE ORIGIN C.S.                                           EVD40050
C                                                                       EVD40060
      DIMENSION S1(20),FACTOR(9)                                        EVD40070
      R(A,B) = SQRT(A**2 + B**2)                                        EVD40080
      CHANGE(A,B,C) = 0.50 * (C * (B-A) + (A+B))                        EVD40090
      S(A) = ABS(A) / A                                                 EVD40100
      SQ(T) = 0.50 * (-D1L/D1*(1.00-S(T)) + D2L/D2*(1.00+S(T)))         EVD40110
      G(A) = ALOG(ABS(A)) - SQ(A) * A                                   EVD40120
      DATA S1/-0.9061798,-0.9324695,-0.8833713,-0.8136032,-0.7243866,-0.EVD40130
     1      -0.6178762,-0.5051585,-0.3878584,-0.2678537,-0.1467914,-0.02EVD40140
     2      -0.9324695,-0.6133714,-0.2386042,0.1118633,0.4686075/       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
      DR = 0.50 * (D1 + D2)                                             EVD40200
   10 XR = X / DR                                                       EVD40210
      XMD = X - D2                                                      EVD40220
      XPD = X + D1                                                      EVD40230
   20 YMD = Y - DELTA                                                   EVD40240
      YPD = Y + DELTA                                                   EVD40250
   30 RMD = R(XMD,YMD)                                                  EVD40260
      RPD = R(XPD,YMD)                                                  EVD40270
      AX = ABS(X)                                                       EVD40280
      AXR = ABS(XR)                                                     EVD40290
      D1L = ALOG(D1)                                                    EVD40300
      D2L = ALOG(D2)                                                    EVD40310
      XD1 = X/D1                                                        EVD40320
      XD2 = X/D2                                                        EVD40330
   40 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(AXR .LT. 7.5) GO TO 70                                         EVD40380
      N = 0                                                             EVD40390
      RK = 0.00                                                         EVD40400
      D1X = D1/X                                                        EVD40410
      D2X = D2/X                                                        EVD40420
      DO 60 N1 = 1,100,2                                                EVD40430
      RKN = 0.000                                                       EVD40440
      DO 50 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
   50 CONTINUE                                                          EVD40510
      RK = RK + RKN                                                     EVD40520
      IF(ABS(RKN/RK) .LT. 1.0E-07) GO TO 190                            EVD40530
   60 CONTINUE                                                          EVD40540
   70 IF(AXR .LT. 1.0E-04) GO TO 200                                    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
   80 IF(RK1P .LT. 1.0E-06) GO TO 90                                    EVD40620
      RK1 = (AL*SQ(AL)*(XD1)) + (XD1+1.00) * D1L) * ALOG(ABS(XPD/X))    EVD40630
   90 RK2P = ABS(XD2-1.00)                                              EVD40640
      IF(RK2P .LT. 1.0E-06) GO TO 100                                   EVD40650
      RK2 = -(ALOG(ABS(XD2)) - (XD2-1.00) * D2L) * ALOG(ABS(XMD/X))     EVD40660
  100 IF(ABS(D1/AX-1.00) .LT. 1.0E-04) GO TO 140                        EVD40670
C  CALCULATE RK3 BY GAUSSIAN INTEGRATION.                               EVD40680
      AL = 1.00                                                         EVD40690
      BL = D1/AX                                                        EVD40700
      DO 120 N = 1,9                                                    EVD40710
  110 T = CHANGE(AL,BL,S1(N))                                           EVD40720
      RK3 = RK3 + FACTOR(N) * (ALOG(T) / (SX+T))                        EVD40730
  120 CONTINUE                                                          EVD40740
      RK3 = 0.50 * (BL-AL) * RK3                                        EVD40750
C  CALCULATE RK4 BY GAUSSIAN INTEGRATION.                               EVD40760
  140 IF(ABS(D2/AX-1.00) .LT. 1.0E-04) GO TO 180                        EVD40770
      AL = 1.00                                                         EVD40780
      BL = D2/AX                                                        EVD40790
      DO 170 N = 1,9                                                    EVD40800
  150 T = CHANGE(AL,BL,S1(N))                                           EVD40810
  160 RK4 = RK4 + FACTOR(N) * (ALOG(T) / (SX-T))                        EVD40820
  170 CONTINUE                                                          EVD40830
      RK4 = 0.50 * (BL-AL) * RK4                                        EVD40840
  180 RK = -2.467401 * SX - (D1L-D2L) + (RK1 + RK2) + (RK3 + RK4)       EVD40850
  190 P = (RQP/YMD - RQPP/YPD) * RK                                     EVD40860
      GO TO 210                                                         EVD40870
  200 P = 0.00                                                          EVD40880
C                                                                       EVD40890
C  CALCULATE E(X) BY GAUSSIAN INTEGRATION.                              EVD40900
  210 IF(X .GT. -D1) .AND. (X .LT. D2)) GO TO 290                       EVD40910
C  Y IS NOT WITHIN THE DIMENSIONS OF THE ELEMENT.                       EVD40920
C  LEFT SIDE INTEGRAL.                                                  EVD40930
      EL = 0.00                                                         EVD40940
      AL = -D1                                                          EVD40950
      BL = 0.00                                                         EVD40960
      DO 240 N = 1,9                                                    EVD40970
  220 SY = CHANGE(AL,BL,S1(N))                                          EVD40980
      XMS = X - SY                                                      EVD40990
      GS = G(SY)                                                        EVD41000
  230 PART1 = GS + (R(XMS,YMD)-RQP) / YMD - GS * (R(XMS,YPD)-RQPP) / YPDEVD41010
      EL = EL + FACTOR(N) * PART1 / XMS                                 EVD41020
  240 CONTINUE                                                          EVD41030
      EL = 0.50 * (BL-AL) * EL                                          EVD41040
C  RIGHT SIDE INTEGRAL                                                  EVD41050
      ER = 0.00                                                         EVD41060
      AL = 0.00                                                         EVD41070
      BL = D2                                                           EVD41080
      DO 270 N = 1,9                                                    EVD41090
  250 SY = CHANGE(AL,BL,S1(N))                                          EVD41100
      XMS = X - SY                                                      EVD41110
      GS = G(SY)                                                        EVD41120
  260 PART1 = GS + (R(XMS,YMD)-RQP) / YMD - GS * (R(XMS,YPD)-RQPP) / YPDEVD41130
      ER = ER + FACTOR(N) * PART1 / XMS                                 EVD41140
  270 CONTINUE                                                          EVD41150
      ER = 0.50 * (BL-AL) * ER                                          EVD41160
      P = EL + ER                                                       EVD41170
      GO TO 460                                                         EVD41180
C                                                                       EVD41190
C  X IS WITHIN THE DIMENSIONS OF THE ELEMENT                            EVD41200
C  CALCULATE EQ                                                         EVD41210
  290 EQ = 0.00                                                         EVD41220
      GPX = 0.00                                                        EVD41230
      GPPX = 0.00                                                       EVD41240
      IF(AXR .LT. 1.0E-04) GO TO 310                                    EVD41250
      SQX = SQ(X) - 1.00/X                                              EVD41260
      GX = G(X)                                                         EVD41270
      GPX = GX - (ABS(YMD)-RQP)                                         EVD41280
      GPPX = GX - (ABS(YPD)-RQPP)                                       EVD41290
      IF((1.00-XD2) .LT. 1.0E-06) .OR. ((XD1+1.00) .LT. 1.0E-06))       EVD41300
     1    GO TO 310                                                     EVD41310
  300 EQ = -(GPX/YMD - GPPX/YPD) + ALOG(ABS(XMD/XPD))                   EVD41320
C                                                                       EVD41330
C  CALCULATE EL                                                         EVD41340
C  LEFT SIDE INTEGRAL                                                   EVD41350
  310 EL = 0.00                                                         EVD41360
      AL = -D1                                                          EVD41370
      BL = 0.00                                                         EVD41380
      DO 330 N = 1,9                                                    EVD41390
  320 SY = CHANGE(AL,BL,S1(N))                                          EVD41400
      XMS = X - SY                                                      EVD41410
      IF(ABS(XMS/D1) .LT. 1.0E-04) GO TO 350                            EVD41420
      GS = G(SY)                                                        EVD41430
  330 PART2 = (GS + (R(XMS,YMD)-RQP) - GPX) / XMS                       EVD41440
  340 PART3 = (GS + (R(XMS,YPD)-RQPP) - GPPX) / XMS                     EVD41450
      PART4 = PART2/YMD - PART3/YPD                                     EVD41460
      GO TO 360                                                         EVD41470
  350 PART2 = RQP/YMD - RQPP/YPD                                        EVD41480
      PART4 = (S(YMD) - S(YPD)) - PART2) * SQX                          EVD41490
  360 EL = EL + FACTOR(N) * PART4                                       EVD41500
  370 CONTINUE                                                          EVD41510
      EL = 0.50 * (BL-AL) * EL                                          EVD41520
C                                                                       EVD41530
C  RIGHT SIDE INTEGRAL                                                  EVD41540
  380 ER = 0.00                                                         EVD41550
      AL = 0.00                                                         EVD41560
      BL = D2                                                           EVD41570
      DO 440 N = 1,9                                                    EVD41580
  390 SY = CHANGE(AL,BL,S1(N))                                          EVD41590
      XMS = X - SY                                                      EVD41600
      IF(ABS(XMS/D1) .LT. 1.0E-04) GO TO 420                            EVD41610
      GS = G(SY)                                                        EVD41620
  400 PART2 = (GS + (R(XMS,YMD)-RQP) - GPX) / XMS                       EVD41630
  410 PART3 = (GS + (R(XMS,YPD)-RQPP) - GPPX) / XMS                     EVD41640
      PART4 = PART2/YMD - PART3/YPD                                     EVD41650
      GO TO 430                                                         EVD41660
  420 PART2 = RQP/YMD - RQPP/YPD                                        EVD41670
      PART4 = (S(YMD) - S(YPD)) - PART2) * SQX                          EVD41680
  430 ER = ER + FACTOR(N) * PART4                                       EVD41690
  440 CONTINUE                                                          EVD41700
      ER = 0.50 * (BL-AL) * ER                                          EVD41710
  450 P = EL + ER + EQ                                                  EVD41720
C                                                                       EVD41730
C  CALCULATE THE DOWNWASH INFLUENCE COEFFICIENT                         EVD41740
  460 EVD4 = (PART5 + PART6 + P + E) / 19.739202                        EVD41750
      RETURN                                                            EVD41760
      END                                                               EVD41770
```

C10

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK SHUFL1                                                          SHF1C010
      SUBROUTINE SHUFL1(W,ISIZE)                                      SHF1C020
C                                                                     SHF1C030
C  THIS SUBROUTINE READS THE PORTION OF THE DOWNWASH MATRIX WHICH     SHF1C040
C  CONTAINS THE DOWNWASH DUE TO THE JET, AUGMENTS IT ACCRODING TO     SHF1C050
C  THE CURRENT CMU VALUES, AND WRITES IT BACK ON UNIT 1               SHF1C060
C  REWIND THE DOWNWASH MATRIX                                         SHF1C070
C                                                                     SHF1C080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IWINGE         SHF1C090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) SHF1C100
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(6CC),XI(6CC),DEL(65C),SHF1C110
     1                 D(4C),KK(6CC),ITYPE(6CC)                     SHF1C111A
      COMMON/JCASE/CMU(4C),CMUP(4C),CMUPP(4C)                         SHF1C120
      DIMENSION A(4CC),AIM1(4CC),W(ISIZE)                             SHF1C130
C                                                                     SHF1C140
      NWT1 = NWT + 1                                                  SHF1C150
      IF(IPRINT .LT. 2) WRITE(6, 1C )                                 SHF1C160
   1C FORMAT(1H1,42X,35HAUGMENTED PORTION OF SOLUTION MATRIX/)        SHF1C170
C                                                                     SHF1C180
C  PREPARE THE SOLUTION MATRIX FOR ROWS ON THE JET                    SHF1C190
C                                                                     SHF1C200
C       I IS THE COUNTER FOR IDENTIFYING ELEMENTS ON THE JET          SHF1C210
C    IREAD IS THE COUNTER FOR DOWNWASH ROWS ON THE JET STORED ON UNIT 1 SHF1C220
C  IWRITE IS THE COUNTER FOR AUGMENTED ROWS TO BE WRITTEN ON UNIT 1   SHF1C230
   2C IREAD = NWT                                                     SHF1C240
      IWRITE = NMAX                                                   SHF1C250
      C1 = 0.125CCC                                                   SHF1C260
      C3 = 0.375CCC                                                   SHF1C270
      DO 15C I = NWT1,NMAX                                            SHF1C280
   3C IM1 = I - 1                                                     SHF1C290
      IP1 = I + 1                                                     SHF1C300
      IREAD = IREAD + 1                                               SHF1C310
      IWRITE = IWRITE + 1                                             SHF1C320
      K = KK(I)                                                       SHF1C330
C                                                                     SHF1C340
C  READ THE ITH ROW OF THE DOWNWASH MATRIX  (IREADTH RECORD)          SHF1C350
   4C CALL READMS(1,W,ISIZE,IREAD)                                    SHF1C360
C                                                                     SHF1C370
C  SAVE THE EXISTING ROW OF SIMPLE DOWNWASH COEFFICIENTS              SHF1C380
      DO 5C J = 1,NMAX                                                SHF1C390
      A(J) = W(J)                                                     SHF1C400
   5C CONTINUE                                                        SHF1C410
C                                                                     SHF1C420
C  SUBTRACT THE PREVIOUS ROW FROM THE PRESENT ROW IF THE DOWNWASH POINT SHF1C430
C  IS NOT ON A LEADING JET ELEMENT                                    SHF1C440
      PROD1 = CMUP(K) * DEL(I) * CHORD(K)                             SHF1C450
      PROD2 = CMUP(K) * DEL(IM1) * CHORD(K)                           SHF1C460
   6C IF(I .EQ. IJ(K)) GO TO 9C                                       SHF1C470
      DO 7C J = 1,NMAX                                                SHF1C480
      W(J) = W(J) - AIM1(J)                                           SHF1C490
   7C CONTINUE                                                        SHF1C500
C                                                                     SHF1C510
C  MODIFY THE TWO OR THREE SPECIAL ELEMENTS FURTHER                   SHF1C520
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 10C                           SHF1C530
C  DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT                 SHF1C540
   8C W(IM1) = W(IM1) + C1 * PROD2                                    SHF1C550
      W(I) = W(I) + C3 * (PROD1 + PROD2)                              SHF1C560
      W(IP1) = W(IP1) + C1 * PROD1                                    SHF1C570
      GO TO 11C                                                       SHF1C580
C  DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT                 SHF1C590
   9C W(I) = W(I) + C3 * PROD1                                        SHF1C600
      W(IP1) = W(IP1) + C1 * PROD1                                    SHF1C610
      GO TO 11C                                                       SHF1C620
C  DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT                SHF1C630
  1CC W(IM1) = W(IM1) + C1 * PROD2                                    SHF1C640
      W(I) = W(I) + C3 * PROD2 + CMUP(K) * D(K)                       SHF1C650
C                                                                     SHF1C660
C  STORE THE AUGMENTED ITH ROW ON UNIT 1  (IWRITETH RECORD)           SHF1C670
  11C CALL WRITMS(1,W,ISIZE,IWRITE)                                   SHF1C680
C  PRINT THE AUGMENTED PORTION OF THE MATRIX                          SHF1C690
      IF(IPRINT .LT. 2) WRITE(6, 12C ) I,W                            SHF1C700
  12C FORMAT(1H0,55X,15HMATRIX ROW,I4,6(/1X,1PE13.5))                 SHF1C710
C                                                                     SHF1C720
C  SAVE THE ITH ROW FOR USE AS THE I-1 ROW ON THE NEXT PASS           SHF1C730
  13C DO 14C J = 1,NMAX                                               SHF1C740
      AIM1(J) = A(J)                                                  SHF1C750
  14C CONTINUE                                                        SHF1C760
  15C CONTINUE                                                        SHF1C770
      RETURN                                                          SHF1C780
C                                                                     SHF1C790
C  DIRECT ACCESS UNIT 1 NOW CONTAINS THE FOLLOWING -                  SHF1C800
C     WING-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NWT RECORDS)    SHF1C810
C        JET-DUE-TO-WING-AND-JET DOWNWASH COEFFICIENTS (NJT RECORDS)  SHF1C820
C        JET-DUE-TO-WING-AND-JET AUGMENTED DOWNWASH COEFFICIENTS (NJT RECS)SHF1C830
      END                                                             SHF1C840
```

```
*DECK SHUFL2                                                          SHF2C010
      SUBROUTINE SHUFL2(W,ISIZE,NEWMAX)                               SHF2C020
C                                                                     SHF2C030
C  THIS SUBROUTINE READS EACH MATRIX ROW CORRESPONDING TO A DOWNWASH  SHF2C040
C  CONTROL POINT ON THE JET, MODIFIES IT ACCRODING TO THE NEW VALUES  SHF2C050
C  OF CMU, AND RESTORES IT IN ITS ORIGINAL PLACE                      SHF2C060
C                                                                     SHF2C070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IWINGE         SHF2C080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) SHF2C090
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(6CC),XI(6CC),DEL(65C),SHF2C100
     1                 D(4C),KK(6CC),ITYPE(6CC)                     SHF2C110
      COMMON/JCASE/CMU(4C),CMUP(4C),CMUPP(4C)                         SHF2C120
      DIMENSION W(ISIZE)                                              SHF2C130
C                                                                     SHF2C140
      IF(IPRINT .LT. 2) WRITE(6, 1C )                                 SHF2C150
   1C FORMAT(1H1,42X,36HAUGMENTED PORTION OF SOLUTION MATRIX)         SHF2C160
C                                                                     SHF2C170
C  CYCLE THE AUGMENTED MATRIX ROWS                                    SHF2C180
      IREAD = NMAX                                                    SHF2C190
      NWT1 = NWT + 1                                                  SHF2C200
      DO 15C I = NWT1,NEWMAX                                          SHF2C210
      IREAD = IREAD + 1                                               SHF2C220
   2C IM1 = I - 1                                                     SHF2C230
      IP1 = I + 1                                                     SHF2C240
      K = KK(I)                                                       SHF2C250
   3C CMUDIF = (CMUP(K) - CMUPP(K)) * CHORD(K)                        SHF2C260
C                                                                     SHF2C270
C  READ THE ITH AUGMENTED MATRIX ROW                                  SHF2C280
   4C CALL READMS(1,W,ISIZE,IREAD)                                    SHF2C290
C                                                                     SHF2C300
C  MODIFY THE TWO OR THREE SPECIAL ELEMENTS ACCORDING TO THE NEW CMU  SHF2C310
      IF(I .EQ. IJ(K)) GO TO 6C                                       SHF2C320
      IF(I .EQ. (IJ(K)+NJ(K)-1)) GO TO 7C                             SHF2C330
C  DOWNWASH CONTROL POINT IS ON A REGULAR JET ELEMENT                 SHF2C340
   5C W(IM1) = W(IM1) + 0.125C * DEL(IM1) * CMUDIF                    SHF2C350
      W(I) = W(I) + 0.175C * (DEL(IM1)+DEL(I)) * CMUDIF               SHF2C360
      W(IP1) = W(IP1) + 0.125C * DEL(I) * CMUDIF                      SHF2C370
      GO TO 9C                                                        SHF2C380
C  DOWNWASH CONTROL POINT IS ON A LEADING JET ELEMENT                 SHF2C390
   6C W(I) = W(I) + 0.375C * DEL(I) * CMUDIF                          SHF2C400
      W(IP1) = W(IP1) + 0.125C * DEL(I) * CMUDIF                      SHF2C410
      GO TO 8C                                                        SHF2C420
C  DOWNWASH CONTROL POINT IS ON A TRAILING JET ELEMENT                SHF2C430
   7C W(IM1) = W(IM1) + 0.125C * DEL(IM1) * CMUDIF                    SHF2C440
      W(I) = W(I) + (0.375C*DEL(IM1) + D(K)/CHORD(K)) * CMUDIF        SHF2C450
C                                                                     SHF2C460
C  WRITE THE REVISED ITH ROW ON UNIT 1                                SHF2C470
   8C CALL WRITMS(1,W,ISIZE,IREAD)                                    SHF2C480
      IF(IPRINT .LT. 3) WRITE(6, 9C ) I,W                             SHF2C490
   9C FORMAT(1H0,55X,15HMATRIX ROW,I4,6(/1X,1PE13.5))                 SHF2C500
  1CC CONTINUE                                                        SHF2C510
      RETURN                                                          SHF2C520
      END                                                             SHF2C530
```

```
*DECK COLUM1                                                          COL1C010
      SUBROUTINE COLUM1(LCASE)                                        COL1C020
C                                                                     COL1C030
C  THIS SUBROUTINE SETS UP THE RIGHT SIDE COLUMN MATRIX WITHOUT       COL1C040
C  CONSIDERATION OF ANY HINGE DOWNWASH INFLUENCE                      COL1C050
C                                                                     COL1C060
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) COL1C070
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(6CC),XI(6CC),DEL(65C),COL1C080
     1                 D(4C),KK(6CC),ITYPE(6CC)                     COL1C090
      COMMON/ECASE3/EPS(4CC,1C),BETA(6CC,1C),THETA(4C,1C),THS(4C,1C)  COL1C100
      COMMON/SOLV1/B(4CC,1C)                                          COL1C110
C                                                                     COL1C120
C  DEFINE THE ELEMENTS ON THE WING                                    COL1C130
      DO 1C I = 1,NWT                                                 COL1C140
      B(I,LCASE) = EPS(I,LCASE) / 57.295779                           COL1C150
   1C CONTINUE                                                        COL1C160
C                                                                     COL1C170
C  DEFINE THE ELEMENTS ON THE JET                                     COL1C180
      I = NWT                                                         COL1C190
      DO 4C K = 1,NROWS                                               COL1C200
      NJK = NJ(K)                                                     COL1C210
      IF(NJK .EQ. 0) GO TO 4C                                         COL1C220
C  FIRST JET ELEMENT                                                  COL1C230
      I = I + 1                                                       COL1C240
      KKT = KK(I)                                                     COL1C250
   2C B(I,LCASE) = THETA(KKT,LCASE) / 57.295779                       COL1C260
C  REMAINING JET ELEMENTS                                             COL1C270
      DO 3C L = 2,NJK                                                 COL1C280
      I = I + 1                                                       COL1C290
      B(I,LCASE) = 0.0C                                               COL1C300
   3C CONTINUE                                                        COL1C310
   4C CONTINUE                                                        COL1C320
      RETURN                                                          COL1C330
      END                                                             COL1C340
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK COLUM2
      SUBROUTINE COLUM2(H,ISIZE,NEWMAX,LCASE)                          COL2C010
C                                                                      COL2C020
C  THIS SUBROUTINE ADDS THE APPROPRIATE HINGE DOWNWASH INFLUENCE       COL2C030
C  TO THE RIGHT SIDE COLUMN MATRIX                                     COL2C040
C                                                                      COL2C050
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) COL2C060
      COMMON/GEOM1/YI(40),CHORD(40),DELTA(40),XA(600),XI(600),DEL(600),COL2C070
     1           DI(40),XK(600),ITYPE(600)                           COL2C080
      COMMON/ECASE3/PRESS(600,10),BETA(600,10),THETA(40,10),THS(40,10) COL2C090
      COMMON/ICASE/CMU(40),CMUP(40)                                    COL2C100
      COMMON/SOLVI/B(600,10)                                           COL2C110
      DIMENSION H(ISIZE)                                               COL2C120
C                                                                      COL2C130
C  DEFINE THE ELEMENTS ON THE WING                                     COL2C140
      DO 10 I = 1,NEWMAX                                               COL2C150
      H(I,LCASE) = B(I,LCASE) - H(I)                                   COL2C160
   10 CONTINUE                                                         COL2C170
C                                                                      COL2C180
C  DEFINE THE ELEMENTS ON THE JET                                      COL2C190
      IF(NEWMAX .EQ. NWT) RETURN                                       COL2C200
      I = NWT                                                          COL2C210
      DO 90 K = 1,NROWS                                                COL2C220
   20 NJK = NJ(K)                                                      COL2C230
      IF(NJK .EQ. 0) GO TO 90                                          COL2C240
C  COMPUTE THE CMU INFLUENCE FACTORS H1 AND H2                         COL2C250
      I = I + 1                                                        COL2C260
      H1 = 2.0E                                                        COL2C270
      H2 = -4.0E                                                       COL2C280
      RTA = BETA(I,LCASE)                                              COL2C290
      IF(ABS(RTA) .LT. 0.0001) GO TO 50                                COL2C300
      RTA = RTA / 57.295779                                            COL2C310
   30 D2 = DEL(I) * CHORD(K)                                           COL2C320
      DL = ALOG(D2)                                                    COL2C330
      PROD = -CMUP(K) * D2 * RTA / 3.1415927                           COL2C340
   40 H1 = PROD * (1.6331472 - 0.750 * DL)                             COL2C350
      H2 = PROD * (0.3248528 - 0.250 * DL)                             COL2C360
C  FIRST POINT ON THE JET                                              COL2C370
   50 B(I,LCASE) = B(I,LCASE) + H1                                     COL2C380
C  SECOND POINT ON THE JET                                             COL2C390
      I = I + 1                                                        COL2C400
   60 B(I,LCASE) = B(I,LCASE) + H(I-1) + H2                            COL2C410
C  REMAINING POINTS ON THE JET                                         COL2C420
      IF(NJK .LT. 3) GO TO 90                                          COL2C430
      DO 80 L = 3,NJK                                                  COL2C440
      I = I + 1                                                        COL2C450
   70 B(I,LCASE) = B(I,LCASE) + H(I-1)                                 COL2C460
   80 CONTINUE                                                         COL2C470
   90 CONTINUE                                                         COL2C480
      RETURN                                                           COL2C490
      END                                                              COL2C500


*DECK HINGE
      SUBROUTINE HINGE(H,ISIZE,NEWMAX,LCASE)                           HINGC010
C                                                                      HINGC020
C  THIS SUBROUTINE CALCULATES THE DOWNWASH INFLUENCE COEFFICIENTS      HINGC030
C  AT EACH DOWNWASH CONTROL POINT DUE TO ALL DEFLECTED HINGE ELEMENTS. HINGC040
C  FOR EACH CONTROL POINT THE INFLUENCE COEFFICIENTS ARE MULTIPLIED BY  HINGC050
C  THEIR RESPECTIVE DEFLECTION ANGLE AND SUMMED UP TO OBTAIN THE       HINGC060
C  COMPLETE HINGE-INDUCED DOWNWASH.                                    HINGC070
C                                                                      HINGC080
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           HINGC090
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) HINGC100
      COMMON/GEOM1/YI(40),CHORD(40),DELTA(40),XA(600),XI(600),DEL(600),HINGC110
     1           DI(40),XK(600),ITYPE(600)                           HINGC120
      COMMON/ECASE3/PRESS(600,10),BETA(600,10),THETA(40,10),THS(40,10) HINGC130
      DIMENSION H(ISIZE)                                               HINGC140
C                                                                      HINGC150
      IF(LCASE .GT. 2) GO TO 22                                        HINGC160
      IF(IPRINT .LT. 0) WRITE(6, 10 )                                  HINGC170
   10 FORMAT(1H1)                                                      HINGC180
      ILINES = 1                                                       HINGC190
C                                                                      HINGC200
C  CYCLE THE DOWNWASH CONTROL POINTS ON THE WING AND JET               HINGC210
   20 DO 110 I = 1,NEWMAX                                              HINGC220
      KI = KK(I)                                                       HINGC230
C                                                                      HINGC240
C  CYCLE THE VORTEX POINTS ON THE WING AND JET                         HINGC250
      DO 100 J = 1,NEWMAX                                              HINGC260
C  CHECK WHETHER THERE IS A DEFLECTED HINGE AT ELEMENT J               HINGC270
   30 A = BETA(J,LCASE)                                                HINGC280
      IF(ABS(A) .LT. 0.0001) GO TO 100                                 HINGC290
      A = A / 57.295779                                                HINGC300
C  COMPUTE THE GEOMETRIC PARAMETERS                                    HINGC310
      KJ = KK(J)                                                       HINGC320
   40 X = XI(I) + DEL(I)*CHORD(KI)/2.00 - XI(J)                        HINGC330
      YY = YI(KI) - YI(KJ)                                             HINGC340
   50 D2 = DEL(J) * CHORD(KJ)                                          HINGC350
      D1 = DEL(J-1) * CHORD(KJ)                                        HINGC360
      IN1 = IW(KJ) + NW(KJ) - 1                                        HINGC370
   60 IF(ITYPE(J) .EQ. 43) D1 = DEL(IN1) * CHORD(KJ)                   HINGC380
C  COMPUTE AND SUM UP THE INFLUENCE OF ELEMENT J                       HINGC390
   70 HI(I) = HI(I) + EVD4(X,YY,D1,D2,DELTA(KJ)) * A                   HINGC400
C  SUPERIMPOSE DOWNWASH FOR SYMETRIC OR ANTI-SYMETRIC GEOMETRY         HINGC410
      IF(ISYMM .GT. 0) GO TO 120                                       HINGC420
      YY = YI(KI) + YI(KJ)                                             HINGC430
   80 HOMMY = EVD4(X,YY,D1,D2,DELTA(KJ))                               HINGC440
      IF(ISYMM .LT. 0) HOMMY = - HOMMY                                 HINGC450
   90 HI(I) = HI(I) + HOMMY * A                                        HINGC460
  100 CONTINUE                                                         HINGC470
  110 CONTINUE                                                         HINGC480
C                                                                      HINGC490
C  PRINT OUT THE DOWNWASH IF REQUIRED                                  HINGC500
      IF(IPRINT .GE. 2) RETURN                                         HINGC510
      NEXT = NEWMAX/10 + 3                                             HINGC520
      IF((ILINES+NEXT) .LT. 56) .OR. (ILINES .EQ. 1)) GO TO 120        HINGC530
      WRITE(6, 10 )                                                    HINGC540
      ILINES = 1                                                       HINGC550
  120 IF(IPRINT .LT. 0) WRITE(6, 130 ) LCASE,H                         HINGC560
  130 FORMAT(1HC,35X,44HHINGE INFLUENCE COEFFICIENTS FOR FUNDAMENTAL,  HINGC570
     1   5H CASE, I3,6E(/1X,10E13.5))                                  HINGC580
      ILINES = ILINES + NEXT                                           HINGC590
      RETURN                                                           HINGC600
      END                                                              HINGC610


*DECK PREP
      SUBROUTINE PREP(TRANS,ISIZE,NEWMAX)                              PREPC010
C                                                                      PREPC020
C  THIS SUBROUTINE PREPARES THE FINAL MATRIX FOR SOLUTION BY           PREPC030
C  CONCATINATING IT WITH THE RIGHT SIDE MATRIX AND STORING IT          PREPC040
C  ON SCRATCH UNIT 2 FOR INPUT TO MATRIX.                              PREPC050
C                                                                      PREPC060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           PREPC070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) PREPC080
      COMMON/SOLVI/B(600,10)                                           PREPC090
      DIMENSION TRANS(ISIZE)                                           PREPC100
C                                                                      PREPC110
      IREAD = 0                                                        PREPC120
      REWIND 2                                                         PREPC130
C                                                                      PREPC140
C  READ THE MATRIX COEFFICIENTS                                        PREPC150
      DO 40 I = 1,NEWMAX                                               PREPC160
      IF(I .EQ. NWT+1) IREAD = NEWMAX                                  PREPC170
      IREAD = IREAD + 1                                                PREPC180
   40 CALL READMS(I,TRANS,ISIZE,IREAD)                                 PREPC190
C                                                                      PREPC200
C  PULL OUT THE RIGHT SIDE MATRIX COEFFICIENTS AND CONCATINATE THEM    PREPC210
      DO 20 N = 1,NCASES                                               PREPC220
      TRANS(NEWMAX+N) = B(I,N)                                         PREPC230
   20 CONTINUE                                                         PREPC240
C  THE MATRIX ROW HAS NOW BEEN ASSEMBLED AND FILLS THE TRANS ARRAY.    PREPC250
C  WRITE THE CONCATINATED ROW ON ON UNIT 2 FOR INPUT TO MATRIX.        PREPC260
      WRITE(2) TRANS                                                   PREPC270
   60 CONTINUE                                                         PREPC280
C  THE SYSTEM OF LINEAR EQUATIONS IS NOW READY FOR SOLUTION            PREPC290
      RETURN                                                           PREPC300
      END                                                              PREPC310


*DECK ST625
      OVERLAY(OVLY,2,2)                                                ST2S0010
      PROGRAM ST625                                                    ST2S0020
C                                                                      ST2S0030
C  THIS PROGRAM CONTROLS SOLUTION OF THE MATRIX SYSTEM                 ST2S0040
C                                                                      ST2S0050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           ST2S0060
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NCALFA,LOGIC,IR                     ST2S0070
      COMMON/SOLVI/GAMMA(600,12)                                       ST2S0080
      COMMON/SOLV3/WKAREA(10000)                                       ST2S0090
      DIMENSION W(600),SUMMER(600)                                     ST2S0100
C                                                                      ST2S0110
      IF(IPRINT .LT. 0) WRITE(6, 10 )                                  ST2S0120
   10 FORMAT(1H1,53X,14HGAMMA SOLUTION)                                ST2S0130
C                                                                      ST2S0140
C  SOLVE THE MATRIX FOR GAMMA USING MATPIX                             ST2S0150
      NSIZE = 10000                                                    ST2S0160
      NIN = 2                                                          ST2S0170
      NSCR1 = 3                                                        ST2S0180
      NSCR2 = 4                                                        ST2S0190
   20 CALL MATPIX(NEWMAX,NCASES,NSIZE,NIN,NSCR1,NSCR2,IR)              ST2S0200
      IF(IR .EQ. 2) GO TO 60                                           ST2S0210
C  MATRIX HAS STORED THE SOLUTION IN THE FIRST STORAGE LOCATIONS OF THE ST2S0220
C  WKAREA ARRAY.  TRANSFER THIS DATA INTO THE GAMMA ARRAY.             ST2S0230
      ISUM = 0                                                         ST2S0240
      DO 70 N = 1,NCASES                                               ST2S0250
      DO 40 J = 1,NEWMAX                                               ST2S0260
      GAMMA(J,N) = WKAREA(ISUM+J)                                      ST2S0270
   40 CONTINUE                                                         ST2S0280
      IF(IPRINT .LT. 0) WRITE(6, 50 ) N,(GAMMA(J,N),J=1,NEWMAX)        ST2S0290
   50 FORMAT(1HC,50X,14HFUNDAMENTAL CASE,I4,60(/1X,10E13.5))           ST2S0300
      ISUM = ISUM + NEWMAX                                             ST2S0310
   70 CONTINUE                                                         ST2S0320
C                                                                      ST2S0330
C  IF REQUIRED, CHECK THE SOLUTION BY BACK SUBSTITUTION                ST2S0340
   80 IF(IPRINT .LT. 0) CALL BAKSUB(W,SUMMER,NEWMAX)                   ST2S0350
      IR = 1                                                           ST2S0360
      GO TO 110                                                        ST2S0370
C                                                                      ST2S0380
C  PRINT THE MATRIX ERROR MESSAGE                                      ST2S0390
   90 WRITE(6, 100 )                                                   ST2S0400
  100 FORMAT(1H1,40X,43HMATRIX DOES NOT HAVE ENOUGH CORE TO WORK/      ST2S0410
     1   45X,29HTHIS CASE HAS BEEN TERMINATED)                         ST2S0420
      IR = 2                                                           ST2S0430
  110 CONTINUE                                                         ST2S0440
      END                                                              ST2S0450
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK MATRIX
      SUBROUTINE MATRIX(ND, MD, KD, NI, MM, NO, IR)                     MATR0020
C                                                                       MATR0030
C              D I R E C T   M A T R I X   S O L U T I O N               MATR0040
C                                                                       MATR0050
      COMMON/SOLV2/A(100000)                                            MATR0060
      LOGICAL  LAST                                                     MATR0070
C                                                                       MATR0080
      N = ND                                                            MATR0090
      M = MD                                                            MATR0100
      KORE = KD                                                         MATR0110
      NPM = N + M                                                       MATR0120
      IF (MAXF13 - NPM, M + N1 .LT. KORE) GO TO 20                      MATR0130
      IR = 2                                                            MATR0140
      RETURN                                                            MATR0150
   20 MT = MM                                                           MATR0160
      REWIND MT                                                         MATR0170
      NIN = NI                                                          MATR0180
      REWIND NIN                                                        MATR0190
      NOUT = NO                                                         MATR0200
      REWIND NOUT                                                       MATR0210
      MP1 = M + 1                                                       MATR0220
      MN = N                                                            MATR0230
      NEL = NPM                                                         MATR0240
C                                                                       MATR0250
C - - CALCULATE THE MAXIMUM NO. OF ROWS, *K*                            MATR0260
C                                                                       MATR0270
   30 K = (KORE - NEL) / NEL                                            MATR0280
C                                                                       MATR0290
C - - TEST TO SEE IF THE REST OF THE MATRIX WILL FIT IN CORE            MATR0300
C                                                                       MATR0310
      LAST = K .GE. NN                                                  MATR0320
      IF (LAST) K = NN                                                  MATR0330
C                                                                       MATR0340
C - - READ *K* ROWS OF THE AUGMENTED *A* MATRIX                         MATR0350
C                                                                       MATR0360
   40 NT = 0                                                            MATR0370
      DO 50 IR = 1, K                                                   MATR0380
      NS = NT + 1                                                       MATR0390
      NT = NT + NEL                                                     MATR0400
   50 CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                            MATR0410
C                                                                       MATR0420
C - - CHECK TO SEE IF WE WERE UNLUCKY ENOUGH TO END UP WITH ONLY ONE ROWMATR0430
C                                                                       MATR0440
      IF (K .EQ. 1) GO TO 110                                           MATR0450
C                                                                       MATR0460
C - - *K* IS GREATER THAN *1* SO WE CAN START THE TRIANGULARIZATION      MATR0470
C                                                                       MATR0480
      NEL01 = NEL + 1                                                   MATR0490
      NS = - NEL                                                        MATR0500
      NEL02 = NEL01 + 1                                                 MATR0510
C                                                                       MATR0520
C - - FORM THE *TRAPEZOIDAL* ARRAY  (A)                                 MATR0530
C                                                                       MATR0540
      DO 60 IR = 2, K                                                   MATR0550
      NR = NEL02 - IR                                                   MATR0560
      NS = NS + NEL01                                                   MATR0570
      NT = NS                                                           MATR0580
      DO 60 IC = IR, K                                                  MATR0590
      NT = NT + NEL                                                     MATR0600
      MN = NT                                                           MATR0610
      NR = NS                                                           MATR0620
      A(NT) = (-A(NT)) / A(NS)                                          MATR0630
      DO 60 NE = 2, NR                                                  MATR0640
      MN = MN + 1                                                       MATR0650
      NR = NR + 1                                                       MATR0660
   60 A(MN) = A(MN) + A(NT) * A(NR)                                     MATR0670
      IF (LAST) GO TO 110                                               MATR0680
C                                                                       MATR0690
C - - WRITE THE *TRAPEZOIDAL* MATRIX ON TAPE                            MATR0700
C                                                                       MATR0710
      NT = 0                                                            MATR0720
      NR = NEL                                                          MATR0730
      NS = - NEL                                                        MATR0740
      DO 70 IC = 1, K                                                   MATR0750
      NS = NS + NEL01                                                   MATR0760
      NT = NT + NEL                                                     MATR0770
      CALL  SAVE(MT, 2, NR, NR, A(NS), 1, AA2)                          MATR0780
   70 NR = NR - 1                                                       MATR0790
      NR = NR - M                                                       MATR0800
      NS = KORE - NEL + 1                                               MATR0810
C                                                                       MATR0820
C - - READ ANOTHER ROW                                                  MATR0830
C                                                                       MATR0840
      DO 100 IC = 1, NR                                                 MATR0850
      CALL  GETT(NIN, 1, NEL, A(NS), 1, AA2)                            MATR0860
C                                                                       MATR0870
C - - MODIFY THIS ROW BY THE *TRAPEZOIDAL* ARRAY                        MATR0880
C                                                                       MATR0890
      NT = 1                                                            MATR0900
      MN = NS                                                           MATR0910
      DO 90 IR = 1, K                                                   MATR0920
      NR = NT                                                           MATR0930
      NT = NT + NEL                                                     MATR0940
      NE = MN + 1                                                       MATR0950
      A(MN) = (-A(MN)) / A(NT)                                          MATR0960
      DO 80 MN = NE, KORE                                               MATR0970
      NR = NR + 1                                                       MATR0980
   80 A(MN) = A(MN) + A(MN) * A(NR)                                     MATR0990
      MN = NE                                                           MATR1000
   90 NT = NT + NEL01                                                   MATR1010
C                                                                       MATR1020
C - - WRITE THE MODIFIED ROW ON TAPE                                    MATR1030
C                                                                       MATR1040
      NN1 = KORE - MN + 1                                               MATR1050
  100 CALL  SAVE(NOUT, 1, NN1, NN1, A(MN), 1, AA2)                      MATR1060
      REWIND NOUT                                                       MATR1070
      REWIND NIN                                                        MATR1080
C                                                                       MATR1090
C - - SWITCH THE TAPES                                                  MATR1100
C                                                                       MATR1110
      NT = NIN                                                          MATR1120
      NIN = NOUT                                                        MATR1130
      NOUT = NT                                                         MATR1140
C                                                                       MATR1150
C - - RE-CALCULATE ROW LENGTH AND LOOP BACK                             MATR1160
C                                                                       MATR1170
      NEL = NEL - K                                                     MATR1180
      NN = NEL - M                                                      MATR1190
      GO TO 30                                                          MATR1200
C                                                                       MATR1210
C - - REWIND ALL TAPES                                                  MATR1220
C                                                                       MATR1230
  110 REWIND MT                                                         MATR1240
      REWIND NIN                                                        MATR1250
      REWIND NOUT                                                       MATR1260
C                                                                       MATR1270
C - - CONDENSE THE MATRIX                                               MATR1280
C                                                                       MATR1290
      NN = NEL                                                          MATR1300
      NL = NEL + 1                                                      MATR1310
      IF (K .EQ. 1) GO TO 130                                           MATR1320
      NS = 1                                                            MATR1330
      NT = NEL                                                          MATR1340
      DO 120 IR = 2, K                                                  MATR1350
      NS = NS + NEL01                                                   MATR1360
      NT = NT + NEL                                                     MATR1370
      DO 120 IC = NS, NT                                                MATR1380
      A(NL) = A(IC)                                                     MATR1390
  120 NL = NL + 1                                                       MATR1400
  130 NI = KORE - K + M + 1                                             MATR1410
C                                                                       MATR1420
C - - THERE, NOW WE CAN START THE BACK-SOLUTION                         MATR1430
C * NOTE..THE FIRST AVAILABLE LOCATION FOR THE SOLUTIONS IS A(NI)        MATR1440
C                                                                       MATR1450
      NREM = N                                                          MATR1460
      NEL = NPM                                                         MATR1470
      LAST = K .EQ. N                                                   MATR1480
      NPASS = 0                                                         MATR1490
C                                                                       MATR1500
C - - SOLVE FOR THE ANSWERS CORRESPONDING TO *K* ROWS                   MATR1510
C                                                                       MATR1520
  140 KM1 = K - 1                                                       MATR1530
      KP1 = K + 1                                                       MATR1540
      NS = NL - MP1                                                     MATR1550
      NPASS = NPASS + 1                                                 MATR1560
      DO 170 MN = 1, M                                                  MATR1570
      NE = NS + MN                                                      MATR1580
      A(NE) = A(NE) / A(NS)                                             MATR1590
      NT = NS                                                           MATR1600
      IF (KM1 .EQ. 0) GO  TO 170                                        MATR1610
      DO 160 IR = 1, KM1                                                MATR1620
      NE = NE - IR - M                                                  MATR1630
      NT = NT - MP1 - IR                                                MATR1640
      SUM = 0.0                                                         MATR1650
      NP = NE                                                           MATR1660
      N2 = MP1 + IR                                                     MATR1670
      DO 150 IO = 1, IR                                                 MATR1680
      NN = NT + IO                                                      MATR1690
      NP = NP + N2 - IO                                                 MATR1700
  150 SUM = SUM + A(NN) * A(NP)                                         MATR1710
  160 A(NE) = (A(NE) - SUM) / A(NT)                                     MATR1720
  170 CONTINUE                                                          MATR1730
C                                                                       MATR1740
C - - MOVE THE SOLUTIONS TO CONTIGUOUS LOCATIONS STARTING AT A(NI)      MATR1750
C                                                                       MATR1760
      NI = KORE + 1                                                     MATR1770
      DO 180 NS = 1, K                                                  MATR1780
      DO 180 MN = 1, M                                                  MATR1790
      NL = NL - 1                                                       MATR1800
      NI = NI - 1                                                       MATR1810
  180 A(NI) = A(NL)                                                     MATR1820
  190 NL = NL - MN                                                      MATR1830
C                                                                       MATR1840
C - - WRITE THE SOLUTIONS ON TAPE                                       MATR1850
C                                                                       MATR1860
      WRITE (NIN) K                                                     MATR1870
      NS = NI - 1                                                       MATR1880
      DO 200 MN = 1, M                                                  MATR1890
      NT = NS + MN                                                      MATR1900
  200 WRITE ( NIN ) (A(IO), IO = NT, KORE, M)                           MATR1910
C                                                                       MATR1920
C - - TEST IF THIS IS THE LAST PASS                                     MATR1930
C                                                                       MATR1940
      IF (LAST) GO TO 280                                               MATR1950
C                                                                       MATR1960
C - - WE MUST NOW MODIFY THE TRIANGULAR MATRIX TO REFLECT THE EFFECT OF  MATR1970
C THE SOLUTIONS OBTAINED SO FAR  (EQ 21)                                MATR1980
C * NOTE..LOCATIONS 1(1) TO A(NI-1) ARE NOW FREE TO USE                 MATR1990
C                                                                       MATR2000
C - - CALCULATE THE NEXT VALUES OF *NEL* AND *NREM*                     MATR2010
C                                                                       MATR2020
      NELOLD = NEL                                                      MATR2030
      KOLD = K                                                          MATR2040
      NEL = NEL - K                                                     MATR2050
      NREM = NREM - K                                                   MATR2060
C                                                                       MATR2070
      CALCULATE NEW K.  B AND C (REAL) WILL ALWAYS BE INTEGERS.         MATR2080
      K WILL BE CALCULATED REAL AND TRUNCATED - - GOOD.                 MATR2090
C                                                                       MATR2100
      B = 1 + 2*M                                                       MATR2110
      C = 2*(KOLD*(M+1) - KORE)                                         MATR2120
      W = F-B + SQRT(B**2 - 4*C)1/2.0                                   MATR2130
      NROW = NREM - K + 1                                               MATR2140
      IF (K .LT. NREM) GO TO 210                                        MATR2150
      LAST = .TRUE.                                                     MATR2160
      NROW = 1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
      K = NREM                                                          MATR2170
  210 NS = 1                                                            MATR2180
      NT = NELOLD + 1                                                   MATR2190
C                                                                       MATR2200
C - - READ IN THE ROWS TO BE MODIFIED                                   MATR2210
C                                                                       MATR2220
      DO 270 IR = 1, NREM                                               MATR2230
      NT = NT - 1                                                       MATR2240
      IF (IR .LE. NROW) GO TO 220                                       MATR2250
      NS = NS + NM                                                      MATR2260
      NT = NT + NN                                                      MATR2270
  220 CALL  GETT(NT, 2, NN, A(NS), 1, AA2)                              MATR2280
      NP = N1 - 1                                                       MATR2290
      NE = NT - M - KM1                                                 MATR2300
      NN = NN - KOLD                                                    MATR2310
      DO 240 MN = 1, M                                                  MATR2320
      N2 = NE                                                           MATR2330
      NA = NP + MN                                                      MATR2340
      NB = NA                                                           MATR2350
      SUM = 0.0                                                         MATR2360
      DO 230 IO = 1, KOLD                                               MATR2370
      SUM = SUM + A(N2) * A(NA)                                         MATR2380
      N2 = N2 + 1                                                       MATR2390
  230 NA = NA + M                                                       MATR2400
      N2 = N2 + MN - 1                                                  MATR2410
  240 A(N2) = A(N2) - SUM                                               MATR2420
C                                                                       MATR2430
C - - WRITE THE MODIFIED ROW ON TAPE OR CONDENSE THE ROW                MATR2440
C                                                                       MATR2450
      NL = NT - N + 1                                                   MATR2460
      IF (IR .GE. NROW) GO TO 250                                       MATR2470
      NE = NL - KM1                                                     MATR2480
      NN1 = NE - NS + 1                                                 MATR2490
      NN2 = NT - NL + 1                                                 MATR2500
      CALL  SAVE(NOUT, 4, NN, NN1, A(NS), NN2, A(NL))                   MATR2510
      GO TO 270                                                         MATR2520
  250 NE = NL - KOLD                                                    MATR2530
      DO 260 MN = NL, NT                                                MATR2540
      A(NE) = A(MN)                                                     MATR2550
  260 NE = NE + 1                                                       MATR2560
  270 CONTINUE                                                          MATR2570
      REWIND MT                                                         MATR2580
      REWIND NOUT                                                       MATR2590
C                                                                       MATR2600
C - - SWITCH THE TAPES                                                  MATR2610
C                                                                       MATR2620
      NT = MT                                                           MATR2630
      MT = NOUT                                                         MATR2640
      NOUT = NT                                                         MATR2650
C                                                                       MATR2660
C - - LOOP BACK THRU THE SOLUTION                                       MATR2670
C                                                                       MATR2680
      NL = NE                                                           MATR2690
      GO TO 140                                                         MATR2700
C                                                                       MATR2710
C - - START TO WRAP IT UP                                               MATR2720
C                                                                       MATR2730
  280 REWIND NIN                                                        MATR2740
      N2 = N                                                            MATR2750
C                                                                       MATR2760
C * * NOTE.. AT THIS POINT ALL LOCATIONS A(1) THRU A(KOPE) ARE FREE      MATR2770
C                                                                       MATR2780
      DO 300 IR = 1, NPASS                                              MATR2790
      READ (NIN) K                                                      MATR2800
      N1 = N2 - K + 1                                                   MATR2810
      NS = N1                                                           MATR2820
      NT = N2                                                           MATR2830
C                                                                       MATR2840
C - - READ IN THE SOLUTIONS                                             MATR2850
C                                                                       MATR2860
      NM = NT - NS + 1                                                  MATR2870
      DO 290 IO = 1, M                                                  MATR2880
      CALL  GETT(NIN, 1,NM, A(NS), 1, AA2)                              MATR2890
  290 NS = NS + N                                                       MATR2900
  300 N2 = N1 - 1                                                       MATR2910
      IR = 1                                                            MATR2920
      RETURN                                                            MATR2930
      END                                                               MATR2940
```

```
*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 ( 10 , 20 , 30 , 40 ), IT                                   SAVE0060
C                                                                       SAVE0070
C  WRITE A1                                                             SAVE0080
   10 WRITE(IU) A1                                                      SAVE0090
      RETURN                                                            SAVE0100
C                                                                       SAVE0110
C  WRITE N AND A1                                                       SAVE0120
   20 WRITE(IU)  N, A1                                                  SAVE0130
      RETURN                                                            SAVE0140
C                                                                       SAVE0150
C  WRITE A1 AND A2                                                      SAVE0160
   30 WRITE(IU) A1, A2                                                  SAVE0170
      RETURN                                                            SAVE0180
C                                                                       SAVE0190
C  WRITE N, A1, AND A2                                                  SAVE0200
   40 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 ( 10 , 20 , 30 , 40 ), IT                                   GETT0060
C                                                                       GETT0070
C  READ A1                                                              GETT0080
   10 READ(IU) A1                                                       GETT0090
      RETURN                                                            GETT0100
C                                                                       GETT0110
C  READ N1 AND A1                                                       GETT0120
   20 READ(IU) N1, A1                                                   GETT0130
      RETURN                                                            GETT0140
C                                                                       GETT0150
C  READ A1 AND A2                                                       GETT0160
   30 READ(IU) A1, A2                                                   GETT0170
      RETURN                                                            GETT0180
C                                                                       GETT0190
C  READ IDUM AND A1                                                     GETT0200
   40 READ(IU) IDUM, A1                                                 GETT0210
      RETURN                                                            GETT0220
      END                                                               GETT0230
```

```
*DECK BAKSUB                                                            BAKS0010
      SUBROUTINE BAKSUB(TRANS,SUMMER,NEWMAX)                            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,JETELG,IGTYPE,IHINGE            BAKS0080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40) BAKS0090
      COMMON/SOLV1/GAMMA(400,1)                                         BAKS0095
      DIMENSION TRANS(NEWMAX),SUMMER(NEWMAX)                            BAKS0100
C                                                                       BAKS0110
      WRITE(6, 10 )                                                     BAKS0120
   10 FORMAT(1H1,47X,24HBACK SUBSTITUTION SOLUTION)                     BAKS0130
C                                                                       BAKS0140
C  CYCLE THE RIGHT HAND SIDES                                           BAKS0150
      DO 60  N = 1,NCASES                                               BAKS0160
      IREAD = 0                                                         BAKS0170
C                                                                       BAKS0180
C  CYCLE THE MATRIX ROWS CORRESPONDING TO ELEMENTS ON THE WING          BAKS0190
      DO 40  I = 1,NEWMAX                                               BAKS0200
C  READ THE COEFFICIENT MATRIX ROW                                      BAKS0210
      IF(I .EQ. NWT+1) IREAD = NMAX                                     BAKS0220
      IREAD = IREAD + 1                                                 BAKS0230
   20 CALL READMS(1,TRANS,NEWMAX,IREAD)                                BAKS0240
C                                                                       BAKS0250
C  SUM UP THE TERMS FOR THIS ROW AND RIGHT SIDE                         BAKS0260
      SUMMER(I) = 0.00                                                  BAKS0270
      DO 30  J = 1,NEWMAX                                               BAKS0280
      SUMMER(I) = SUMMER(I) + TRANS(J) * GAMMA(J,N)                     BAKS0290
   30 CONTINUE                                                          BAKS0300
   40 CONTINUE                                                          BAKS0310
C                                                                       BAKS0320
C  PRINT THE NTH RIGHT SIDE COLUMN                                      BAKS0330
      WRITE(6, 50 ) N,SUMMER                                            BAKS0340
   50 FORMAT(1HC,50X,17HRIGHT SIDE COLUMN,I4,40(/1X,10E13.5))           BAKS0350
   60 CONTINUE                                                          BAKS0360
      RETURN                                                            BAKS0365
      END                                                               BAKS0370
                                                                        BAKS0380
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK STAGE3                                                            STG30C10
      OVERLAY(OVLY,3,0)                                                 STG30020
      PROGRAM STAGE3                                                    STG30030
C                                                                       STG30040
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL LOADINGS FOR THE         STG30050
C  FUNDAMENTAL AND COMPOSITE CASES                                      STG30060
C                                                                       STG30070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           STG30079
      COMMON/MARK/NROWS,NPROWS,J,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) STG30080
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CRAR,XMC,XCG     STG30090
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALEA,LOGIC,IC                      STG30100
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(4C0),XI(6C0),DEL(6C0),  STG30110
     1          D(6C),XK(4C),ITYPE(4C)                                 STG30120
      COMMON/GEOM2/XLEAD(4C),XTRAIL(4C),TANLE(4C),TANTE(4C)             STG30130
      COMMON/ECASE1/EPS(4CC,1C),BETA(4C,1C),THETA(4C,1C),THS(40,10)     STG30140
      COMMON/COMPOS/FACTOR(10,24),NCC                                   STG30150
      COMMON/ICASE/CMU(4C),CMUPD(4C),CMUPP(4C)                          STG30160
      COMMON /DERIV/ CL(4C),CLO,CMO,CMOMC                               STG30170
      DIMENSION CPREAD(4C1C),CPR(6CC),CPA(6CC),CPRA(6CC),CPPA(6CC),     STG30180
     1          CPP(6CC)                                                STG30190
      EQUIVALENCE (BETA(1,1),CPC(1)),(BETA(1,2),CPA(1)),                STG30200
     1    (BETA(1,3),CPR(1)),(BETA(1,4),CPRA(1)),(BETA(1,5),CPP(1))     STG30210
C                                                                       STG30220
      IF(LOGIC .EQ. 3) GO TO 80                                         STG30230
C  CALCULATE AND PRINT THE LOADING FOR ALL FUNDAMENTAL CASES            STG30240
   10 CALL STG3EC(NEWMAX)                                               STG30250
      DO 30  N = 1,NCASES                                               STG30260
      LCASE = N                                                         STG30270
   20 CALL STG3ES(CLO,CMO,CMOMC,DUM4,NEWMAX,NOALEA,LCASE)              STG30280
   30 CONTINUE                                                          STG30290
      IF(LOGIC .EQ. 2) WRITE(4, 40 ) CLO,CMO,CMOMC                      STG30300
   40 FORMAT(1HC/ 24X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,   STG30310
     1       17H ABOUT XCG, CL0 =,E10.6 /                               STG30320
     2       14X,51HPITCHING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN, STG30330
     3       33H DUE TO PITCHING ABOUT XCG, CMO =,E10.6 /               STG30340
     4       16X,47HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,         STG30350
     5       33H DUE TO PITCHING ABOUT XCG, CMOMC =, E10.6)             STG30360
      CALL STG3ET                                                       STG30370
C                                                                       STG30380
C  CALCULATE AND PRINT THE LOADING FOR ALL COMPOSITE CASES              STG30390
      IF(NCC .LT. 1) GO TO 100                                          STG30400
      DO 60  N = 1,NCC                                                  STG30410
      NCASE = N                                                         STG30420
   50 CALL STG3C(NEWMAX,NCASE,NOALEA)                                   STG30430
   60 CONTINUE                                                          STG30440
   70 GO TO 100                                                         STG30450
C                                                                       STG30460
C  CALCULATE AND PRINT THE COEFFICIENTS AND DERIVATIVES FOR ALL         STG30470
C  FUNDAMENTAL CASES                                                    STG30480
   80 CALL FUNDPE(EPS,CPC,CPA,CPRC,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,AREA, STG30490
     1       CLO,CMO,CMOMC,CLLO,CN02,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALEA, STG30500
     2       NROWS,ISYMM,XLEAD,TANLE,XMC)                               STG30510
C                                                                       STG30520
C  CALCULATE AND PRINT THE STABILITY DERIVATIVES FOR ALL COMPOSITE CASES STG30530
   90 CALL COMPE(EPS,CPC,CPA,CPRC,CPRA,CPP,CPREAD,DEL,CHORD,Y,CMU,      STG30540
     1       DELTA,AREA,CL0,CMO,CMOMC,CLLO,CN02,NW,IJ,NMAX,NJT,NEWMAX,  STG30550
     2       NCASES,NROWS,ISYMM,XLEAD,TANLE,XMC)                        STG30560
  100 CONTINUE                                                          STG30570
      END                                                               STG30580
```

```
*DECK STG3EC                                                            S3EC0C10
      SUBROUTINE STG3EC(NEWMAX)                                         S3EC0020
C                                                                       S3EC0030
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE LOADING            S3EC0040
C  FOR FUNDAMENTAL CASES                                                S3EC0050
C                                                                       S3EC0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE           S3EC0069
      COMMON/MARK/NROWS,NPROWS,J,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) S3EC0070
      COMMON/GEOM1/Y(4C),CHORD(4C),DELTA(4C),XR(4CC),XI(6CC),DEL(6CC),  S3EC0080
     1          D(6C),XK(4C),ITYPE(4C)                                 S3EC0090
      COMMON/ECASE1/EPS(4CC,1C),BETA(4C,1C),THETA(4C,1C),THS(4C,1C)     S3EC0100
      COMMON/ICASE/CMU(4C),CMUPD(4C),CMUPP(4C)                          S3EC0110
      COMMON/SOLV1/CP(6CC,1C)                                           S3EC0120
      DIMENSION XRR(5),CPEXP(6,1C)                                      S3EC0130
C                                                                       S3EC0140
C  CALCULATE AND PRINT THE CHORDWISE LOADING OF THE FUNDAMENTAL CASES   S3EC0150
C                                                                       S3EC0160
C  INITIALIZE THE UNUSED VALUES OF CP FOR PRINTING                      S3EC0170
      NC1 = NCASES + 1                                                  S3EC0180
      IF(NC1 .GT. 1C) GO TO 30                                          S3EC0190
      DO 25 N = NC1,1C                                                  S3EC0200
      DO 15 I = 1,NEWMAX                                                S3EC0210
      CP(I,N) = 0.0                                                     S3EC0220
   15 CONTINUE                                                          S3EC0230
   25 CONTINUE                                                          S3EC0240
   30 I = 0                                                             S3EC0250
      II = NWT                                                          S3EC0260
      DO 32C M = 1,NPROWS                                               S3EC0270
      IF(IPRINT .GT. 0) GO TO 70                                        S3EC0280
      WRITE(6, 40 )                                                     S3EC0290
   40 FORMAT(1HI,35X,12(4H****)/1H*/                                    S3EC0300
     1       34X,43H* CHORDWISE LOADING FOR ALL FUNDAMENTAL CASES  */   S3EC0310
     2       36X,12(4H****)/1HO)                                        S3EC0320
      WRITE(6, 50 ) M,Y(M),CHORD(M),(N,N=1,12)                          S3EC0330
   50 FORMAT(1HC,35X,7HSECTION,I3,5X,2HY =,F10.6,5X,7HCHORD =,F10.4/    S3EC0340
     1       8X,1HI,5X,2HXR,3X,3(5X,4HCASE,I2),5X,4HCASE,I3)            S3EC0350
C                                                                       S3EC0360
C  ON THE WING                                                          S3EC0370
      WRITE(6, 60 )                                                     S3EC0380
   60 FORMAT(1H0,14HWING)                                               S3EC0390
   70 NWK = NW(K)                                                       S3EC0400
      DO 100 L = 1,NWK                                                  S3EC0410
      I = I + 1                                                         S3EC0420
      DO 80 N = 1,NCASES                                                S3EC0430
      CP(I,N) = 2.00 * CP(I,N)                                          S3EC0440
   80 CONTINUE                                                          S3EC0450
      IF(IPRINT .LT. 1) WRITE(6, 90 ) I,XR(I),(CP(I,N),N=1,1C)          S3EC0460
   90 FORMAT(1H ,I3,11F11.6)                                            S3EC0470
  100 CONTINUE                                                          S3EC0480
      ILINES = NWK + 4                                                  S3EC0490
C                                                                       S3EC0500
C  ON THE JET                                                           S3EC0510
      NJK = NJ(K)                                                       S3EC0520
      IF(CHUIK) .LT. 0.0001) GO TO 140                                  S3EC0530
      IF(IPRINT .LT. 2) WRITE(6, 110 )                                  S3EC0540
  110 FORMAT(1H ,1X,3HJET)                                              S3EC0550
      DO 130 L = 1,NJK                                                  S3EC0560
      II = II + 1                                                       S3EC0570
      DO 120 N = 1,NCASES                                               S3EC0580
      CP(II,N) = 2.00 * CP(II,N)                                        S3EC0590
  120 CONTINUE                                                          S3EC0600
      IF(IPRINT .LT. 1) WRITE(6, 90 ) II,XR(II),(CP(II,N),N=1,1C)       S3EC0610
  130 CONTINUE                                                          S3EC0620
      ILINES = ILINES + NJK + 1                                         S3EC0630
C                                                                       S3EC0640
C  PRINT THE DETAILED LOADING ON THE SINGULAR ELEMENTS                  S3EC0650
C                                                                       S3EC0660
C  LEADING EDGE                                                         S3EC0670
  140 IF(IPRINT .GT. 0) GO TO 320                                       S3EC0680
      WRITE(6, 150 )                                                    S3EC0690
  150 FORMAT(1HC,45X,29HDETAILED LEADING EDGE LOADING)                  S3EC0700
      IP = IW(K)                                                        S3EC0710
      DO 160 N = 1,NCASES                                               S3EC0720
      LCASE = N                                                         S3EC0730
      CALL EXPLE(LCASE,CP(IP,LCASE),CP(IP+1,LCASE),DEL(IP),XRR,CPEXP)  S3EC0740
  160 CONTINUE                                                          S3EC0750
      DO 170 M = 1,5                                                    S3EC0760
      WRITE(6, 90 ) M,XRR(M),(CPEXP(M,N),N=1,NCASES)                    S3EC0770
  170 CONTINUE                                                          S3EC0780
      ILINES = ILINES + 7                                               S3EC0790
C                                                                       S3EC0800
C  HINGES                                                               S3EC0810
      IF(IHINGE .EQ. 0) GO TO 320                                       S3EC0820
      J = IW(K) - 1                                                     S3EC0830
      DO 260 L = 1,NWK                                                  S3EC0840
      J = J + 1                                                         S3EC0850
      IF(ITYPE(J) .LT. 4C) GO TO 250                                    S3EC0860
      DO 180 N = 1,NCASES                                               S3EC0870
      LCASE = N                                                         S3EC0880
      CALL EXPH1(LCASE,CP(J,N),CP(J-1,N),DEL(J-1),BETA(J,N),           S3EC0890
     1           CHORD(K),XR(J),XRR,CPEXP)                              S3EC0900
  180 CONTINUE                                                          S3EC0910
      IF(ILINES .LT. 46) GO TO 200                                      S3EC0920
      WRITE(6, 190 )                                                    S3EC0930
  190 FORMAT(1H1)                                                       S3EC0940
      ILINES = 1                                                        S3EC0950
  200 WRITE(6, 210 ) J                                                  S3EC0960
  210 FORMAT(1HC,42X,33HDETAILED HINGE LOADING ON ELEMENT,I6)           S3EC0970
      DO 220 M = 1,5                                                    S3EC0980
      WRITE(6, 90 ) M,XRR(M),(CPEXP(M,N),N=1,NCASES)                    S3EC0990
  220 CONTINUE                                                          S3EC1000
      DO 230 N = 1,NCASES                                               S3EC1010
      LCASE = N                                                         S3EC1020
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),             S3EC1030
     1           CHORD(K),XR(J),XRR,CPEXP)                              S3EC1040
  230 CONTINUE                                                          S3EC1050
      DO 240 M = 6,10                                                   S3EC1060
      WRITE(6, 90 ) M,XRR(M-5),(CPEXP(M-5,N),N=1,NCASES)                S3EC1070
  240 CONTINUE                                                          S3EC1080
      ILINES = ILINES + 12                                              S3EC1090
  250 CONTINUE                                                          S3EC1100
      IF(NJ(K) .EQ. 0) .OR.(CMU(K) .LT. 0.0001) GO TO 320               S3EC1110
      J = IJK                                                           S3EC1120
      IF(ITYPE(J) .NE. 43) GO TO 320                                    S3EC1130
      DO 260 N = 1,NCASES                                               S3EC1140
      LCASE = N                                                         S3EC1150
      II = IW(K) + NW(K) - 1                                            S3EC1160
      CALL EXPH1(LCASE,CP(J,N),CP(II,N),DEL(II),BETA(J,N),             S3EC1170
     1           CHORD(K),XR(J),XRR,CPEXP)                              S3EC1180
  260 CONTINUE                                                          S3EC1190
      IF(ILINES .LT. 46) GO TO 270                                      S3EC1200
      WRITE(6, 190 )                                                    S3EC1210
      ILINES = 1                                                        S3EC1220
  270 WRITE(6, 280 ) J                                                  S3EC1230
  280 FORMAT(1HC,4X,37HDETAILED JET HINGE LOADING ON ELEMENT,I4)        S3EC1240
      DO 290 M = 1,5                                                    S3EC1250
      WRITE(6, 90 ) M,XRR(M),(CPEXP(M,N),N=1,NCASES)                    S3EC1260
  290 CONTINUE                                                          S3EC1270
      DO 310 N = 1,NCASES                                               S3EC1280
      LCASE = N                                                         S3EC1290
      CALL EXPH2(LCASE,CP(J,N),CP(J+1,N),DEL(J),BETA(J,N),             S3EC1300
     1           CHORD(K),XR(J),XRR,CPEXP)                              S3EC1310
  300 CONTINUE                                                          S3EC1320
      DO 310 M = 6,10                                                   S3EC1330
      WRITE(6, 90 ) M,XRR(M-5),(CPEXP(M-5,N),N=1,NCASES)                S3EC1340
  310 CONTINUE                                                          S3EC1350
      ILINES = ILINES + 12                                              S3EC1360
  320 CONTINUE                                                          S3EC1370
      RETURN                                                            S3EC1380
      END                                                               S3EC1390
```

C15

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

C16

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK STG3ES                                                                    S3ES0010
      SUBROUTINE STG3ES(CLQ,CMQ,CMQMC,CLLP,NEWMAX,NOALFA,LC)                    S3ES0020
C                                                                               S3ES0030
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL SPANWISE AND TOTAL               S3ES0040
C  LOADING FOR FUNDAMENTAL CASES                                                S3ES0050
C                                                                               S3ES0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE                    S3ES0070
      COMMON/MAREA/NROWS,NPOWS(4),NUT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)      S3ES0080
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG             S3ES0090
      COMMON/GEO41/YY(40),CHORD(40),DELTA(40),XB(40),XJ(40),DEL(600),           S3ES0100
     1             DY(40),KK(40),ITYPE(40)                                      S3ES0110
      COMMON/GEO42/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)                     S3ES0120
      COMMON/ECASE2/TST(40,4),HE(40,4),DJ(40),ARTE(40),AC(20,40),               S3ES0130
     1             XHP(4,40),BET(4,40),IFS(4,40),ICT(40),IHT(40),NCT,NHT        S3ES0140
      COMMON/ECASE3/CP3(400,10),BETA(400,10),THETA(40,10),THS(40,10)            S3ES0150
      COMMON/JCASE/CMU(40),CMUP(40),CMUPP(40)                                   S3ES0160
      COMMON/SOLVE/CP(400,10)                                                   S3ES0170
      COMMON/LOAD1/ TWIST(40),HE(40),THE(40),THETS(40),                         S3ES0180
     1             RTA(600),EP(400),CPD(600)                                    S3ES0190
      COMMON/LOAD2/ CLG(40),CLMU(40),CL(40),CDMU(40),CDG(40),CDI(40),           S3ES0200
     1             CS(40),CMG(40),CMMU(40),CMT(40),CM(40),XBCP(40),XBCL(40)     S3ES0210
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),                S3ES0220
     1      CCMT(10),CCM(10),CMCMC(10),CMJMC(10),CMTMC(10),CMMC(10),            S3ES0230
     2      CXCP(10),CXCL(10),CCDJ(10),CCDG(10),CCDJJ(10),CCS(10),CCDI(10),     S3ES0240
     3      CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNT(10),CCY(10),        S3ES0250
     4      CXCPR(10),CXCLR(10)                                                 S3ES0260
      COMMON/LOAD4/ CLG(40),CLMU(40),CLQ(40),CDMU(40),CDQ(40),                  S3ES0270
     1      CDI(40),CS(40),CTS(40),CMQ(40),CMMU(40),CMT(40),CM(40),             S3ES0280
     2      XBCP(40),XBCL(40),FACT(10)                                          S3ES0290
      COMMON/LOAD5/ CGAM(40),CGAMC(40),ALFINF(40),ALFINC(40),DUMB(40)           S3ES0300
      COMMON/LOAD7/CBPR(10),CBPL(10),CBJR(10),CBJL(10),                         S3ES0310
     1             CBR(10),CBL(10),CBMR(10),CBML(10),CL2R(10),CL2L(10)          S3ES0320
C                                                                               S3ES0330
   10 N = LC                                                                    S3ES0340
      LCASE = LC                                                                S3ES0350
C                                                                               S3ES0360
C  LOAD THOSE GEOMETRIC PARAMETERS WHICH ARE DIFFERENT FOR EACH CASE            S3ES0370
C  INTO THEIR RESPECTIVE DUMMY ARRAYS                                           S3ES0380
      DO 30  K = 1,NROWS                                                        S3ES0390
      TWIST(K) = TST(K,LCASE)                                                   S3ES0400
   20 THE(K) = THETA(K,LCASE)                                                   S3ES0410
      THETS(K) = THS(K,LCASE)                                                   S3ES0420
      HE(K) = HE(K,LCASE)                                                       S3ES0430
      DUMB(K) = 0.00                                                            S3ES0440
   30 CONTINUE                                                                  S3ES0450
      DO 40  I = 1,NEWMAX                                                       S3ES0460
      RTA(I) = RTA(I,LCASE)                                                     S3ES0470
      EP(I) = EPS(I,LCASE)                                                      S3ES0480
      CPD(I) = CP(I,LCASE)                                                      S3ES0490
   40 CONTINUE                                                                  S3ES0500
C                                                                               S3ES0510
C  COMPUTE SECTIONAL COEFFICIENTS                                               S3ES0520
      ALPHA = 0.00                                                              S3ES0530
      IF(LCASE.EQ. 1) .AND. (NOALFA .NE. 0)) ALPHA = 1.00                       S3ES0540
   60 CALL SLOADC(ALPHA,IJ,NW,N,CHORD,CMU,TH,THETS,TWIST,                       S3ES0550
     1      XB,DEL,RTA,EP,CPD,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XBCP,XBCL,           S3ES0560
     2      CDI,CDMU,CDG,CS,CTS,NROWS,IHINGE)                                   S3ES0570
C                                                                               S3ES0580
C  COMPUTE SECTIONAL VORTICITY OF WING-JET SYSTEM                               S3ES0590
      CALL SLOADG(CPD,DEL,RTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)          S3ES0600
C                                                                               S3ES0610
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY                                       S3ES0620
   70 CALL TREFTZ(Y,DELTA,CMU,CGAM,ALFINF,NROWS,ISYMM)                          S3ES0630
C                                                                               S3ES0640
C  COMPUTE TOTAL LINEAR COEFFICIENTS                                            S3ES0650
   80 CALL TLOADC(ALPHA,AREA,CREF,XMC,Y,DELTA,CHORD,HC,XB,XLEAD,RTA,            S3ES0660
     1      CLG,CLMU,CMG,CMMU,CMT,CMU,CCLG(N),CCLJ(N),CCL(N),CCMG(N),           S3ES0670
     2      CCMJ(N),CCMT(N),CCM(N),CMCMC(N),CMJMC(N),CMTMC(N),CMMC(N),          S3ES0680
     3      CXCP(N),CXCL(N),CXCPR(N),CXCLR(N),CCJJ(N),CLLG(N),CLLJ(N),          S3ES0690
     4      CLL(N),ITYPE,IW,NW,NROWS,ISYMM,                                     S3ES0700
     5      CBJR(N),CBJL(N),CBR(N),CBL(N),                                      S3ES0710
     6      CBMR(N),CBML(N),CL2R(N),CL2L(N))                                    S3ES0720
C                                                                               S3ES0730
C  COMPUTE TOTAL NONLINEAR COEFFICIENTS                                         S3ES0740
   90 CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,DUMB,               S3ES0750
     1   CCDG(N),CCDJJ(N),CCS(N),CCDI(N),CDITZ(N),DUMB,ALFINF,DUMMY,CCJJ(1),    S3ES0760
     2      CNJ(N),CNT(N),CCY(N),XLEAD,TANLE,XMC,NROWS,ISYMM)                   S3ES0770
C                                                                               S3ES0780
C  DEFINE THE CONSTANT STABILITY DERIVATIVES                                    S3ES0790
      CLQ = CCLG(NCASES)                                                        S3ES0800
      CMQ = CCMG(NCASES)                                                        S3ES0810
      CMQMC = CMGMC(NCASES)                                                     S3ES0820
      CLLP = CLLG(NCASES)                                                       S3ES0830
C                                                                               S3ES0840
C  PRINT LIFT AND DRAG COEFFICIENTS                                             S3ES0850
      IF(IPRINT .GT. 1) RETURN                                                  S3ES0860
      WRITE(6, 100 ) LCASE                                                      S3ES0870
  100 FORMAT(1H1,36X,11(4H****),2H**/37X,23H*  SPANWISE LOADING FOR,            S3ES0880
     1       17H FUNDAMENTAL CASE,I3,3H  */37X,11(4H****),2H**)               S3ES0890
      WRITE(6, 110 )                                                            S3ES0900
  110 FORMAT(1H ,19X,29H.......... LIFT ..........,5X,                          S3ES0910
     1       2H.,7(4H....),14H INDUCED DRAG ,7(4H....))                       S3ES0920
     2       1X, 7HSECTION, 5X,1HY,8X,3HCLG,7X,4HCLMU,4X,2HCL,5X,             S3ES0930
     3       4H  ,3X,3HCDG,8X,4HCDMU,7X,2HCS,9X,2HCD,9X,3HCMU,                S3ES0940
     4       8X,5HGAMMA,6X,5HALFIN)                                           S3ES0950
      WRITE(6, 120 ) (K,YY(K),CLG(K),CLMU(K),CL(K),CDG(K),CDMU(K),              S3ES0960
     1                CS(K),CDI(K),CMU(K),CGAM(K),ALFINF(K),K=1,NROWS)        S3ES0970
  120 FORMAT(1H ,I4, 4X, 4F10.5, 4H  *  ,7F11.7)                                S3ES0980
      WRITE(6, 130 ) CCLG(N),CCLJ(N),CCL(N),CCDG(N),CCDJJ(N),CCS(N),            S3ES0990
     1               CCDI(N),CNJ(N),CDITZ(N)                                  S3ES1000
  130 FORMAT(1H , 8X, 4(1CH ----------),4H  *  ,7(11H ----------)/              S3ES1010
     1       12X,5HTOTAL,2X,3F10.4,4H  *  ,5F11.7,11X,F11.7)                  S3ES1020
C                                                                               S3ES1030
C  PRINT PITCHING MOMENT AND CENTER OF LIFT DATA                                S3ES1040
      IF(NROWS .GT. 21) WRITE(6, 140 )                                          S3ES1050
  140 FORMAT(1H1)                                                               S3ES1060
      WRITE(6, 150 )                                                            S3ES1070
  150 FORMAT(1HC,24X,39H..... PITCHING MOMENT .....10X,
     1       2CH  ... LIFT CENTER ...,/                                         S3ES1080
     2       6X, 7HSECTION,5X,1HY,8X,3HCMG,7X,4HCMMU,6X,3HCMT,7X,2HCM,          S3ES1090
     3       5X,10H   *  *  ,3X,5HXCP/C,5X,5HXCL/C)                            S3ES1100
      WRITE(6, 160 ) (K,YY(K),CMG(K),CMMU(K),CMT(K),CM(K),XBCP(K),XBCL(K)       S3ES1110
     1,               K=1,NROWS)                                               S3ES1120
  160 FORMAT(1H ,I5,4X,5F10.4,10H   *  *  ,2F10.4)                             S3ES1130
      WRITE(6, 170 ) CCMG(N),CCMJ(N),CCMT(N),CCM(N),CXCP(N),CXCL(N),            S3ES1140
     1               CMGMC(N),CMJMC(N),CMTMC(N),CMMC(N),CXCPR(N),CXCLR(N)      S3ES1150
  170 FORMAT(1H ,13X, 5(1CH ----------),10H   *  *  ,2(1CH ----------)/        S3ES1160
     1       17X,5HTOTAL,2X, 4F10.4, 10H   (APEX) ,2F10.4,9H  (X/CREF)/       S3ES1170
     2       24X,6F10.4,17H  (XMC)   ,2F10.4,9H  (X/B/2))                     S3ES1180
      RETURN                                                                    S3ES1190
      END                                                                       S3ES1200
*DECK STG3ET                                                                    S3ET0010
      SUBROUTINE STG3ET                                                         S3ET0020
C                                                                               S3ET0030
C  THIS SUBROUTINE PRINTS A TABLE OF ALL TOTAL COEFFICIENTS                     S3ET0040
C  FOR ALL FUNDAMENTAL CASES                                                    S3ET0050
C                                                                               S3ET0060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE                    S3ET0070
      COMMON/LOAD3/ CCLG(10),CCLJ(10),CCL(10),CCMG(10),CCMJ(10),                S3ET0080
     1      CCMT(10),CCM(10),CMCMC(10),CMJMC(10),CMTMC(10),CMMC(10),            S3ET0090
     2      CXCP(10),CXCL(10),CCDJ(10),CCDG(10),CCDJJ(10),CCS(10),CCDI(10),     S3ET0100
     3      CDITZ(10),CLLG(10),CLLJ(10),CLL(10),CNJ(10),CNT(10),CCY(10),        S3ET0110
     4      CXCPR(10),CXCLR(10)                                                 S3ET0120
      COMMON/LOAD6/COEFF(35)                                                    S3ET0130
      COMMON/LOAD7/CBPR(10),CBPL(10),CBJR(10),CBJL(10),                         S3ET0140
     1             CBR(10),CBL(10),CBMR(10),CBML(10),CL2R(10),CL2L(10)          S3ET0150
C                                                                               S3ET0160
C  INITIALIZE ALL UNUSED TOTAL COEFFICIENTS TO ZERO FOR PRINTING                S3ET0170
      IF(NCASES .EQ. 10) GO TO 20                                               S3ET0180
      NC1 = NCASES + 1                                                          S3ET0190
      DO 10  N = NC1,10                                                         S3ET0200
      CCLG(N) = 0.00                                                            S3ET0210
      CCLJ(N) = 0.00                                                            S3ET0220
      CCL(N) = 0.00                                                             S3ET0230
      CCMG(N) = 0.00                                                            S3ET0240
      CCDJ(N) = 0.00                                                            S3ET0250
      CCS(N) = 0.00                                                             S3ET0260
      CCDI(N) = 0.00                                                            S3ET0270
      CDITZ(N) = 0.00                                                           S3ET0280
      CLLG(N) = 0.00                                                            S3ET0290
      CCMG(N) = 0.00                                                            S3ET0300
      CCMJ(N) = 0.00                                                            S3ET0310
      CCMT(N) = 0.00                                                            S3ET0320
      CCM(N) = 0.00                                                             S3ET0330
      CXCP(N) = 0.00                                                            S3ET0340
      CXCL(N) = 0.00                                                            S3ET0350
      CXCPR(N) = 0.00                                                           S3ET0360
      CXCLR(N) = 0.00                                                           S3ET0370
      CMGMC(N) = 0.00                                                           S3ET0380
      CMJMC(N) = 0.00                                                           S3ET0390
      CMTMC(N) = 0.00                                                           S3ET0400
      CMMC(N) = 0.00                                                            S3ET0410
      CCLG(N) = 0.00                                                            S3ET0420
      CLLJ(N) = 0.00                                                            S3ET0430
      CLL(N) = 0.00                                                             S3ET0440
      CNJ(N) = 0.00                                                             S3ET0450
      CNT(N) = 0.00                                                             S3ET0460
      CCY(N) = 0.00                                                             S3ET0470
      CBGR(N) = 0.00                                                            S3ET0480
      CBGL(N) = 0.00                                                            S3ET0490
      CBJR(N) = 0.00                                                            S3ET0500
      CBJL(N) = 0.00                                                            S3ET0510
      CBR(N) = 0.00                                                             S3ET0520
      CBL(N) = 0.00                                                             S3ET0530
      CBMR(N) = 0.00                                                            S3ET0540
      CBML(N) = 0.00                                                            S3ET0550
   10 CONTINUE                                                                  S3ET0560
   20 WRITE(6, 30 ) (N,N=1,10)                                                  S3ET0570
   30 FORMAT(1H1,41X, 9(4H****)/                                                S3ET0580
     1       42X,36H*  TOTAL AERODYNAMIC COEFFICIENTS  */42X,9(4H****)//        S3ET0590
     2       14X,9(4HCASE,I2,5X), 4HCASE,I3)                                    S3ET0600
   40 FORMAT(1HC, 2X,A8,10F11.7,34(/3X,A8,10F11.7))                             S3ET0610
      WRITE(6, 40 ) COEFF( 1), CCLG, COEFF( 2), CCLJ, COEFF( 3),  CCL,          S3ET0620
     1              COEFF( 4), CCMG, COEFF( 5), CCMJ, COEFF( 6),  CCS,          S3ET0630
     2              COEFF( 7), CCDI, COEFF( 8),CDITZ, COEFF( 9),  CCJJ,         S3ET0640
     3              COEFF(10), CCMG, COEFF(11), CCMJ, COEFF(12),  CCMT,         S3ET0650
     4              COEFF(13),  CCM, COEFF(14), CXCP, COEFF(15),  CXCL,         S3ET0660
     5              COEFF(16),CXCPR, COEFF(17),CXCLR, COEFF(18),CMGMC,          S3ET0670
     6              COEFF(19),CMJMC, COEFF(20),CMTMC, COEFF(21),  CMMC,         S3ET0680
     7              COEFF(22), CLLG, COEFF(23), CLLJ, COEFF(24),  CLL,          S3ET0690
     8              COEFF(25),  CNJ, COEFF(26),  CNT, COEFF(27),  CCY,          S3ET0695
     9       COEFF(28),CBGR,COEFF(29),CBGL,COEFF(30),CBJR,                      S3ET0700
     *       COEFF(31),CBJL,COEFF(32),  CBR,COEFF(33),CBL,                      S3ET0710
     1       COEFF(34),CBMR,COEFF(35),CBML                                      S3ET0720
C                                                                               S3ET0730
   50 RETURN                                                                    S3ET0740
      END                                                                       S3ET0750
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
CHECK STG3C
      SUBROUTINE STG3C(NEWMAX,M,NOALFA)                                 ST3C0010
C                                                                       ST3C0020
C  THIS SUBROUTINE CONTROLS CALCULATION OF CHORDWISE, SPANWISE AND      ST3C0030
C  TOTAL LOADING FOR THE REQUIRED COMPOSITE CASES                       ST3C0040
C                                                                       ST3C0050
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,ICTYPE,IHINGE           ST3C0060
      COMMON/MARK/NROWS,NPANSJ,NWT,NJT,NMAX,NW(4C),NJ(4C),IW(4C),IJ(4C) ST3C0070
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CBAR,XMC,XCG     ST3C0080
      COMMON/GEOM1/YV(4C),CHORD(4C),DELTA(4C),XR(4CC),XI(4CC),DEL(6CC),  ST3C0090
     1          DI(43),KK(4C3),ITYPE(43C)                               ST3C0100
      COMMON/GEOM2/YLEAD(4C),XTAIL(4C),TANLE(4C),TANTE(4C)              ST3C0110
      COMMON/ECASE2/FST(4C,1C),HL(4C,1C),DJT(4C),ACTE(4C),AC(3C,6C),    ST3C0120
     1          XHR(4,4C),RT(4,4C),ES(4,4C),ICT(4C),THT(4C),NCT,NHT     ST3C0130
      COMMON/ECASE3/EPS(4CC,1C),RETA(4CC,1C),THETA(4C,1C),THS(4C,1C)    ST3C0140
      COMMON/JCASE/CHI(4C),CHIP(4C),CHIPP(4C)                           ST3C0150
      COMMON/SOLVE/CP(4CC,1C)                                           ST3C0160
      COMMON/COMPOS/FACTOR(1C,24),NCC                                   ST3C0170
      COMMON/LOAD1/ TWIST(4C),XP(4C),TH(4C),THETS(4C),                  ST3C0180
     1          RTA(4CC),EP(4CC),CPO(4CC)                               ST3C0190
      COMMON/LOAD2/ CLG(4C),CLMU(4C),CL(4C),CDMU(4C),CDG(4C),CDI(4C),   ST3C0200
     1          CS(4C),CMC(4C),CMWU(4C),CMI(4C),CMC4C),XPCP(5C),XRCL(4C)ST3C0210
      COMMON/LOAD3/ CCLG(1C),CCLJ(1C),CCLL(1C),CCMG(1C),CCMJ(1C),       ST3C0220
     1          CCMT(1C),CCMC(1C),CMGMC(1C),CMJMC(1C),CMTMC(1C),CMMC(1C),ST3C0230
     2          CXCP(1C),CXCL(1C),CC(1C),CCDG(1C),CCDJ(1C),CCSJ(1C),CCD(1C),ST3C0240
     3          CDITT(1C),CLLG(1C),CLLJ(1C),CLL(1C),CMJ(1C),CMT(1C),CCY(1C),ST3C0250
     4          CXCPR(1C),CXCLR(1C)                                     ST3C0260
      COMMON/LOAD4/ CLG(4C),CLMU(4C),CL(4C),CDMU(4C),CDG(4C),           ST3C0270
     1          CDI(4C),CS(4C),CT(4C),CMG(4C),CMMU(4C),CMT(4C),CM(4C),ST3C0280
     2          XRCP(4C),XRCL(4C),FACT(1C)                              ST3C0290
      COMMON/LOAD5/ CGAM(4C),CSAM(4C),ALEINF(4C),ALEINO(4C),DUMB(4C)    ST3C0300
      COMMON/LOAD6/COEFF(35)                                            ST3C0310
      COMMON/LOAD7/CMG2(1C),CMGL(1C),CMJR(1C),CMJL(1C),                 ST3C0320
     1          CMR(1C),CML(1C),CMMR(1C),CMML(1C),CL2R(1C),CL2L(1C)     ST3C0330
      DIMENSION CDMUX(4C),CDGX(4C),CDIX(4C),CSX(4C),CPA(4CC)            ST3C0340
      EQUIVALENCE (PP(1),CPA(1))                                        ST3C0350
C                                                                       ST3C0360
C  CALCULATE AND PRINT THE CHORDWISE LOADING FOR ALL COMPOSITE CASES    ST3C0370
C                                                                       ST3C0380
      I = C                                                             ST3C0390
      L = C                                                             ST3C0400
      II = NWT                                                          ST3C0410
      ILINES = 6                                                        ST3C0420
      WRITE(6, 1C ) M                                                   ST3C0430
   1C FORMAT(1H1,36X,111(4H****),1H*/                                   ST3C0440
     1          37X,4HM=  ,I2,  CHORDWISE LOADING FOR COMPOSITE CASE,I3,3H  */ST3C0450
     2          37X,111(4H****),1H*)                                    ST3C0460
      WRITE(6, 2C ) (N,N=1,1C),(FACTOR(N,M),N=1,1C)                     ST3C0470
   2C FORMAT(1H ,49X,24HFUNDAMENTAL CASE FACTORS/1CX,9(5X,2HA(,I1,1H),  ST3C0480
     1          3X),5X,2HA(,I2,1H)/1CX,1CF12.4)                         ST3C0490
      WRITE(6, 3C )                                                     ST3C0500
   3C FORMAT(1HC,7X,47HX** NOTE ***  EACH LEADING EDGE CP VALUE IS THE, ST3C0510
     1          43H AVERAGE VALUE OF THE SINGULAR DISTRIBUTION)         ST3C0520
      IF(IHINGE .NE. C) WRITE(6, 4C )                                   ST3C0530
   4C FORMAT(1H ,7IX,47HIF A HINGE IS DEFLECTED THE LOADING IS SINGULAR, ST3C0540
     1          43H AND THE CP(A=C) VALUE IS FOR THE REGULAR EVD PORTION ONLY)ST3C0550
      WRITE(6, 5C )                                                     ST3C0560
   5C FORMAT(1H ,21X,41HDO NOT PLOT THESE LOADING POINTS DIRECTLY)      ST3C0570
      DO 22C K = 1,NROWS                                                ST3C0580
C                                                                       ST3C0590
C  ON THE WING                                                          ST3C0600
      NWK = NW(K)                                                       ST3C0610
      DO 8C  L = 1,NWK                                                  ST3C0620
      I = I + 1                                                         ST3C0630
   6C CP(I) = C.CC                                                      ST3C0640
      DO 7C  N = 1,NCASES                                               ST3C0650
      FACT(N) = FACTOR(N,M)                                             ST3C0660
      CP(I) = CP(I) + CP(I,N) * FACT(N)                                 ST3C0670
   7C CONTINUE                                                          ST3C0680
   8C CONTINUE                                                          ST3C0690
      NCI = NCASES + 1                                                  ST3C0700
      IF(NCI .GT. 1C) GO TO 1CC                                         ST3C0710
      DO 9C  N = NCI,1C                                                 ST3C0720
      FACT(N) = C.CC                                                    ST3C0730
   9C CONTINUE                                                          ST3C0740
  1CC J1 = I+1(K)                                                       ST3C0750
      J2 = I+1(K) + NWK - 1                                             ST3C0760
      NEXT = 3 + (2+NOALFA) * (NJ(K)/1C+1)                              ST3C0770
      IF(NW(K) .GT. 1C) NEXT = NEXT + (2+NOALFA) * (NJ(K)/1C+1) + 1     ST3C0780
      ILINES = ILINES + NEXT                                            ST3C0790
      IF(ILINES .LT. 6A) GO TO 12C                                      ST3C0800
      WRITE(6, 11C )                                                    ST3C0810
  11C FORMAT(1H1)                                                       ST3C0820
      ILINES = 1                                                        ST3C0830
  12C WRITE(6, 13C ) K,YV(K),CHORD(K)                                   ST3C0840
  13C FORMAT(1HC,35X,7HSECTION,I3,5X,3HY =,F1C.6,5X,7HCHORD =,F1C.6/    ST3C0850
     1          3X,44H(AC)                                              ST3C0860
      WRITE(6, 14C ) (XP(J),J=J1,J2)                                    ST3C0870
  14C FORMAT(1H ,27X,2HXP,1CF12.4/(31X,1CF12.6))                        ST3C0880
      WRITE(6, 15C ) (CP(J),J=J1,J2)                                    ST3C0890
  15C FORMAT(1H ,2X, 7HCP(A=1),1CF12.4,3(/1CX,1CF12.6))                 ST3C0900
      IF(NOALFA .GT. C) WRITE(6, 16C ) (CPA(J),J=J1,J2)                 ST3C0910
  16C FORMAT(1H ,2X, 7HCP(A=C),1CF12.4,3(/1CX,1CF12.6))                 ST3C0920
C                                                                       ST3C0930
C  ON THE JET                                                           ST3C0940
      NJK = NJ(K)                                                       ST3C0950
      IF(CMJ(K) .LT. C.CCCC1) GO TO 22C                                 ST3C0960
      DO 19C L = 1,NJK                                                  ST3C0970
      II = II + 1                                                       ST3C0980
  17C CP(II) = C.CC                                                     ST3C0990
      DO 18C N = 1,NCASES                                               ST3C1000
      CP(II) = CP(II) + CP(II,N) * FACT(N)                              ST3C1010
  18C CONTINUE                                                          ST3C1020
  19C CONTINUE                                                          ST3C1030
      J1 = I+1(K)                                                       ST3C1040
      J2 = I+1(K) + NJK - 1                                             ST3C1050
      WRITE(6, 21C )                                                    ST3C1060
  21C FORMAT(1H ,4H JET)                                                ST3C1070
      WRITE(6, 14C ) (XP(J),J=J1,J2)                                    ST3C1080
      WRITE(6, 15C ) (CP(J),J=J1,J2)                                    ST3C1090
      IF(NOALFA .GT. C) WRITE(6, 16C ) (CPA(J),J=J1,J2)                 ST3C1100
  22C CONTINUE                                                          ST3C1110
C                                                                       ST3C1120
C  COMPUTE AND PRINT SPANWISE AND TOTAL LOADINGS FOR EACH COMPOSITE CASEST3C1130
C                                                                       ST3C1140
C  DEFINE THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = 1                  ST3C1150
      IF(NOALFA .EQ. C) GO TO 24C                                       ST3C1160
      DO 23C K = 1,NROWS                                                ST3C1170
      TWIST(K) = TST(K,1)                                               ST3C1180
      TH(K) = THETA(K,1)                                                ST3C1190
      THETS(K) = THS(K,1)                                               ST3C1200
  24C CONTINUE                                                          ST3C1210
      DO 26C I = 1,NEWMAX                                               ST3C1220
      RTA(I) = RETA(I,1)                                                ST3C1230
      EP(I) = EPS(I,1)                                                  ST3C1240
  26C CONTINUE                                                          ST3C1250
C                                                                       ST3C1260
C  COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = 1                         ST3C1270
      ALPHA = 1.C                                                       ST3C1280
  27C CALL SLOADI(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,              ST3C1290
     1          XP,DEL,RTA,EP,CPA,CL,CLG,CLMU,CM,CMC,CMMU,CMT,XRCP,XRCL,ST3C1300
     2          CDI,CMMU,CDG,CS,CTT,NROWS,IHINGE)                       ST3C1310
C                                                                       ST3C1320
C  COMPUTE SECTIONAL VORTICITY FOR ALPHA = 1                            ST3C1330
      CALL SLOADG(CPA,DEL,RTA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,NROWS,IHINGE)  ST3C1340
C                                                                       ST3C1350
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = 1                 ST3C1360
      CALL TREFTZ(Y,DELTA,CMU,CGAM,ALEINF,NROWS,ISYMM)                  ST3C1370
C                                                                       ST3C1380
C  MODULATE AND SUM THE FUNDAMENTAL CASE VARIABLES FOR ALPHA = D        ST3C1390
  28C DO 31C K = 1,NROWS                                                ST3C1400
      TWIST(K) = C.CC                                                   ST3C1410
      TH(K) = C.CC                                                      ST3C1420
      THETS(K) = C.CC                                                   ST3C1430
      HC(K) = C.CC                                                      ST3C1440
      DO 3CC N = 1,NCASES                                               ST3C1450
  29C TWIST(K) = TWIST(K) + TST(K,N) * FACT(N)                          ST3C1460
      TH(K) = TH(K) + THETA(K,N) * FACT(N)                              ST3C1470
      THETS(K) = THETS(K) + THS(K,N) * FACT(N)                          ST3C1480
      HC(K) = HC(K) + HL(K,N) * FACT(N)                                 ST3C1490
  3CC CONTINUE                                                          ST3C1500
  31C CONTINUE                                                          ST3C1510
      DO 35C I = 1,NEWMAX                                               ST3C1520
  32C RTA(I) = C.CC                                                     ST3C1530
      EP(I) = C.CC                                                      ST3C1540
      DO 34C N = 1,NCASES                                               ST3C1550
  33C RTA(I) = RTA(I) + RETA(I,N) * FACT(N)                             ST3C1560
      EP(I) = EP(I) + EPS(I,N) * FACT(N)                                ST3C1570
  34C CONTINUE                                                          ST3C1580
  35C CONTINUE                                                          ST3C1590
C                                                                       ST3C1600
C  COMPUTE SECTIONAL COEFFICIENTS FOR ALPHA = D                         ST3C1610
      ALPHA = C.CC                                                      ST3C1620
      IF(NOALFA .GT. C) ALPHA = FACT(1)                                 ST3C1630
  36C CALL SLOADI(ALPHA,IJ,NW,NJ,CHORD,CMU,TH,THETS,TWIST,              ST3C1640
     1          XP,DEL,RTA,EP,CPC,CL,CLGC,CLMUC,CMC,CMGC,CMMUC,CMTC,    ST3C1650
     2          XRCP,XRCL,CDIC,CDMUC,CDGC,CSC,CTC,NROWS,IHINGE)         ST3C1660
C                                                                       ST3C1670
C  COMPUTE SECTIONAL VORTICITY FOR ALPHA = C                            ST3C1680
  37C CALL SLOADG(CPC,DEL,RTA,CHORD,D,CMU,NJ,IJ,CLGC,CGAMC,NROWS,IHINGE)ST3C1690
C                                                                       ST3C1700
C  COMPUTE SECTIONAL DOWNWASH AT INFINITY FOR ALPHA = C                 ST3C1710
  38C CALL TREFTZ(Y,DELTA,CMU,CGAMC,ALEINO,NROWS,ISYMM)                 ST3C1720
C                                                                       ST3C1730
C  COMPUTE SECTIONAL NONLINEAR CROSS-PRODUCT COEFFICIENTS               ST3C1740
      IF(NOALFA .EQ. C) GO TO 4CC                                       ST3C1750
  39C CALL SLOADX(CPA,CPC,DEL,EP,CMU,TH,NW,NJ,IJ,                       ST3C1760
     1          CLGC,CDGX,CDMUX,CSX,CDIX,NROWS)                         ST3C1770
C                                                                       ST3C1780
C  COMPUTE TOTAL LINEAR COEFFICIENTS FOR ALPHA = C                      ST3C1790
  4CC CALL TLOADC(CPRR,CGLG,CCLJ,CCMG,CCMJ,CCMT,CMGMC,CMJMC,CMTMC,      ST3C1800
     1          CLLG,CLLJ,FACT,CCLGC,CCLJC,CCLLC,CCMGC,CCMJC,CCMTC,CCMC,ST3C1810
     2          CMGMCC,CMJMCC,CMTMCC,CMMCC,CXCPC,CXCLC,CXCPRC,CXCLRC,   ST3C1820
     3          CLLGC,CLLJC,CLLC,NCASES,ISYMM,                          ST3C1830
     4          CMGR,CMGL,CMJR,CMJL,CMR,CML,CMRNC,CMRML,CL2RC,CL2LC,    ST3C1840
     5          CMGR2,CMGL2,CMJR2,CMJL2,CMR2,CML2,CMRM2,CMRML2,CL2R2C,CL2L2C)ST3C1850
C                                                                       ST3C1860
C  COMPUTE TOTAL NONLINEAR COEFFICIENTS FOR ALPHA = C                   ST3C1870
  41C CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGC,CDMUC,CSC,CDIC,CLC,DUMB,   ST3C1880
     1          CDGGC,CCDJC,CCSC,CCDIC,CDITTC,DUMB,ALEINO,CCTC,CCJC(1), ST3C1890
     2          DUMMY,CNIC,CCYC,XLEAD,TANLE,XMC,NROWS,ISYMM)            ST3C1900
C                                                                       ST3C1910
C  COMPUTE TOTAL NONLINEAR CROSS-PRODUCT COEFFICIENTS                   ST3C1920
      IF(NOALFA .EQ. C) GO TO 43C                                       ST3C1930
  42C CALL TLOADX(AREA,CHORD,DELTA,Y,CMU,CDGX,CDMUX,CSX,CDIX,CL,CL2,    ST3C1940
     1          CDGGX,CCDJX,CCSX,CCDIX,CDITTX,ALEINF,ALEINO,DUMMY,CCJC(1),ST3C1950
     2          DUMMY,CNIX,CCYX,XLEAD,TANLE,XMC,NROWS,ISYMM)            ST3C1960
C                                                                       ST3C1970
C  PRINT THE SECTIONAL AND TOTAL COEFFICIENTS                           ST3C1980
C                                                                       ST3C1990
  43C WRITE(6, 44C ) M                                                  ST3C2000
  44C FORMAT(1H1,47X,51(4H****)/48X,18H*  COMPOSITE  CASE,I3,3H  */     ST3C2010
     1          49X,51(4H****))                                         ST3C2020
      WRITE(6, 45C ) (N,N=1,1C),(FACT(N),N=1,1C)                        ST3C2030
  45C FORMAT(1H ,49X,24HFUNDAMENTAL CASE FACTORS/1CX,9(4X,2HA(,I1,1H),  ST3C2040
     1          2X),3X,2HA(,I2,1H)/1CX,1CF1C.6/)                        ST3C2050
      WRITE(6, 46C )                                                    ST3C2060
  46C FORMAT(1H ,2CX,29H,**********  LIFT  **********11H   *  *  *     ST3C2070
     1          39H**********  PITCHING MOMENT  **********1CH   *  *  *  ST3C2080
     2          2CH  ***  LIFT CENTER  ***/ 2X,7HSECTION,4X,1HY,8X,4HCLGC,4X,ST3C2090
     3          5HCLMUC,3X,3HCLC,4X,1CH   *  *  *  ,3X,4HCNGC,5X,5HCMMUC,3X,ST3C2100
     4          4HCMTC,4X,4HCMC,4X,1CH   *  *  *  ,3X,4HXCPC/C,4X,4HXCLC)/C) ST3C2110
      IF(NOALFA .GT. C) WRITE(6, 47C )                                  ST3C2120
  47C FORMAT(1H ,                                                       ST3C2130
     1          22X,4HCLGA,4X,5HCLMUA,3X,3HCLA,4X,1CH   *  *  *  ,3X,4HCNGA,ST3C2140
     2          6X,5HCMMUA,3X,4HCMTA,4X,4HCMA,4X,1CH   *  *  *  ,3X,4HXCPA/C, ST3C2150
```

```
     1      4X,4HXCLA/6)                                                    ST3C2150
      DO 500 K = 1,NPOWS                                                   ST3C2160
      WRITE(6, 48 ) K,YEK),CLGC(K),CLMUC(K),CLC(K),CMGC(K),CMMUC(K),       ST3C2170
     1      CMTC(K),CMC(K),XBCPA(K),XRCL(K)                               ST3C2180
 480  FORMAT(1H ,15,4X,4E10.4,1CH   "  "  ,4E10.4,1CH   "  "  ,2E10.6)   ST3C2190
      IF(NOALFA .GT. 0) WRITE(6, 490 ) CLGC(K),CLMU(K),CLC(K),             ST3C2200
     1      CMGC(K),CMMUC(K),CMTC(K),CMC(K),XBCP(K),XRCL(K)               ST3C2210
 490  FORMAT(1H ,19X,3E1C.6,1CH   "  "  ,4E10.6,1CH   "  "  ,2E10.6)      ST3C2220
 500  CONTINUE                                                             ST3C2230
      WRITE(6, 51C ) CCLGC,CCLJC,CCLC,CCMGC,CCMJC,CCMTC,CCMC,CXCPC,CXCLC   ST3C2240
 510  FORMAT(1H , 9X,4(1PH ----------),1CH   "  "  ,4(1CH ----------),     ST3C2250
     1      1CH   "  "  ,2(12H ----------),/13X,6HTOTAL,2X,3E10.6,          ST3C2260
     2      1CH   "  "  ,4E10.6,1CH   (ABFX)  ,2E10.6)                     ST3C2270
      IF(NOALFA .GT. 0) WRITE(6, 520 ) CCLGC(1),CCLJC(1),CCLC(1),          ST3C2280
     1      CCMGC(1),CCMJC(1),CCMTC(1),CCMC(1),CXCPC(1),CXCLC(1)          ST3C2290
 520  FORMAT(1H ,19X,3E1C.6,                                               ST3C2300
     1      1CH   "  "  ,4E10.6,1CH   (ABFX)  ,2E10.6)                     ST3C2310
      WRITE(6, 53C ) CMGMCC,CMJMCC,CMTMCC,CMMCC,CXCPRC,CXCLRC              ST3C2320
 530  FORMAT(1H ,49X,                                                      ST3C2330
     1      1CH   "  "  ,4E10.6,1CH   (XMC)   ,2E10.6)                     ST3C2340
      IF(NOALFA .GT. 0) WRITE(6, 540 )                                     ST3C2350
     1      CMGMCC(1),CMJMCC(1),CMTMCC(1),CMMCC(1),CXCPR(1),CXCLR(1)      ST3C2360
 540  FORMAT(1H ,49X,13H   "  "  ,4E10.6,1CH   (XMC)   ,2E10.6)           ST3C2370
      WRITE(6, 55C )                                                       ST3C2380
 550  FORMAT(1H1,25X,9(4H----),15H INDUCED DRAG ,,9(4H----)/               ST3C2390
     1      7X,7HSECTION,6X,1HY,7X,6HCDGC,7X,6HCDMUC,6X,5HCSC,8X,5HCDC,   ST3C2400
     2      8X,6HGAMMAC,3X,5HALFINC,5X,5HCTCC,8X,5HCMUC)                  ST3C2410
      IF(NOALFA .GT. 0) WRITE(6, 560 )                                     ST3C2420
 560  FORMAT(1H ,27X,6HCDGX,7X,5HCDMUX,6X,5HCSX,8X,5HCDX/                 ST3C2430
     1      29X,6HCDGA2,6X,6HCDMUA2,5X,6HCSA2,7X,6HCDA2,7X,6HGAMMA2,      ST3C2440
     2      5X,6HALFINA)                                                  ST3C2450
      DO 59C K = 1,NPOWS                                                   ST3C2460
      WRITE(6, 57C ) K,YEK),CDGC(K),CDMUC(K),CSC(K),CDIC(K),               ST3C2470
     1      CGAMC(K),ALFINC(K),CTC(K),CMU(K)                              ST3C2480
 570  FORMAT(1H , 6X,I4 ,6X,F10.4,8E11.7)                                  ST3C2490
      IF(NOALFA .GT. 0) WRITE(6, 590 )                                     ST3C2500
     1      CDGX(K),CDMUX(K),CSX(K),CDIX(K),CDG(K),CDMU(K),CS(K),         ST3C2510
     2      CDI(K),CGAM(K),ALFINF(K)                                      ST3C2520
 580  FORMAT(1H ,24X,4E11.7/25X,4E11.7)                                    ST3C2530
 590  CONTINUE                                                             ST3C2540
      WRITE(6, 600 ) CDDGC,CDDJC,CDSC,CDDIC,CDITZC,CCYC,CCJU(1)           ST3C2550
 600  FORMAT(1H ,26X,8(1PH ----------)/1HX,5HTOTAL,2X,4E11.7,11X,3E11.7)/  ST3C2560
     1      IF(NOALFA .GT. 0) WRITE(6, 610 ) CDGX,                        ST3C2570
     1      CCDJX,CCSX,CCDIX,CDITZX,CCDG(1),CCDJ(1),CCS(1),CCDI(1),CDITZ(1)ST3C2580
 610  FORMAT(1H ,24X,4E11.7,11X,E11.7/25X,4E11.7,11X,E11.7)                ST3C2590
C                                                                          ST3C2600
C  PRINT A TABLE OF ALL TOTAL COEFFICIENTS FOR ALPHA = 0,ALPHA,ALPHA**2    ST3C2610
C                                                                          ST3C2620
      WRITE(6, 62C )                                                       ST3C2630
 62C  FORMAT(1H1,41X,9(6H******)/42X,2CH*  TOTAL AERODYNAMIC,              ST3C2640
     1      42X, 7HALPHA=0 *//  8X, 5HALPHA,1CX, 8HALPHA**2)              ST3C2650
      IF(NOALFA .GT. 0) WRITE(6, 630 ) ALFA                                ST3C2660
 630  FORMAT(1H ,25X,A8,F15.7)                                             ST3C2680
      WRITE(6, 63C ) COEFFE(1),CCLGC,COEFFE(2),CCLJC,COEFFE(3),CCL),       ST3C2690
     1      COEFFE(4),CCDGC,COEFFE(5),CCDJC,COEFFE(6),CCSC,COEFFE(7),CCDIC,ST3C2700
     2      COEFFE(8),CDITZC,COEFFE(9),CCJU(1),COEFFE(10),CCMGC,           ST3C2710
     3      COEFFE(11),CCMJC,COEFFE(12),CCMTC,COEFFE(13),CCMC,             ST3C2720
     4      COEFFE(14),CXCPC,COEFFE(15),CXCLC,COEFFE(16),CXCPR,            ST3C2730
     5      COEFFE(17),CXCLR,COEFFE(18),CMGMCC,COEFFE(19),CMJMCC,          ST3C2740
     6      COEFFE(20),CMTMCC,COEFFE(21),CMMCC,COEFFE(22),CLLGC,           ST3C2750
     7      COEFFE(23),CLLJC,COEFFE(24),CLLC,COEFFE(25),CNJ(1),            ST3C2760
     8      COEFFE(26),CNIC,COEFFE(27),CCYC,                               ST3C2770
     9      COEFFE(28),CRGAP,COEFFE(29),CAGLC,COEFFE(30),CRJRC,            ST3C2780
     1      COEFFE(31),CRJLC,COEFFE(32),CRRC,COEFFE(33),CRLC,              ST3C2790
     1      COEFFE(34),CPRMRC,COEFFE(35),CPRMLC                            ST3C2800
      RETURN                                                               ST3C2810
C                                                                          ST3C2820
 640  FORMAT(1H ,25X,AR,3E15.6)                                            ST3C2830
 650  FORMAT(1H ,25X,A8,1X,3E15.7)                                         ST3C2840
 660  WRITE(6, 640 )COEFFE( 1), CCLGC,CCLG(1)                              ST3C2850
      WRITE(6, 640 )COEFFE( 2), CCLJC,CCLJ(1)                              ST3C2860
      WRITE(6, 640 )COEFFE( 3), CCLC ,CCL(1)                               ST3C2870
      WRITE(6, 650 )COEFFE( 4), CCDGC,CCDGX,CCDG(1)                        ST3C2880
      WRITE(6, 650 )COEFFE( 5), CCDJC,CCDJX,CCDJ(1)                        ST3C2890
      WRITE(6, 650 )COEFFE( 6), CCSC ,CCSX ,CCS(1)                         ST3C2900
      WRITE(6, 650 )COEFFE( 7), CCDIC,CCDIX,CCDI(1)                        ST3C2910
      WRITE(6, 650 )COEFFE( 8), CDITZC,CDITZX,CDITZ(1)                     ST3C2920
      WRITE(6, 650 )COEFFE( 9), CCJU(1)                                    ST3C2930
      WRITE(6, 640 )COEFFE(10), CCMGC,CCMG(1)                              ST3C2940
      WRITE(6, 640 )COEFFE(11), CCMJC,CCMJ(1)                              ST3C2950
      WRITE(6, 640 )COEFFE(12), CCMTC,CCMT(1)                              ST3C2960
      WRITE(6, 640 )COEFFE(13), CCMC ,CCM(1)                               ST3C2970
      WRITE(6, 640 )COEFFE(14), CXCPC,CXCP(1)                              ST3C2980
      WRITE(6, 640 )COEFFE(15), CXCLC,CXCL(1)                              ST3C2990
      WRITE(6, 640 )COEFFE(16), CXCPR,CXCPR(1)                             ST3C3000
      WRITE(6, 640 )COEFFE(17), CXCLR,CXCLR(1)                             ST3C3010
      WRITE(6, 640 )COEFFE(18), CMGMCC,CMGMC(1)                            ST3C3020
      WRITE(6, 640 )COEFFE(19), CMJMCC,CMJMC(1)                            ST3C3030
      WRITE(6, 640 )COEFFE(20), CMTMCC,CMTMC(1)                            ST3C3040
      WRITE(6, 640 )COEFFE(21), CMMCC,CMMC(1)                              ST3C3050
      WRITE(6, 640 )COEFFE(22), CLLGC,CLLG(1)                              ST3C3060
      WRITE(6, 640 )COEFFE(23), CLLJC,CLLJ(1)                              ST3C3070
      WRITE(6, 640 )COEFFE(24), CLLC ,CLL(1)                               ST3C3080
      WRITE(6, 640 )COEFFE(25), CNJ(1)                                     ST3C3090
      WRITE(6, 640 )COEFFE(26), CNIC,CNIX,CNI(1)                           ST3C3100
      WRITE(6, 640 )COEFFE(27), CCYC,CCYX,CCY(1)                           ST3C3110
      WRITE(6, 640 )COEFFE(28), CRGAP,  CRGP(1)                            ST3C3120
      WRITE(6, 640 )COEFFE(29), CAGLC,  CAGL(1)                            ST3C3130
      WRITE(6, 640 )COEFFE(30), CRJRC,  CRJR(1)                            ST3C3140
      WRITE(6, 640 )COEFFE(31), CRJLC,  CRJL(1)                            ST3C3150
      WRITE(6, 640 )COEFFE(32), CRRC,   CRR(1)                             ST3C3160
      WRITE(6, 640 )COEFFE(33), CRLC,   CRL(1)                             ST3C3170
      WRITE(6, 640 )COEFFE(34), CPRMRC, CPRMR(1)                           ST3C3180
      WRITE(6, 640 )COEFFE(35), CPRMLC, CPRML(1)                           ST3C3190
C                                                                          ST3C3200
C  COMPUTE AND PRINT A TABLE OF THE VARIATION OF THE TOTAL COEFFICIENTS     ST3C3210
C  WITH ANGLE OF ATTACK                                                    ST3C3220
 670  CALL TABLE(CCLG,CCL(1),CMMCC,CMMC(1),CLLG,CLL(1),CDITZC,CDITZX,      ST3C3230
     1      CDITZ(1),CCJ(1),CNIC,CNIX,CNI(1),CNJ(1),                      ST3C3240
     2      CCYC,CCYX,CCY(1),M)                                           ST3C3250
C                                                                          ST3C3260
      RETURN                                                               ST3C3270
      END                                                                  ST3C3280
```

```
*DECK DATA
      BLOCK DATA                                                           DATA0010
      COMMON/LOAD6/COEFFE(35)                                              DATA0020
      DATA COEFFE/8H    CCLG,8H    CCLJ,8H**   CCL,8H**   CCDG,8H    CCDJ,DATA0030
     1           8H    CCS,8H    CCDI,8H** CDITZ,8H**   CCJ,8H    CCMG,DATA0040
     2           8H    CCMJ,8H    CCMT,8H    CCM,8H    CXCP,8H    CXCL,DATA0050
     3           8H  CXCPR,8H  CXCLR,8H  CCMGMC,8H  CCMJMC,8H  CCMTMC,DATA0060
     4           8H** CCMMC,8H   CLLG,8H    CLLJ,8H**   CLL,8H    CNJ,DATA0070
     5           8H*  CNIMC,8H*   CCY,                                   DATA0080
     6           8H   CAGR,8H   CAGL,8H   CRJR,8H   CRJL,                DATA0090
     7           8H    CPR,8H    CPL,8H  CPMRR,8H  CPMRL/                DATA0100
      END                                                                  DATA0110


*DECK EXPLE
      SUBROUTINE EXPLE(LCASE,CPI,CPI1,DEL,XBB,CPEXP)                       EXLE0010
C                                                                          EXLE0020
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF A LEADING EDGE FVD             EXLE0030
C  AT 5 INTERMEDIATE POINTS ON THE ELEMENT                                 EXLE0040
C                                                                          EXLE0050
      DIMENSION XBB(5),CPEXP(5,10)                                         EXLE0060
C                                                                          EXLE0070
   10 DN = 0.20                                                            EXLE0080
      DO 40 N = 1,5                                                        EXLE0090
   20 X = DN * N                                                           EXLE0100
      XBB(N) = X + DEL                                                     EXLE0110
   30 CPEXP(N,LCASE) = 0.66666*CPI*(1.0/SQRT(X)-X) + CPI1*X                EXLE0120
   40 CONTINUE                                                             EXLE0130
      RETURN                                                               EXLE0140
      END                                                                  EXLE0150


*DECK EXPH1
      SUBROUTINE EXPH1(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)              EXH10010
C                                                                          EXH10020
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE FORWARD HALF OF A HINGE    EXH10030
C  FVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT                             EXH10040
C                                                                          EXH10050
      DIMENSION XBB(5),CPEXP(5,10)                                         EXH10060
C                                                                          EXH10070
      DN = 0.20                                                            EXH10080
      DO 30 N = 1,5                                                        EXH10090
   10 X = XB - DEL + (N-1)*DN*DEL                                          EXH10100
      DX = X - XB                                                          EXH10110
      XBB(N) = X                                                           EXH10120
   20 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(-C*DX)                EXH10130
     1      + ALOG(C*DEL)*DX/DEL) +(CPI + (CPI-CPI1)*DX/DEL)              EXH10140
   30 CONTINUE                                                             EXH10150
      RETURN                                                               EXH10160
      END                                                                  EXH10170


*DECK EXPH2
      SUBROUTINE EXPH2(LCASE,CPI,CPI1,DEL,BTA,C,XB,XBB,CPEXP)              EXH20010
C                                                                          EXH20020
C  THIS SUBROUTINE COMPUTES THE CP VALUE OF THE REAR HALF OF A HINGE       EXH20030
C  FVD AT 5 INTERMEDIATE POINTS ON THE ELEMENT                             EXH20040
C                                                                          EXH20050
      DIMENSION XBB(5),CPEXP(5,10)                                         EXH20060
C                                                                          EXH20070
      DN = 0.20                                                            EXH20080
      DO 30 N = 1,5                                                        EXH20090
   10 DX = N*DN*DEL                                                        EXH20100
      X = XB + DX                                                          EXH20110
      XBB(N) = X                                                           EXH20120
   20 CPEXP(N,LCASE) = -1.273240*BTA/57.295779*(ALOG(C*DX)                 EXH20130
     1      - ALOG(C*DEL)*DX/DEL) + (CPI - (CPI-CPI1)*DX/DEL)             EXH20140
   30 CONTINUE                                                             EXH20150
      RETURN                                                               EXH20160
      END                                                                  EXH20170
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK SLOAD
      SUBROUTINE SLOAD(ALPHA,IJ,NW,NJ,CHORD,CMU,THETA,THETAS,TST,      SLOD0010
     1    XR,DEL,BETA,EPS,CP,CL,CLG,CLMU,CM,CMG,CMMU,CMT,XRCP,XRCL,    SLOD0020
     2    CDI,CDMU,CDG,CS,CT,NROWS,IHINGE)                             SLOD0030
C                                                                      SLOD0040
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF LIFT, PITCHING   SLOD0050
C  MOMENT, AND INDUCED DRAG FOR EITHER A FUNDAMENTAL OR A COMPOSITE CASESLOD0060
C                                                                      SLOD0070
      DIMENSION IJ(40),NW(40),NJ(40)                                   SLOD0080
      DIMENSION CHORD(40),CMU(40),THETA(40),THETAS(40),TST(40)         SLOD0090
      DIMENSION CP(600),XB(600),DEL(620),BETA(600),EPS(600)            SLOD0100
      DIMENSION CLG(40),CLMU(40),CL(40),CMG(40),CMMU(40),CMT(40),CM(40),SLOD0110
     1    XRCP(40),XRCL(40),CDG(40),CDMU(40),CS(40),CDI(40),CT(40)     SLOD0120
C                                                                      SLOD0130
C  INTEGRATE THE CHORDWISE PRESSURES FOR EACH SPANWISE SECTION         SLOD0140
   10 I = 0                                                            SLOD0150
      IF(IHINGE .GT. IJ) IHINGE = 1                                    SLOD0160
      DO 150 K = 1,NROWS                                               SLOD0170
C                                                                      SLOD0180
C  LEADING EDGE CONTRIBUTIONNS                                         SLOD0190
      I = I + 1                                                        SLOD0200
   20 CLI = DEL(I) * (CP(I)+0.50*CP(I+1))                              SLOD0210
      CLG(K) = CLI                                                     SLOD0220
      CMG(K) = -DEL(I)**2 * (0.66667*CP(I)+CP(I+1)) / 3.0              SLOD0230
      CDG(K) = CLI * EPS(I)/57.295779                                  SLOD0240
   30 CS(K) = 0.1745329 * DEL(I) * CP(I)**2                            SLOD0250
C                                                                      SLOD0260
      BCF = 0.00                                                       SLOD0270
      NWK = NW(K)                                                      SLOD0280
      DO 120 L = 2,NWK                                                 SLOD0290
      I = I + 1                                                        SLOD0300
      CPI = CP(I)                                                      SLOD0310
      IF(L .EQ. NWK) GO TO 40                                          SLOD0320
      CPI1 = CP(I+1)                                                   SLOD0330
      GO TO 50                                                         SLOD0340
C  DEFINE TRAILING EDGE CP VALUE                                       SLOD0350
   40 CPI1 = 0.0                                                       SLOD0360
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 50            SLOD0370
      IJK = IJ(K)                                                      SLOD0380
      CPI1 = CP(IJK)                                                   SLOD0390
C                                                                      SLOD0400
C  REGULAR EVD CONTRIBUTIONS                                           SLOD0410
   50 CLI = 0.50 * DEL(I) * (CPI+CPI1)                                 SLOD0420
      CLG(K) = CLG(K) + CLI                                            SLOD0430
      CMG(K) = CMG(K) - CLI*XB(I) - (CPI+2.0*CPI1)*DEL(I)**2/6.00      SLOD0440
   60 CDG(K) = CDG(K) + CLI * EPS(I)/57.295779                         SLOD0450
      BCF = BCF + BETA(I) * (1.0-XB(I))                                SLOD0460
C                                                                      SLOD0470
C  HINGE CONTRIBUTIONS                                                 SLOD0480
      IF(IHINGE .EQ. 0) GO TO 100                                      SLOD0490
      B2 = BETA(I+1)                                                   SLOD0500
      IF(L .LT. NWK) GO TO 70                                          SLOD0510
      B2 = 0.00                                                        SLOD0520
      IJK = IJ(K)                                                      SLOD0530
      IF(CMU(K) .GT. 0.0001) B2 = BETA(IJK)                            SLOD0540
   70 IF((ABS(BETA(I)) .LT. 0.00011.AND.(ABS(B2) .LT. 0.00011)) GO TO 100SLOD0550
      CLI = 0.00                                                       SLOD0560
      CMI = 0.00                                                       SLOD0570
      DL = ALOG(DEL(I) * CHORD(K))                                     SLOD0580
      CON = 0.6366198 * DEL(I) / 57.295779                             SLOD0590
      IF(ABS(BETA(I)) .LT. 0.0001) GO TO 80                            SLOD0600
      CLI = CON * BETA(I) * (2.00 - DL)                                SLOD0610
      CMI = BETA(I) * (0.50-DL/3.00)                                   SLOD0620
   80 IF(ABS(B2) .LT. 0.0001) GO TO 90                                 SLOD0630
      CLI = CLI + CON * B2 * (2.00 - DL)                               SLOD0640
      CMI = CMI + 2.00 * B2 * (0.7500 - DL/3.00)                       SLOD0650
   90 CLG(K) = CLG(K) + CLI                                            SLOD0660
      CMG(K) = CMG(K) - CON * DEL(I) * CMI - CLI * XB(I)               SLOD0670
      CDG(K) = CDG(K) + CLI * EPS(I)/57.295779                         SLOD0680
  100 CONTINUE                                                         SLOD0690
C                                                                      SLOD0700
C  COMPUTE THE SECTIONAL COEFFICIENTS                                  SLOD0710
  110 CLMU(K) = CMU(K) * THETA(K)/57.295779                            SLOD0720
      CL(K) = CLG(K) + CLMU(K)                                         SLOD0730
  120 CMMU(K) = -CMU(K) * THETA(K)/57.295779                           SLOD0740
      CMT(K) = CMU(K) * (ALPHA+TST(K)-THETAS(K)+BCF)/57.295779         SLOD0750
      CM(K) = CMG(K) + CMMU(K) + CMT(K)                                SLOD0760
      XRCP(K) = 0.00                                                   SLOD0770
      XRCL(K) = 0.00                                                   SLOD0780
  130 IF(CLG(K) .NE. 0.00) XRCP(K) = -CMG(K) / CLG(K)                  SLOD0790
      IF(CL(K) .NE. 0.00) XRCL(K) = -(CMG(K)+CMMU(K)) / CL(K)          SLOD0800
  140 CDMU(K) = CMU(K) * (THETA(K)/57.295779)**2 / 2.00                SLOD0810
      CDI(K) = CDG(K) + CDMU(K) - CS(K)                                SLOD0820
      CT(K) = CMU(K) - CDI(K)                                          SLOD0830
  150 CONTINUE                                                         SLOD0840
  160 RETURN                                                           SLOD0850
      END                                                              SLOD0860
```

```
*DECK SLOADX
      SUBROUTINE SLOADX(CPA,CPO,DEL,EPS,CMU,TH,NW,NJ,IJ,               SLDX0010
     1    CLGO,CDGX,CDMUX,CSX,CDIX,NROWS)                              SLDX0020
C                                                                      SLDX0030
C  THIS SUBROUTINE CALCULATES THE SECTIONAL CROSS-PRODUCT VALUES       SLDX0042
C  OF THE NONLINEAR DRAG COEFFICIENTS                                  SLDX0050
C                                                                      SLDX0060
      DIMENSION CPA(600),CPO(600),DEL(600),EPS(600)                    SLDX0070
      DIMENSION CMU(40),TH(40),NW(40),NJ(40),IJ(40)                    SLDX0080
      DIMENSION CLGO(40),CDGX(40),CDMUX(40),CSX(40),CDIX(40)           SLDX0090
C                                                                      SLDX0100
      I = 0                                                            SLDX0110
      DO 70  K = 1,NROWS                                               SLDX0120
C  LEADING EDGE EVD CONTRIBUTION                                       SLDX0130
      I = I + 1                                                        SLDX0140
   10 CDGX(K) = DEL(I)*(CPA(I)+0.50*CPA(I+1))*EPS(I)/57.295779         SLDX0150
   20 CSX(K) = 0.3490658 * DEL(I) * (CPO(I) + CPA(I))                  SLDX0160
      NWK = NW(K)                                                      SLDX0170
      DO 50 L = 2,NWK                                                  SLDX0180
      I = I + 1                                                        SLDX0190
   30 CPI1 = CPA(I+1)                                                  SLDX0200
C  DEFINE TRAILING EDGE CP VALUE                                       SLDX0210
      IF(L .LT. NWK) GO TO 40                                          SLDX0220
      CPI1 = 0.0                                                       SLDX0230
      IF((NJ(K) .EQ. 0) .OR. (CMU(K) .LT. 0.0001)) GO TO 40            SLDX0240
      IJK = IJ(K)                                                      SLDX0250
      CPI1 = CPA(IJK)                                                  SLDX0260
C  REGULAR EVD CONTRIBUTION                                            SLDX0270
   40 CDGX(K) = CDGX(K) + 0.50*DEL(I)*(CPA(I)+CPI1)*EPS(I)/57.295779   SLDX0280
   50 CONTINUE                                                         SLDX0290
C                                                                      SLDX0300
C  COMPUTE THE REMAINING SECTIONAL COEFFICIENTS                        SLDX0310
   60 CDGX(K) = CDGX(K) + CLGO(K)/57.295779                            SLDX0320
      CDMUX(K) = CMU(K) * TH(K)/57.295779**2                           SLDX0330
      CDIX(K) = CDGX(K) + CDMUX(K) - CSX(K)                            SLDX0340
   70 CONTINUE                                                         SLDX0350
      RETURN                                                           SLDX0360
      END                                                              SLDX0370
```

```
*DECK SLOADG
      SUBROUTINE SLOADG(CP,DEL,BETA,CHORD,D,CMU,NJ,IJ,CLG,CGAM,        SLDG0010
     1    NROWS,IHINGE)                                                SLDG0020
C                                                                      SLDG0030
C  THIS SUBROUTINE COMPUTES THE SPANWISE VARIATION OF TOTAL VORTICITY  SLDGCC40
C  ON THE WING-JET SYSTEM                                              SLDG0050
C                                                                      SLDG0060
      DIMENSION CP(600),DEL(600),BETA(600)                             SLDG0070
      DIMENSION CHORD(40),D(40),CMU(40),NJ(40),IJ(40)                  SLDG0080
      DIMENSION CLG(40),CGAM(40)                                       SLDG0090
C                                                                      SLDG0100
      DO 60  K = 1,NROWS                                               SLDG0110
C                                                                      SLDG0120
C  COMPUTE THE SECTIONAL JET VORTICITY, INTEGRATED FROM T.E. TO INFINITYSLDG0130
   10 CGAM(K) = 0.00                                                   SLDG0140
      IF(CMU(K) .LT. 0.0001) GO TO 50                                  SLDG0150
      II = IJ(K)                                                       SLDG0160
C  HINGE EVD CONTRIBUTION                                              SLDG0170
      IF(IHINGE .EQ. 0) GO TO 20                                       SLDG0180
      IF(BETA(II) .NE. 0.00) CGAM(K) = 0.6366198 * DEL(II) *           SLDG0190
     1    BETA(II)/57.295779 * (2.0C-ALOG(CHORD(K)*DEL(II)))           SLDG0200
C  REGULAR EVD CONTRIBUTION                                            SLDG0210
   20 NJK1 = NJ(K) - 1                                                 SLDG0220
      II = II - 1                                                      SLDG0230
      DO 40 L = 1,NJK1                                                 SLDG0240
      II = II + 1                                                      SLDG0250
   30 CGAM(K) = CGAM(K) + 0.50 * DEL(II) * (CP(II)+CP(II+1))           SLDG0260
   40 CONTINUE                                                         SLDG0270
C  FAR-JET EVD CONTRIBUTION                                            SLDG0280
      II = II + 1                                                      SLDG0290
      CGAM(K) = CGAM(K) + 0.50 * D(K) / CHORD(K) * CP(II)              SLDG0300
C                                                                      SLDG0310
C  SUM UP THE WING AND JET CONTRIBUTIONS                               SLDG0320
   50 CGAM(K) = 0.50 * CHORD(K) * (CGAM(K)+CLG(K))                     SLDG0330
   60 CONTINUE                                                         SLDG0340
      RETURN                                                           SLDG0350
      END                                                              SLDG0360
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK TLOAD
      SUBROUTINE TLOAD(ALPHA,AREA,CREF,XMC,Y,DELTA,C,HC,XR,XLEAD,BETA,
     1     CLG,CLMU,CMG,CMMU,CMI,CMU,CCLG,CCLJ,CCL,CCMG,CCMJ,CCM,
     2     CMGMC,CMJMC,CMTMC,CMMF,CXCP,CXCL,CXCPR,CXCLR,CCJ,CLLG,CLLJ,
     3     CLL,ITYPE,IM,NW,NROWS,ISYMM,
     4     CRGR,CRGL,CRJR,CRJL,CAR,CAL,CPRMR,CPRML,CL2R,CL2L)
C
C  THIS SUBROUTINE CALCULATES ALL OF THE TOTAL LOADING PARAMETERS
C  FOR A FUNDAMENTAL CASE
C
      DIMENSION Y(40),DELTA(40),C(40),HC(40),XLEAD(40),IM(40),NW(40)
      DIMENSION XR(40),BETA(40),ITYPE(40)
      DIMENSION CLG(40),CLMU(40),CMG(40),CMMU(40),CMT(40),CMU(40)
C
C  INITIALIZE THE TOTAL COEFFICIENTS
   10 CCLG = 0.00
      CCLJ = 0.00
      CCMG = 0.00
      CCMJ = 0.00
      CCMT = 0.00
      CXCP = 0.00
      CXCL = 0.00
      CXCPR = 0.00
      CXCLR = 0.00
      CCJ  = 0.00
      CLLG = 0.00
      CLLJ = 0.00
      CLL  = 0.00
      CRGR = 0.00
      CRGL = 0.00
      CRJR = 0.00
      CRJL = 0.00
      CAR  = 0.00
      CAL  = 0.00
      CL2R = 0.00
      CL2L = 0.00
      CPRMR = 0.00
      CPRML = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 120 K = 1,NROWS
   20 CDEL = C(K) * DELTA(K)
      IF(ISYMM .LT. 0) GO TO 80
C
C  LIFT COEFFICIENTS
      CCLG = CCLG + CDEL * CLG(K)
   30 CCLJ = CCLJ + CDEL * CLMU(K)
C
C  PITCHING MOMENT COEFFICIENTS
      CCDEL = CDEL * C(K)
      XLR = XLEAD(K) / C(K)
C  COMPUTE LEADING EDGE HEIGHT ABOVE WING APEX
   40 L = IM(K) - 1
      NWK = NW(K)
      XDS = 0.00
      DO 60 L = 1,NWK
      I = I + 1
      IF(ITYPE(I) - 4) 50, 50, 70
   50 XDS = XDS + XR(I) * BETA(I)/57.295779
   60 CONTINUE
   70 HLR = HC(K) - XLR * ALPHA/57.295779 - XDS
      CCMG = CCMG + CCDEL * (CMG(K) - CLG(K)*XLR)
      CCMJ = CCMJ + CCDEL * (CMMU(K) - CLMU(K)*XLR)
      CCMT = CCMT + CCDEL * (CMT(K) - CMU(K)*HLR)
   80 CCJ  = CCJ  + CDEL * CMU(K)
C
C  ROLLING MOMENT COEFFICIENTS AND ROOT BENDING MOMENTS
      CDELY = CDEL * Y(K)
   90 IF(ISYMM .GE. 0) GO TO 100
      CLLG = CLLG + CDELY * CLG(K)
      CLLJ = CLLJ + CDELY * CLMU(K)
  100 IF(Y(K) .LT. 0.) GO TO 110
      CRGR = CRGR + CDELY * CLG(K)
      CRJR = CRJR + CDELY * CLMU(K)
      CL2R = CL2R + CDEL * (CLG(K)*CLMU(K))
      GO TO 120
  110 CRGL = CRGL - CDELY * CLG(K)
      CRJL = CRJL - CDELY * CLMU(K)
      CL2L = CL2L + CDEL * (CLG(K)*CLMU(K))
  120 CONTINUE
C
C  COMPUTE THE FINAL VALUES OF ALL THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
  130 CCLG = FACTOR * CCLG
      CCLJ = FACTOR * CCLJ
      CCL  = CCLG + CCLJ
  140 CCJ  = FACTOR * CCJ
  150 FACTOR = FACTOR / CREF
      CCMG = FACTOR * CCMG
      CCMJ = FACTOR * CCMJ
      CCMT = FACTOR * CCMT
      CCM  = CCMG + CCMJ + CCMT
      IF(ISYMM .LT. 1) GO TO 160
      IF(CCLG .NE. 0.00) CXCP = -.CCMG / CCLG
      IF(CCL .NE. 0.00) CXCL = -(CCMG+CCMJ) / CCL
      CXCPR = CXCP * CREF
      CXCLR = CXCL * CREF
  160 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
  170 IF(ISYMM .EQ. 1) GO TO 180
      FACTOR = -1.00 / AREA
      IF(ISYMM .LT. 0) FACTOR = -2.00 / AREA
      CLLG = FACTOR * CLLG
      CLLJ = FACTOR * CLLJ
      CLL  = CLLG + CLLJ
  180 FACTOR = 4.00 / AREA
      CRGR = FACTOR * CRGR
      CRJR = FACTOR * CRJR
      CL2R = FACTOR * CL2R
      IF(ISYMM .GT. 0) GO TO 190
      IF(ISYMM .LT. 0) GO TO 200
      CRGL = CRGR
      CRJL = CRJR
      CL2L = CL2R
      GO TO 210
  190 CRGL = FACTOR * CRGL
      CRJL = FACTOR * CRJL
      CL2L = FACTOR * CL2L
      GO TO 210
  200 CRGL = -CRGR
      CRJL = -CRJR
      CL2L = -CL2R
  210 CAR = CRGR + CRJR
      CAL = CRGL + CRJL
      IF(CL2R .NE. 0.0) CPRMR = CAR/CL2R
      IF(CL2L .NE. 0.0) CPRML = CAL/CL2L
      RETURN
      END
```

```
*DECK TLOADX
      SUBROUTINE TLOADX(AREA,CHORD,DELTA,Y,CMU,CDG,CDMU,CS,CDI,CL,CLG,
     1                  CCDG,CCDJ,CCS,CCDI,CDITZ,ALEINF,ALEINC,CCF,CCJ,
     2                  CNJ,CNI,CCY,XLEAD,TANLE,XMC,NROWS,ISYMM)
C
C  THIS SUBROUTINE CALCULATES THE NONLINEAR TOTAL LOADING COEFFICIENTS
C  FOR ALPHA = 0.0 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),CDIT(40),CL(40),CLG(40)
      DIMENSION ALEINF(40),ALEINC(40)
C
C  INITIALIZE THE COEFFICIENTS
   10 CCDG = 0.00
      CCDJ = 0.00
      CCS = 0.00
      CDITZ = 0.00
      CNJ = 0.00
      CNI = 0.00
      CCY = 0.00
C
C  INTEGRATE THE SECTIONAL VALUES OVER THE SPAN
      DO 40 K = 1,NROWS
   20 CDEL = CHORD(K) * DELTA(K)
      CCDG = CCDG + CDEL * CDG(K)
      CCDJ = CCDJ + CDEL * CDMU(K)
   30 CCS = CCS + CDEL * CS(K)
      CDITZ = CDITZ + CDEL * (CL(K)*ALEINC(K) + CLG(K)*ALEINF(K))
      IF(ISYMM .LT. 1) GO TO 40
      CNJ = CNJ + CDEL * Y(K) * CMU(K)
      CNI = CNI + CDEL * (Y(K)*CDI(K) - CS(K)*TANLE(K)*(XLEAD(K)-XMC))
      CCY = CCY + CDEL * CS(K) * TANLE(K)
   40 CONTINUE
C  COMPUTE THE FINAL VALUES OF THE TOTAL COEFFICIENTS
      FACTOR = 2.00 / AREA
      IF(ISYMM .LT. 1) FACTOR = 4.00 / AREA
   50 CCDG = FACTOR * CCDG
      CCDJ = FACTOR * CCDJ
      CCS = FACTOR * CCS
      CDITZ = FACTOR * CDITZ / 2.00
      CCDI = CCDG + CCDJ - CCS
      CCF = CCJ - CCDI
   60 IF(ISYMM .LT. 1) RETURN
      FACTOR = 1.00 / AREA
      CNJ = -FACTOR * CNJ
      CNI = FACTOR * CNI
      CCY = FACTOR * CCY
   70 RETURN
      END
```

C20

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

C21 THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK TLOADC                                                                    TLD00010
      SUBROUTINE TLOADC(CREF,CCLO,CCLA,CCMO,CCMJ,CCMT,CMOMC,CMJMC,CMTMC,TLD00020
     1           CLLO,CLLJ,CACT,CCLO2,CCLJ2,CCMO2,CCMJ2,CCMT2,CCMC,TLD00030
     2           CMOMC2,CMJMC2,CMTMC2,CCMC2,CXCP2,CXCLP,CXCPAC,CXCLAC,TLD00040
     3           CLLO2,CLLJ2,FLEE,NCASES,ISYMM,                       TLD00050
     4     CRO2,CRL2,CRJ2,CRJL2,CRO,CRL,CRMO,CRML,CL2O,CL2L,          TLD00060
     5     CRO2C,CRL2C,CRJ2C,CRJL2C,CRO2C,CRLO,CRMO2C,CRML2C,CL2O,CL2LO)TLD00070
C                                                                               TLD00080
C  THIS SUBROUTINE CALCULATES THE LINEAR TOTAL LOADING COEFFICIENTS             TLD00090
C  FOR ALPHA = 0.  LINEAR QUANTITIES ARE MODULATED AND SUMMED                   TLD00100
C  ACCORDING TO THE COMPOSITE CASE REQUIREMENTS.                                TLD00110
C                                                                               TLD00120
      DIMENSION CCLO(11),CCLJ(11),CCMO(11),CCMJ(11),CCMT(11),      TLD00130
     1          CMOMC(11),CMJMC(11),CMTMC(11),CLLO(11),CLLJ(11),FACT(11)TLD00140
      DIMENSION CRO2(11),CRL2(11),CRJ2(11),CRJL2(11),CRO(11),CRL(11),TLD00150
     1          CRMO(11),CRML(11),CL2O(11),CL2L(11)                  TLD00160
C                                                                               TLD00170
C  INITIALIZE THE COEFFICIENTS                                                  TLD00180
      IC CCLOC = 0.00                                                           TLD00190
         CCLJC = 0.00                                                           TLD00200
         CCMOC = 0.00                                                           TLD00210
         CCMJC = 0.00                                                           TLD00220
         CCMTC = 0.00                                                           TLD00230
         CMOMC2 = 0.00                                                          TLD00240
         CMJMC2 = 0.00                                                          TLD00250
         CMTMC2 = 0.00                                                          TLD00260
         CCMC2 = 0.00                                                           TLD00270
         CLLO2 = 0.00                                                           TLD00280
         CLLJ2 = 0.00                                                           TLD00290
         CXCP2 = 0.00                                                           TLD00300
         CXCLP = 0.00                                                           TLD00310
         CXCPAC = 0.00                                                          TLD00320
         CXCLAC = 0.00                                                          TLD00330
         CRO2C = 0.00                                                           TLD00340
         CRL2C = 0.00                                                           TLD00350
         CRJ2C = 0.00                                                           TLD00360
         CRJL2C = 0.00                                                          TLD00370
         CRO = 0.00                                                             TLD00380
         CRL = 0.00                                                             TLD00390
         CL2O = 0.00                                                            TLD00400
         CL2L = 0.00                                                            TLD00410
         CRMO = 0.00                                                            TLD00420
         CRML = 0.00                                                            TLD00430
C                                                                               TLD00440
C  MODULATE AND SUM THE TOTAL COEFFICIENTS                                      TLD00450
      DO 40 N = 1,NCASES                                                        TLD00460
   20 CCLOC = CCLOC + CCLO(N) * FACT(N)                                         TLD00470
      CCLJC = CCLJC + CCLJ(N) * FACT(N)                                         TLD00480
      CCMOC = CCMOC + CCMO(N) * FACT(N)                                         TLD00490
      CCMJC = CCMJC + CCMJ(N) * FACT(N)                                         TLD00500
      CCMTC = CCMTC + CCMT(N) * FACT(N)                                         TLD00510
   30 CMOMC2 = CMOMC2 + CMOMC(N) * FACT(N)                                      TLD00520
      CMJMC2 = CMJMC2 + CMJMC(N) * FACT(N)                                      TLD00530
      CMTMC2 = CMTMC2 + CMTMC(N) * FACT(N)                                      TLD00540
      CLLOC = CLLOC + CLLO(N) * FACT(N)                                         TLD00550
      CLLJC = CLLJC + CLLJ(N) * FACT(N)                                         TLD00560
      CRORC = CRORC + CRO(N) * FACT(N)                                          TLD00570
      CRLC = CRLC + CRL(N) * FACT(N)                                            TLD00580
      CRJRC = CRJRC + CRJR(N) * FACT(N)                                         TLD00590
      CRJLC = CRJLC + CRJL(N) * FACT(N)                                         TLD00600
      CL2RC = CL2RC + CL2R(N) * FACT(N)                                         TLD00610
      CL2LC = CL2LC + CL2L(N) * FACT(N)                                         TLD00620
   40 CONTINUE                                                                  TLD00630
C  DEFINE THE REMAINING TOTAL COEFFICIENTS                                      TLD00640
      CCLO = CCLOC + CCLJC                                                      TLD00650
   50 CCMC = CCMOC + CCMJC + CCMTC                                              TLD00660
      CRO = CRORC + CRJRC                                                       TLD00670
      CRL = CRLC + CRJLC                                                        TLD00680
      IF(CL2RC .NE. 0.0) CRMRC = CRRC / CL2RO                                   TLD00690
      IF(CL2LC .NE. 0.0) CRMLO = CRLO / CL2LO                                   TLD00700
      IF(ISYMM .LT. 0) GO TO 60                                                 TLD00710
      IF(CCLOC .NE. 0.0) CXCLO = - (CCMCO+CCMJO) / CCLO                         TLD00720
      CXCPRO = CXCPO * CREF                                                     TLD00730
      CXCLRO = CXCLO * CREF                                                     TLD00740
   60 CMMCC = CMOMCO + CMJMCO + CMTMCO                                          TLD00750
      CLLC = CLLOC + CLLJC                                                      TLD00760
   70 RETURN                                                                    TLD00770
      END
```

```
*DECK TREFTZ                                                                    TRFT0010
      SUBROUTINE TREFTZ(LY,DELTA,CMU,GAMR,ALEINF,NROWS,LIKE)                    TRFT0020
      DIMENSION Y(40),DELTA(40),CMU(40),GAMA(40),ALEINF(40)                     TRFT0030
      DIMENSION E(40,3),C(40),OP(40),DM(40),DGAM(40)                            TRFT0040
      LOGICAL Z1,Z2,Z3,Z4,Z5,Z6,ZT1,ZT2                                         TRFT0050
C                                                                               TRFT0060
C  FIND COEFFICIENTS OF CURVES                                                  TRFT0070
      J=0                                                                       TRFT0080
      DO 60 I=1,NROWS                                                           TRFT0090
      Z1=I.EQ.1                                                                 TRFT0100
      Z2=LIKE.GT.0.AND.I.EQ.NROWS                                               TRFT0110
      Z3=(LIKE.LE.0.AND.I.EQ.NROWS).OR.(LIKE.GT.0.AND.I.NE.NROWS.AND.           TRFT0120
     1Y(I).GT.0.0.AND.Y(I+1).LT.0.0)                                            TRFT0130
      Z4=LIKE.GT.0.AND.I.GT.1.AND.Y(I).LT.0.0.AND.Y(I-1).GT.0.0                 TRFT0140
      Z5=(I.NE.NROWS).AND.((CMU(I).LT.0.0CC1.AND.CMU(I+1).GT.0.001).OR.         TRFT0150
     1(CMU(I).LT.-.0CC1.AND.CMU(I+1).GT.0.0))                                   TRFT0160
      Z6=(I.GT.1).AND.((CMU(I).GT.0.0.AND.CMU(I-1).LT.0.0CC1).OR.               TRFT0170
     1(CMU(I).LT.-.0CC1.AND.CMU(I-1).GT.0.0))                                   TRFT0180
      D1=SQRT(1.0-ABS(Y(I)))                                                    TRFT0190
      ZT1=Z1.OR.Z4.OR.Z6                                                        TRFT0200
      ZT2=Z2.OR.Z3.OR.Z5                                                        TRFT0210
      IF(I.EQ.1)GO TO 10                                                        TRFT0220
      IF(ZT1)J=1                                                                TRFT0230
      IF(ZT1)GO TO 40                                                           TRFT0240
      J=0                                                                       TRFT0250
   10 CONTINUE                                                                  TRFT0260
      IF(ZT2) GO TO 30                                                          TRFT0270
      D2=SQRT(1.0-ABS(Y(I+1)))                                                  TRFT0280
      D3=SQRT(1.0-ABS(Y(I-1)))                                                  TRFT0290
      G1=GAMR(I)                                                                TRFT0300
      G2=GAMR(I+1)                                                              TRFT0310
      G3=GAMR(I-1)                                                              TRFT0320
      D13=D1**3                                                                 TRFT0330
      D23=D2**3                                                                 TRFT0340
      D33=D3**3                                                                 TRFT0350
      D15=D1**5                                                                 TRFT0360
      D25=D2**5                                                                 TRFT0370
      D35=D3**5                                                                 TRFT0380
      E1=D23*D35-D33*D25                                                        TRFT0390
      E2=D13*D35-D33*D15                                                        TRFT0400
      E3=D13*D25-D23*D15                                                        TRFT0410
      E(I)=(G1-F1-G2*E2+G3*E3)/(D1*E1-D2*E2+D3*E3)                              TRFT0420
      E(I)=((G1-E(I)*D1)-D25*(G3-E(I)*D3)/D35)/(D13*D25-D23*D15)                TRFT0430
      C(I)=(G1-E(I)*D1)/D1-M(I)*D13/D15                                         TRFT0440
      IF(J.EQ.0) GO TO 40                                                       TRFT0450
   20 E(I-1)=E(I)                                                               TRFT0460
      B(I-1)=B(I)                                                               TRFT0470
      C(I-1)=C(I)                                                               TRFT0480
      J=0                                                                       TRFT0490
      GO TO 40                                                                  TRFT0500
   30 E(I)=E(I-1)                                                               TRFT0510
      B(I)=B(I-1)                                                               TRFT0520
      C(I)=C(I-1)                                                               TRFT0530
   40 CONTINUE                                                                  TRFT0540
      IF(Y(I).LT.0.0) GO TO 50                                                  TRFT0550
      DP(I)=SQRT(ABS(D1**2-DELTA(I)))                                           TRFT0560
      DM(I)=SQRT(D1**2+DELTA(I))                                                TRFT0570
      GO TO 60                                                                  TRFT0580
   50 DP(I)=SQRT(D1**2+DELTA(I))                                                TRFT0590
      DM(I)=SQRT(D1**2-DELTA(I))                                                TRFT0600
   60 CONTINUE                                                                  TRFT0610
C                                                                               TRFT0620
C  CALCULATE STRENGTH OF DISCRETE VORTICIES                                     TRFT0630
      DO 80 I=1,NROWS                                                           TRFT0640
      IF(I.LT.NROWS.AND.((CMU(I).LT.-.0CC1.AND.CMU(I+1).GT.0.0).OR.             TRFT0650
     1(CMU(I).GT.-.0CC1.AND.CMU(I+1).LT.0.0CC1))) GO TO 70                      TRFT0660
      DGAM(I)=0.0                                                               TRFT0670
      GO TO 80                                                                  TRFT0680
   70 DGAM(I)=E(I+1)-E(I))*DM(I)+(B(I+1)-B(I))*DM(I)**3+(C(I+1)-C(I))           TRFT0690
     1*DM(I)**5                                                                 TRFT0700
   80 CONTINUE                                                                  TRFT0710
C                                                                               TRFT0720
C  CALCULATE DOWNWASH                                                           TRFT0730
      DO 100 I=1,NROWS                                                          TRFT0740
      ALEINF(I)=0.0                                                             TRFT0750
      F2=SQRT(1.0-Y(I))                                                         TRFT0760
      F4=SQRT(1.0+Y(I))                                                         TRFT0770
      DO 90 J=1,NROWS                                                           TRFT0780
      IF(Y(J).LT.0.0) F2=F4                                                     TRFT0790
      CON=1.0                                                                   TRFT0800
      IF(Y(J).LT.-.01) CON=-1.0                                                 TRFT0810
      F1=ALOG(ABS((DP(J)-F2)*(DM(J)+F2))/((DP(J)+F2)*(DM(J)-F2))))              TRFT0820
      S1=CON*F1/F2                                                              TRFT0830
      S2=(2.0*(DP(J)-DM(J))+F2*F1)*CON                                          TRFT0840
      S3=CON*(2.0*(DP(J)**3-DM(J)**3)/3.0)+F2**2*S2                             TRFT0850
      DELG=-2GAM(J)/(Y(J)-Y(I)+DELTA(J))                                        TRFT0860
      IF(LIKE.GT.0) GO TO 85                                                    TRFT0870
      F3=ALOG(ABS((DP(J)-F4)*(DM(J)+F4))/((DP(J)+F4)*(DM(J)-F4))))              TRFT0880
      S2P=2.0*(DP(J)-DM(J))+F4*F3                                               TRFT0890
      IF(LIKE.LT.0) SY=-1.0                                                     TRFT0900
      IF(LIKE.EQ.0) SY=1.0                                                      TRFT0910
      S1=S1+SY*F3/F4                                                            TRFT0920
      S2=S2+SY*S2P                                                              TRFT0930
      S3=S3+SY*((2.0*(DP(J)**3-DM(J)**3)/3.0)+F4**2*S2P)                        TRFT0940
      DELG=DELG-SY*DGAM(J)/(-Y(I)-Y(J)+DELTA(J))                                TRFT0950
   90 ALEINF(I)=ALEINF(I)+C.1591549*(0.5*E(J)*S1+1.5*B(J)*S2+2.5*C(J)*          TRFT0960
     1S3+DELG)                                                                  TRFT0970
  100 CONTINUE                                                                  TRFT0980
      RETURN                                                                    TRFT0990
      END                                                                       TRFT0999
```

```
      SUBROUTINE TABLE(CLO,CLA,CMO,CMA,CLLO,CLLA,CDIO,CDIX,CDIA2,CJ,    TABL0010
     1                 CNIO,CNIX,CNIA2,CNJ,CYO,CYX,CYA2,MCASE)           TABL0020
C                                                                       TABL0030
C  THIS SUBROUTINE COMPUTES AND PRINTS A TABLE OF TOTAL COEFFICIENTS     TABL0040
C  FOR A RANGE OF ANGLES OF ATTACK                                      TABL0050
C                                                                       TABL0060
C  COMPUTE THE COEFFICIENTS                                             TABL0070
      WRITE(6, 10 ) MCASE                                               TABL0080
   10 FORMAT(1H1,31X,14(4H****)      /32X,27H*  TABULATED TOTAL COEFFICI,TABL0090
     1     23HENTS FOR COMPOSITE CASE, I3, 3H  */ 32X,14(4H****)//      TABL0100
     2  4X,5HALPHA,10X,3HCCL,8X,6HCCL**2,5X,7HCCM(MC),4X,3HCLL,13X,     TABL0110
     3  5HCDITZ,6X,3HFCT,13X,3HCNT,8X,2HCN,9X,3HCCY)                     TABL0120
      ALPHA = -11.00                                                    TABL0130
      DO 60  M = 1,41                                                   TABL0140
   20 ALPHA = ALPHA + 1.CC                                              TABL0150
      ALPHA2 = ALPHA * ALPHA                                            TABL0160
C  LINEAR COEFFICIENTS                                                  TABL0170
   30 CL = CLO + CLA * ALPHA                                            TABL0180
      CL2 = CL * CL                                                     TABL0190
      CM = CMO + CMA * ALPHA                                            TABL0200
      CLL = CLLO + CLLA * ALPHA                                         TABL0210
C  NONLINEAR COEFFICIENTS                                               TABL0220
   40 CDI = CDIO + CDIX * ALPHA + CDIA2 * ALPHA2                        TABL0230
      CT = CJ - CDI                                                     TABL0240
      CNI = CNIO + CNIX * ALPHA + CNIA2 * ALPHA2                        TABL0250
      CN = CNJ + CNI                                                    TABL0260
      CY = CYO + CYX * ALPHA + CYA2 * ALPHA2                            TABL0270
```

```
C
C  PRINT THE TABLE                                                      TABLO280
      WRITE(6, 50 ) ALPHA,CL,CL2,CM,CLL,CDI,CT,CNI,CN,CY               TABLO290
   50 FORMAT(1H , F10.4,5H  *  ,4F11.7,5H  *  ,2F11.7,5H  *  ,3F11.7)  TABLO300
   60 CONTINUE                                                         TABLO310
      RETURN                                                           TABLO320
      END                                                              TABLO330
                                                                       TABLO340


*DECK FUNDER
      SUBROUTINE FUNDER(EPS,CPC,CPA,CPRC,CPRA,CPP,DEL,CHORD,Y,DELTA,CMU,FUNDC010
     1 AREA,CLO,CMO,CMOMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX,NCASES,NOALFA,FUNDC020
     2    NROWS,ISYMM,XL,TL,XMC)                                       FUNDC030
C                                                                      FUNDC040
C  THIS SUBROUTINE CONTROLS CALCULATION OF ALL AERODYNAMIC COEFFICIENTS FUNDC050
C  AND STABILITY DERIVATIVES FOR THE FUNDAMENTAL CASES                 FUNDC060
C                                                                      FUNDC070
      COMMON/SOLV1/CP(600,10)                                          FUNDC080
      DIMENSION CPC(NEWMAX),CPA(NEWMAX),CPRC(600),CPRA(600),CPP(600)   FUNDC090
      DIMENSION DEL(600),EPS(600,10),EP(600)                           FUNDC100
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)        FUNDC110
      DIMENSION NW(40),IJ(40)                                          FUNDC120
C                                                                      FUNDC130
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED        FUNDC140
      IREAD = NMAX + NJT                                               FUNDC150
      IRIDGE = 0                                                       FUNDC160
      CLLP = 0.00                                                      FUNDC170
      CNP2 = 0.00                                                      FUNDC180
      CNPC = 0.00                                                      FUNDC190
C  CALCULATE THE CHORDWISE LOADING                                     FUNDC200
      IC CALL STGAFC(NEWMAX)                                           FUNDC210
C                                                                      FUNDC220
C  DEFINE THE DUMMY CP ARRAYS FOR THE ALPHA CASE                       FUNDC230
C  AND THE CP ARRAY FOR THE ROLLING FUNDAMENTAL CASE                   FUNDC240
      DO 20 I = 1,NEWMAX                                               FUNDC250
      CPA(I) = 0.00                                                    FUNDC260
      CPRA(I) = 0.00                                                   FUNDC270
      CPP(I) = CP(I,NCASES)                                            FUNDC280
   20 CONTINUE                                                         FUNDC290
      DO 130 N = 1,NCASES                                              FUNDC300
      LCASE = N                                                        FUNDC310
      IREAD = IREAD + 1                                                FUNDC320
C                                                                      FUNDC330
C  CALCULATE THE SPANWISE AND TOTAL LOADING                            FUNDC340
      IF(N .EQ. NCASES)                                                FUNDC350
     1     CALL STGAFC(DUM1,DUM2,DUM3,CLLP,NEWMAX,NOALFA,LCASE)        FUNDC360
C                                                                      FUNDC370
C  READ THE FIRST RUN SOLUTION                                         FUNDC380
      CALL READMS(1,CP,NEWMAX,IREAD)                                   FUNDC390
C                                                                      FUNDC400
C  DEFINE CP AND EP ARRAYS FOR THE PRESENT SECOND RUN FUNDAMENTAL CASE FUNDC410
      DO 40 I = 1,NEWMAX                                               FUNDC420
      CPR(I) = CP(I,LCASE)                                             FUNDC430
      EP(I) = EPS(I,LCASE)                                             FUNDC440
   40 CONTINUE                                                         FUNDC450
C                                                                      FUNDC460
C  CALCULATE THE STABILITY DERIVATIVES                                 FUNDC470
      CALL SUMIT1(CPC,CPA,CPP,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL,FUNDC480
     1    XMC,AREA,DUM1,DUM2,CNP2,CLLPC,DUM3,DUM5,DUM6,CYP2,           FUNDC490
     2    NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)                             FUNDC500
      IF(N .LT. NCASES) GO TO 50                                       FUNDC510
      CNPC = DUM1                                                      FUNDC520
      GO TO 110                                                        FUNDC530
   50 CALL SUMIT2(CPC,CPA,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,FUNDC540
     1    XMC,CNPC,DUM1,DUM2,CNR2C,DUM3,DUM4,CYPC,DUM5,DUM6,           FUNDC550
     2    CYR2C,DUM7,DUM8,NW,IJ,NROWS,ISYMM,NEWMAX)                    FUNDC560
C                                                                      FUNDC570
C  PRINT THE STABILITY DERIVATIVE DATA                                 FUNDC580
   60 WRITE(6, 70 ) LCASE                                              FUNDC590
   70 FORMAT(1H1,32X,1314H****,  3H*** / 33X,                          FUNDC600
     1       40H*  STABILITY DERIVATIVE DATA FOR FUNDAMENTAL CASE, I3,  FUNDC610
     2       2H  * / 33X,1314H****,  3H*** /)                          FUNDC620
      WRITE(6, 80 ) CLLPC                                              FUNDC630
   80 FORMAT(1HC/ 25X, 44HROLLING MOMENT COEFFICIENT DERIVATIVE DUE TO,  FUNDC640
     1       15H YAWING, CLLP =, F12.7)                                FUNDC650
      WRITE(6, 90 ) CNP2,CNR2C                                         FUNDC660
   90 FORMAT(1HC/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,    FUNDC670
     1       53H YAWING ABOUT XCG, CN(P) MAY BE CALCULATED AS FOLLOWS //FUNDC680
     2       40X, 25HCN(P) = CNR*R + CNR2*PP**2 //                     FUNDC690
     3       40X, 13HWHERE  CNR =,F14.9 / 56X, 6HCNR2 =,F12.7)         FUNDC700
      WRITE(6, 100 ) CYP2,CYR2C                                        FUNDC710
  100 FORMAT(1HC/ 23X,43HSIDE FORCE COEFFICIENT DUE TO YAWING, CY(P),  FUNDC720
     1       20H MAY BE CALCULATED AS FOLLOWS//                        FUNDC730
     2       40X,25HCY(P) = CYR*R + CYR2*PP**2//                       FUNDC740
     3       40X,13HWHERE  CYR =,E14.9/ 56X,6HCYR2 =,F12.7)            FUNDC750
     4       44H ROLLING, CM(P) MAY BE CALCULATED AS FOLLOWS //        FUNDC760
      GO TO 130                                                        FUNDC770
  110 WRITE(6, 70 ) LCASE                                              FUNDC780
      WRITE(6, 120 ) CLLP,CNP2,CYP2                                    FUNDC790
  120 FORMAT(1HC/ 24X, 33HROLLING MOMENT COEFF DERIVATIVE DUE TO,      FUNDC800
     1       14H ROLLING, CLLP =, F12.7 ///                            FUNDC810
     2       17X, 43HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,       FUNDC820
     3       42HROLLING, CN(P) MAY BE CALCULATED AS FOLLOWS //         FUNDC830
     4       53X, 17HCN(P) = CNP2*PP**2 //                             FUNDC840
     5       53X, 13HWHERE  CNP2 =,F12.7///                            FUNDC850
     6       32X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P),      FUNDC860
     7       22H MAY BE CALCULATED AS FOLLOWS//                        FUNDC870
     8       53X,17HCY(P) = CYP2*PP**2//                               FUNDC880
     9       53X,13HWHERE  CYP2 =,F12.7)                               FUNDC890
  130 CONTINUE                                                         FUNDC900
      RETURN                                                           FUNDC910
      END                                                              FUNDC920
```

```
*DECK COMDER
      SUBROUTINE COMDER(EPS,CPC,CPA,CPRC,CPRA,CPP,CPREAD,DEL,CHORD,Y,  COMDC010
     1    CMU,DELTA,AREA,CLO,CMO,CMOMC,CLLP,CNP2,NW,IJ,NMAX,NJT,NEWMAX, COMDC020
     2    NCASES,NROWS,ISYMM,XL,TL,XMC)                                COMDC030
C                                                                      COMDC040
C  THIS SUBROUTINE CONTROLS CALCULATION OF STABILITY DERIVATIVES       COMDC050
C  FOR ALL COMPOSITE CASES                                             COMDC060
C                                                                      COMDC070
      COMMON/SOLV1/CP(600,10)                                          COMDC080
      COMMON/COMBOS/FACTOR(10,24),NCC                                  COMDC090
      DIMENSION CPC(NEWMAX),CPA(NEWMAX),CPRC(600),CPRA(600),CPP(600),  COMDC100
     1    CPREAD(NEWMAX)                                               COMDC110
      DIMENSION DEL(600),EPS(600,10),EP(600)                           COMDC120
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)        COMDC130
      DIMENSION NW(40),IJ(40)                                          COMDC140
C                                                                      COMDC150
C  CYCLE THROUGH ALL COMPOSITE CASES                                   COMDC160
      IFUDGE = 1                                                       COMDC170
      DO 100 M = 1,NCC                                                 COMDC180
      MCASE = M                                                        COMDC190
      NC1 = NCASES - 1                                                 COMDC200
C  FIND THE SPOT WHERE THE SOLUTION OF THE FIRST RUN WAS STORED        COMDC210
      IREAD = NMAX + NJT + 1                                           COMDC220
C                                                                      COMDC230
C  DEFINE CP AND EP ARRAYS FOR THIS SECOND RUN COMPOSITE CASE          COMDC240
C  NOTE THAT CPP HAS PREVIOUSLY BEEN DEFINED IN FUNDER                 COMDC250
      DO 20 I = 1,NEWMAX                                               COMDC260
      CPR(I) = 0.00                                                    COMDC270
      EP(I) = 0.00                                                     COMDC280
      CPA(I) = CP(I,1)                                                 COMDC290
      DO 10 N = 1,NC1                                                  COMDC300
      CPR(I) = CPR(I) + FACTOR(N,M) * CP(I,N)                          COMDC310
      EP(I) = EP(I) + FACTOR(N,M) * EPS(I,N)                           COMDC320
   10 CONTINUE                                                         COMDC330
   20 CONTINUE                                                         COMDC340
C                                                                      COMDC350
C  DEFINE CP FOR THE FIRST RUN COMPOSITE CASE                          COMDC360
      IF(M .EQ. 1) CALL READMS(1,CPA,NEWMAX,IREAD)                     COMDC370
      DO 30 I = 1,NEWMAX                                               COMDC380
      CPR(I) = 0.00                                                    COMDC390
   30 CONTINUE                                                         COMDC400
      IREAD = IREAD - 1                                                COMDC410
      DO 60 N = 1,NC1                                                  COMDC420
      IREAD = IREAD + 1                                                COMDC430
      IF(FACTOR(N,M) .EQ. 0.0) GO TO 60                                COMDC440
   40 CALL READMS(1,CPREAD,NEWMAX,IREAD)                               COMDC450
      DO 50 I = 1,NEWMAX                                               COMDC460
      CPR(I) = CPR(I) + FACTOR(N,M) * CPREAD(I)                        COMDC470
   50 CONTINUE                                                         COMDC480
   60 CONTINUE                                                         COMDC490
C  CALCULATE THE COMPOSITE CASE DERIVATIVES                            COMDC500
   70 CALL SUMIT1(CPC,CPA,CPP,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,CMU,XL,TL, COMDC510
     1    XMC,AREA,CNPC,CNPA,CNP2,CLLPA,CLLPC,CYPC,CYPA,CYP2,          COMDC520
     2    NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)                             COMDC530
   80 CALL SUMIT2(CPC,CPA,CPRC,CPRA,DEL,EP,CHORD,Y,DELTA,AREA,CMU,XL,TL,COMDC540
     1 XMC,CNPC,CNPA,CNRA2,CNR2C,CNR2A,CNP2A2,CYPC,CYRA,CYRA2,          COMDC550
     2 CYR2C,CYR2A,CYP2A2,NW,IJ,NROWS,ISYMM,NEWMAX)                   COMDC560
C                                                                      COMDC570
C  PRINT A SUMMARY TABLE OF ALL STABILITY DERIVATIVES                  COMDC580
   90 CALL STABLE(CLO,CMO,CMOMC,CLLP,CNPC,CNPA,CNP2,CYP2,CYPA,CYP2,    COMDC590
     1    CLLPC,CLLPA,CNPC,CNRA,CNRA2,CNR2C,CNR2A,CNP2A2,              COMDC600
     2    CYR2C,CYRA,CYRA2,CYR2A,CYR2A,CYP2A2,MCASE)                   COMDC610
  100 CONTINUE                                                         COMDC620
      RETURN                                                           COMDC630
      END                                                              COMDC640
```

C22

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK SUMIT1
      SUBROUTINE SUMIT1(CPC,CPA,CPP,CPPC,CPPA,DEL,EPS,CHORD,Y,DELTA,CMU,SUM1C010
     1  XL,TL,XMC,AREA,CNPC,CNPA,CNP2,CLLPC,CLLRA,CYPC,CYPA,CYP2,      SUM1C020
     2  NW,IJ,NROWS,ISYMM,NEWMAX,IFUDGE)                               SUM1C030
C                                                                      SUM1C040
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE   SUM1C050
C  THE TERMS OF CNP AND CLLR DERIVATIVES                               SUM1C060
C                                                                      SUM1C070
      COMMON /DERIV/ UO(40),CLO,CMO,CMOMC                              SUM1C080
      DIMENSION CP(NEWMAX),CPA(NEWMAX),CPP(600),CPPC(600),CPPA(600)    SUM1C090
      DIMENSION DEL(601),EPS(40)                                       SUM1C100
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)        SUM1C110
      DIMENSION NW(40),IJ(40)                                          SUM1C120
C                                                                      SUM1C130
C  INITIALIZE THE DERIVATIVE TERMS                                     SUM1C140
      CNP0 = 0.00                                                      SUM1C150
      CNPA = 0.00                                                      SUM1C160
      CNP2 = 0.00                                                      SUM1C170
      CYP0 = 0.00                                                      SUM1C180
      CYPA = 0.00                                                      SUM1C190
      CYP2 = 0.00                                                      SUM1C200
      CLLPC = 0.00                                                     SUM1C210
      CLLRA = 0.00                                                     SUM1C220
C                                                                      SUM1C230
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE                            SUM1C240
      I = 0                                                            SUM1C250
      DO 100 K = 1,NROWS                                               SUM1C260
C                                                                      SUM1C270
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE                           SUM1C280
C  LEADING EDGE CONTRIBUTION                                           SUM1C290
      I = I + 1                                                        SUM1C300
C  YAWING DUE TO ROLLING                                               SUM1C310
   20 TERM0 = DEL(I) * (CPP(I) + 0.5C*CPP(I+1))                        SUM1C320
      EP = -EPS(I) / (57.295779 * UO(K))                               SUM1C330
      DP0 = TERM0 * EP                                                 SUM1C340
      DPA = TERM0                                                      SUM1C350
      SPC = 0.3490658 * DEL(I) * CPP(I) * CPC(I)                       SUM1C360
      SPA = 0.3490658 * DEL(I) * CPP(I) * CPA(I)                       SUM1C370
      SP2 = 0.1745329 * DEL(I) * CPP(I)**2                             SUM1C380
C  ROLLING DUE TO YAWING                                               SUM1C390
   30 CLC0 = DEL(I) * (CPPC(I) + 0.5C*CPPC(I+1))                       SUM1C400
      CLCA = DEL(I) * (CPPA(I) + 0.5C*CPPA(I+1))                       SUM1C410
C                                                                      SUM1C420
C  REGULAR EVD CONTRIBUTIONS                                           SUM1C430
   40 NWK = NW(K)                                                      SUM1C440
      DO 70 L = 2,NWK                                                  SUM1C450
      I = I + 1                                                        SUM1C460
      CPP1 = CPP(I+1)                                                  SUM1C470
      CPPC1 = CPPC(I+1)                                                SUM1C480
      CPPA1 = CPPA(I+1)                                                SUM1C490
      IF(L .LT. NWK) GO TO 50                                          SUM1C500
      CPP1 = 0.00                                                      SUM1C510
      CPPC1 = 0.00                                                     SUM1C520
      CPPA1 = 0.00                                                     SUM1C530
      IF(CMU(K) .LT. 0.0001) GO TO 50                                  SUM1C540
      IJK = IJ(K)                                                      SUM1C550
      CPP1 = CPP(IJK)                                                  SUM1C560
      CPPC1 = CPPC(IJK)                                                SUM1C570
      CPPA1 = CPPA(IJK)                                                SUM1C580
   50 EP = -EPS(I) / (57.295779 * UO(K))                               SUM1C590
      TERM0 = 0.5C * DEL(I) * (CPP(I) + CPP1)                          SUM1C600
      DP0 = DP0 + TERM0 * EP                                           SUM1C610
      DPA = DPA + TERM0                                                SUM1C620
   60 CLC0 = CLC0 + 0.50 * DEL(I) * (CPPC(I) + CPPC1)                  SUM1C630
      CLCA = CLCA + 0.50 * DEL(I) * (CPPA(I) + CPPA1)                  SUM1C640
   70 CONTINUE                                                         SUM1C650
C                                                                      SUM1C660
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE                            SUM1C670
      FACTOR = CHORD(K) * Y(K) * DELTA(K)                              SUM1C680
      FACTO = CHORD(K) * DELTA(K) * TL(K)                              SUM1C690
      FACT = FACTO * (XL(K)-XMC)                                       SUM1C700
   80 CNP0 = CNP0 + (DP0 - SP0 * FACTOR - SP0 * FACT                   SUM1C710
      CNPA = CNPA + (DPA/57.295779 - SPA) * FACTOR - SPA * FACT        SUM1C720
      CYP2 = CYP2 + SP2 * FACTOR + SP2 * FACT                          SUM1C730
      CYP0 = CYP0 + SP0 * FACTO                                        SUM1C740
      CYPA = CYPA + SPA * FACTO                                        SUM1C750
      CYP2 = CYP2 + SP2 * FACTO                                        SUM1C760
   90 CLLRC = CLLRC + CLC0 * FACTOR                                    SUM1C770
      CLLRA = CLLRA + CLCA * FACTOR                                    SUM1C780
  100 CONTINUE                                                         SUM1C790
C                                                                      SUM1C800
C  PUT THE TERMS IN FINAL FORM                                         SUM1C810
      FACTOR = 1.00 / AREA                                             SUM1C820
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA                            SUM1C830
  110 CNP0 = FACTOR * CNP0                                             SUM1C840
      CNPA = FACTOR * CNPA                                             SUM1C850
      CNP2 = -FACTOR * CNP2                                            SUM1C860
      CLLRC = -FACTOR * CLLRC                                          SUM1C870
      CLLRA = -FACTOR * CLLRA                                          SUM1C880
      FACTOR = 2.00 * FACTOR                                           SUM1C890
      CYP0 = FACTOR * CYP0                                             SUM1C900
      CYPA = FACTOR * CYPA                                             SUM1C910
      CYP2 = FACTOR * CYP2                                             SUM1C920
      IF(IFUDGE .NE. 0) GO TO 120                                      SUM1C930
      CNP0 = 0.00                                                      SUM1C940
      CYP0 = 0.00                                                      SUM1C950
  120 IF(ISYMM .GE. 0) GO TO 130                                       SUM1C960
      CNP2 = 0.00                                                      SUM1C970
      CYP2 = 0.00                                                      SUM1C980
  130 RETURN                                                           SUM1C990
      END                                                              SUM1C000
```

```
*DECK SUMIT2
      SUBROUTINE SUMIT2(CPC,CPA,CPP0,CPPA,DEL,EPS,CHORD,Y,DELTA,AREA,  SUM2C010
     1  CMU,XL,TL,XMC,CNPC,CNRA,CNRA2,CNR2C,CNR2A,CNR2A2,CYPC,CYRA,     SUM2C020
     2  CYR2C,CYR2A,CYR2A2,NW,IJ,NROWS,ISYMM,NEWMAX)                    SUM2C030
C                                                                      SUM2C040
C  THIS SUBROUTINE INTEGRATES CP CHORDWISE AND SPANWISE TO CALCULATE   SUM2C050
C  THE TERMS OF THE YAWING AND SIDE FORCE COEFFICIENTS DUE TO YAWING   SUM2C060
C                                                                      SUM2C070
      COMMON /DERIV/ UO(40),CLO,CMO,CMOMC                              SUM2C080
      DIMENSION CP(NEWMAX),CPA(NEWMAX),CPPC(600),CPPA(600)            SUM2C090
      DIMENSION DEL(601),EPS(40)                                       SUM2C100
      DIMENSION CHORD(40),Y(40),DELTA(40),CMU(40),XL(40),TL(40)        SUM2C110
      DIMENSION NW(40),IJ(40)                                          SUM2C120
C                                                                      SUM2C130
C  INITIALIZE THE YAWING COEFFICIENT TERMS                             SUM2C140
      CNP0 = 0.00                                                      SUM2C150
      CNPA = 0.00                                                      SUM2C160
      CNRA2 = 0.00                                                     SUM2C170
      CNR2C = 0.00                                                     SUM2C180
      CNR2A = 0.00                                                     SUM2C190
      CNR2A2 = 0.00                                                    SUM2C200
      CYPC = 0.00                                                      SUM2C210
      CYRA = 0.00                                                      SUM2C220
      CYR2A = 0.00                                                     SUM2C230
      CYR2C = 0.00                                                     SUM2C240
      CYR2A = 0.00                                                     SUM2C250
      CYR2A2 = 0.00                                                    SUM2C260
C                                                                      SUM2C270
      I = 0                                                            SUM2C280
      DO 100 K = 1,NROWS                                               SUM2C290
C                                                                      SUM2C300
C  INTEGRATE THE COEFFICIENT TERMS CHORDWISE                           SUM2C310
C  LEADING EDGE CONTRIBUTIONS                                          SUM2C320
      I = I + 1                                                        SUM2C330
   20 TERM0 = DEL(I) * (CPPC(I) + 0.5C*CPPC(I+1))                      SUM2C340
      TERMA = DEL(I) * (CPPA(I) + 0.5C*CPPA(I+1))                      SUM2C350
      EP = -EPS(I) / (57.295779 * UO(K))                               SUM2C360
      DP0 = TERM0 * EP                                                 SUM2C370
      DPA = TERM0/57.295779 + TERMA * EP                               SUM2C380
      DPA2 = TERMA                                                     SUM2C390
   30 SP0 = 0.3490658 * DEL(I) * CPP0(I) * CPC(I)                      SUM2C400
      SPA = 0.3490658 * DEL(I) * (CPPC(I)*CPA(I) + CPPA(I)*CPC(I))     SUM2C410
      SPA2 = 0.3490658 * DEL(I) * (CPPA(I)*CPA(I))                     SUM2C420
      IF(ISYMM .LT. 0) GO TO 40                                        SUM2C430
      SP2C = 0.1745329 * DEL(I) * CPPC(I)**2                           SUM2C440
      SR2A = 0.3490658 * DEL(I) * CPPC(I)*CPPA(I)                      SUM2C450
      SR2A2 = 0.1745329 * DEL(I) * CPPA(I)**2                          SUM2C460
C                                                                      SUM2C470
C  REGULAR EVD CONTRIBUTIONS                                           SUM2C480
   40 NWK = NW(K)                                                      SUM2C490
      DO 70 L = 2,NWK                                                  SUM2C500
      I = I + 1                                                        SUM2C510
      CPP01 = CPP0(I+1)                                                SUM2C520
      CPPA1 = CPPA(I+1)                                                SUM2C530
      IF(L .LT. NWK) GO TO 50                                          SUM2C540
      CPP01 = 0.00                                                     SUM2C550
      CPPA1 = 0.00                                                     SUM2C560
      IF(CMU(K) .LT. 0.0001) GO TO 50                                  SUM2C570
      IJK = IJ(K)                                                      SUM2C580
      CPP01 = CPP0(IJK)                                                SUM2C590
      CPPA1 = CPPA(IJK)                                                SUM2C600
   50 EP = -EPS(I) / (57.295779 * UO(K))                               SUM2C610
C  REGULAR EVD CONTRIBUTIONS                                           SUM2C620
      TERM0 = 0.5C * DEL(I) * (CPP0(I)+CPP01)                          SUM2C630
   60 TERMA = 0.5C * DEL(I) * (CPPA(I)+CPPA1)                          SUM2C640
      DP0 = DP0 + TERM0 * EP                                           SUM2C650
      DPA = DPA + TERM0/57.295779 + TERMA * EP                         SUM2C660
      DPA2 = DPA2 + TERMA                                              SUM2C670
   70 CONTINUE                                                         SUM2C680
C                                                                      SUM2C690
C  INTEGRATE THE COEFFICIENT TERMS SPANWISE                            SUM2C700
      FACTOR = CHORD(K) * Y(K) * DELTA(K)                              SUM2C710
      FACTO = CHORD(K) * TL(K) * DELTA(K)                              SUM2C720
      FACT = FACTO * (XL(K)-XMC)                                       SUM2C730
   80 CNP0 = CNP0 + (DP0 - SP0 * FACTOR - SP0 * FACT                   SUM2C740
      CNPA = CNPA + (DPA-SPA) * FACTOR - SPA * FACT                    SUM2C750
      CNRA2 = CNRA2 + (DPA2/57.295779-SPA2) * FACTOR - SPA2 * FACT     SUM2C760
      CYP0 = CYP0 + SP0 * FACTO                                        SUM2C770
      CYPA = CYPA + SPA * FACTO                                        SUM2C780
      CYRA = CYRA + SPA2 * FACTO                                       SUM2C790
      IF(ISYMM .LT. 0) GO TO 100                                       SUM2C800
   90 CNR2C = CNR2C - SR2C * FACTOR -SR2C * FACT                       SUM2C810
      CNR2A = CNR2A - SR2A * FACTOR - SR2A * FACT                      SUM2C820
      CNR2A2 = CNR2A2 - SR2A2 * FACTOR - SR2A2 * FACT                  SUM2C830
      CYR2C = CYR2C + SP2C * FACTO                                     SUM2C840
      CYR2A = CYR2A + SP2A * FACTO                                     SUM2C850
      CYR2A2 = CYR2A2 + SR2A2 * FACTO                                  SUM2C860
  100 CONTINUE                                                         SUM2C870
C                                                                      SUM2C880
C  PUT THE TERMS IN FINAL FORM                                         SUM2C890
      FACTOR = 1.00 / AREA                                             SUM2C900
      IF(ISYMM .LT. 0) FACTOR = 2.00 / AREA                            SUM2C910
  110 CNPA = FACTOR * CNPC                                             SUM2C920
      CNPA = FACTOR * CNPA                                             SUM2C930
      CNRA2 = FACTOR * CNRA2                                           SUM2C940
      CNR2C = FACTOR * CNR2C                                           SUM2C950
      CNR2A = FACTOR * CNR2A                                           SUM2C960
      CNR2A2 = FACTOR * CNR2A2                                         SUM2C970
      FACTOR = 2.0C * FACTOR                                           SUM2C980
      CYR0 = FACTOR * CYR0                                             SUM2C990
      CYRA = FACTOR * CYRA                                             SUM2C000
      CYRA2 = FACTOR * CYRA2                                           SUM2C010
      CYR2C = FACTOR * CYR2C                                           SUM2C020
      CYR2A = FACTOR * CYR2A                                           SUM2C030
      CYR2A2 = FACTOR * CYR2A2                                         SUM2C040
  120 RETURN                                                           SUM2C050
      END                                                              SUM2C060
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
*DECK STABLE                                                          STBLC010
      SUBROUTINE STABLE(CLQ,CMQ,CMQMC,CLLP,CNPC,CNPA,CNP2,CYPQ,CYPA,  STBLC020
     1       CYP2,CLLP2,CLLPA,CNPC2,CNPA2,CNPA2,CNP2C,CNP2A,CNP2A2,   STBLC030
     2       CYPC2,CYP2A,CYPA2,CYP2C,CYP2A,CYP2A2,MCASE)              STBLC040
C                                                                     STBLC050
C  THIS SUBROUTINE CALCULATES AND PRINTS A COMPLETE SUMMARY TABLE     STBLC060
C  OF ALL STABILITY DERIVATIVE DATA FOR EACH COMPOSITE CASE           STBLC070
C                                                                     STBLC080
      COMMON/COMPOS/FACTOR(10,24),NCC                                 STBLC090
C                                                                     STBLC100
C  PRINT ALL CONSTANT DERIVATIVES                                     STBLC110
      WRITE(6, 10 ) NCASE                                             STBLC120
   10 FORMAT(1H1,33X,13(4H****),1HM /  34X,                           STBLC130
     1    47HM  STABILITY DERIVATIVE DATA FOR COMPOSITE CASE, I3,     STBLC140
     2    34H  */  34X,13(4H****),1HM  /)                            STBLC150
      WRITE(6, 20 ) (N,N=1,10), (FACTOR(N,NCASE),N=1,10)              STBLC160
   20 FORMAT(1H ,48X,24HFUNDAMENTAL CASE FACTORS/ 10X,9(4X,2HA(,I1,1H),STBLC170
     1    2X),3X,2HA(,I2,1H) / 10X,10E10.6)                          STBLC180
      WRITE(6, 30 ) CLQ,CMQ,CMQMC                                     STBLC190
   30 FORMAT(1HC/ 24X,43HLIFT COEFFICIENT DERIVATIVE DUE TO PITCHING,  STBLC200
     1    17H ABOUT XCG, CLQ =,F10.6 /                               STBLC210
     2    17X,41HYAWING MOMENT COEFFICIENT DERIVATIVE ABOUT ORIGIN,  STBLC220
     3    33H DUE TO PITCHING ABOUT XCG, CMQ =,F10.6 /               STBLC230
     4    16X,43HPITCHING MOMENT COEFF DERIVATIVE ABOUT XMC,         STBLC240
     5    35H DUE TO PITCHING ABOUT XCG, CMQMC =, F10.6)             STBLC250
      WRITE(6, 40 ) CLLP,CNPC,CNPA,CNP2                               STBLC260
   40 FORMAT(1HC/ 24X,28HROLLING MOMENT COEFF DERIVATIVE DUE TO,      STBLC270
     1    16H ROLLING, CLLP =,F12.7 ///                              STBLC280
     2    17X,41HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO ROLLING,STBLC290
     3    35H CN(P) MAY BE CALCULATED AS FOLLOWS /                   STBLC300
     4    48X,25HCN(P) = CNPXP + CNP2*P**2 /                         STBLC310
     5    42X,31HWHERE    CNP = CNPC + CNPA*ALPHA /                  STBLC320
     6    53X, 6HCNPC =,F12.7/52X, 6HCNPA =,F12.7/52X,6HCNP2 =,F12.7)STBLC330
      WRITE(6, 50 ) CYPC,CYPA,CYP2                                    STBLC340
   50 FORMAT(1HC/26X,44HSIDE FORCE COEFFICIENT DUE TO ROLLING, CY(P), STBLC350
     1    29H MAY BE CALCULATED AS FOLLOWS/                          STBLC360
     2    48X,25HCY(P) = CYPXP + CYP2*P**2/                          STBLC370
     3    42X,31HWHERE    CYP = CYPC + CYPA*ALPHA/                   STBLC380
     4    52X, 6HCYPC =,F12.7/52X,6HCYPA =,F12.7/52X,6HCNP2 =,F12.7) STBLC390
      WRITE(6, 60 ) CLLR2,CLLP2                                       STBLC400
   60 FORMAT(1HC/15X,38HROLLING MOMENT COEFF DERIVATIVE DUE TO,       STBLC410
     1    52H YAWING ABOUT XCG, CLLR MAY BE CALCULATED AS FOLLOWS /  STBLC420
     2    48X,26HCLLR = CLLPC + CLLPA*ALPHA /                        STBLC430
     3    48X,14HWHERE    CLLPC =,F13.7 / 56X,7HCLLRA =,F13.7)       STBLC440
      WRITE(6, 70 ) CNRC,CNRA,CNRA2,CNR2C,CNR2A,CNR2A2                STBLC450
   70 FORMAT(1HC/12X,42HYAWING MOMENT COEFFICIENT ABOUT XMC DUE TO,   STBLC460
     1    52H YAWING ABOUT XCG, CN(R) MAY BE CALCULATED AS FOLLOWS / STBLC470
     2    48X,25HCN(R) = CNRXR + CNR2*R**2 /                         STBLC480
     3    36X,48HWHERE    CNR = CNRC + CNRA*ALPHA + CNRA2*ALPHA**2 / STBLC490
     4    52X,6HCNRC =,F13.7/52X,6HCNRA =,F13.7/51X,7HCNRA2 =,F13.7/ STBLC500
     5    36X,51HAND    CNR2 = CNR2C + CNR2A*ALPHA + CNR2A2*ALPHA**2/ STBLC510
     6    51X,7HCNR2C =,F13.7/51X,7HCNR2A =,F13.7/52X,6HCNR2A2 =,F13.7)STBLC520
      WRITE(6, 80 ) CYPC,CYPA,CYP2,CYP2C,CYP2A,CYP2A2                 STBLC530
   80 FORMAT(1HC/13X,39HSIDE FORCE COEFFICIENT ABOUT XMC DUE TO,      STBLC540
     1    53H YAWING ABOUT XCG, CY(R) MAY BE CALCULATED AS FOLLOWS / STBLC550
     2    48X,25HCY(R) = CYRXR + CYR2*R**2 /                         STBLC560
     3    36X,48HWHERE    CYR = CYRC + CYRA*ALPHA + CYRA2*ALPHA**2 / STBLC570
     4    52X,6HCYRC =,F13.7/52X,6HCYRA =,F13.7/51X,7HCYRA2 =,F13.7/ STBLC580
     5    36X,51HAND    CYR2 = CYR2C + CYR2A*ALPHA + CYR2A2*ALPHA**2/ STBLC590
     6    51X,7HCYR2C =,F13.7/51X,7HCYR2A =,F13.7/52X,6HCYR2A2 =,F13.7)STBLC600
C                                                                     STBLC610
C  PRINT TABLE OF DERIVATIVE TERMS WHICH DEPEND ON ALPHA              STBLC620
      WRITE(6, 90 )                                                   STBLC630
   90 FORMAT(1H1,32X,34H***  1ST-ORDER  ***/                          STBLC640
     1 33X,43HV VARIATION OF STABILITY TERMS WITH ANGLE OF ATTACK =,/, STBLC650
     2 33X,3H***,13(4H****),3H***,//,                                STBLC660
     3 9X,6HALPHA,12X,3HCNP,10X,4HCNP2,14X,3HCYP,10X,4HCYP2,         STBLC670
     4 14X,6HCLLP,9X,3HCNR,10X,4HCNP2)                               STBLC680
      ALPHA = -11.00                                                  STBLC690
      DO 140 M = 1,41                                                 STBLC700
  100 ALPHA = ALPHA + 1.00                                            STBLC710
      CNP = CNPC + CNPA * ALPHA                                       STBLC720
      CYP = CYPC + CYPA * ALPHA                                       STBLC730
  110 CLLR = CLLRC + CLLRA * ALPHA                                    STBLC740
      CNR = CNRC + CNRA * ALPHA + CNRA2 * ALPHA**2                    STBLC750
  120 CNP2 = CNP2C + CNP2A * ALPHA + CNP2A2 * ALPHA**2                STBLC760
      WRITE(6, 130 ) ALPHA,CNP,CNP2,CYP,CYP2,CLLR,CNR,CNP2           STBLC770
  130 FORMAT(1H ,6X,F10.4,5H   ,2F13.7,5H   ,2F13.7,6H   ,           STBLC780
     1    3E13.7)                                                    STBLC790
  140 CONTINUE                                                        STBLC800
      RETURN                                                          STBLC810
      END                                                             STBLC820
```

```
*DECK STAGE4                                                          STG4C010
      OVERLAY(OVLY,4,0)                                               STG4C020
      PROGRAM STAGE4                                                  STG4C030
C                                                                     STG4C040
C  THIS PROGRAM CONTROLS THE EXECUTION OF UTILITY ROUTINES AND        STG4C050
C  BOUNDARY CONDITION SETUP FOR STABILITY DERIVATIVE RUNS             STG4C060
C                                                                     STG4C070
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE         STG4C080
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)STG4C090
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CMAP,XMC,XCG   STG4C100
      COMMON/SPIRIT/ NEWMAX,NEWCMU,NOALFA,LOGIC,IR                    STG4C105
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XR(40),XI(600),DEL(600), STG4C110
     1            D(40),KK(600),ITYPE(40)                           STG4C115
      COMMON/ECASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),   STG4C120
     1            XHR(40,40),RET(40,40),IFS(40,40),ICT(40),THT(40),NCT,NHT STG4C125
      COMMON/ECASE3/EPS(ACC,10),BETA(ACC,10),THETA(40,10),THS(40,10)  STG4C130
      COMMON/SOLV1/CP(600,10)                                         STG4C135
      DIMENSION CPREAD(420)                                           STG4C140
C                                                                     STG4C150
C  ADD THE EXTRA FUNDAMENTAL CASE FOR DERIVATIVES DUE TO PITCHING     STG4C160
      IF(LOGIC .GT. 1) GO TO 60                                       STG4C170
   10 IF(NCASES .LT. 10) GO TO 30                                     STG4C180
      WRITE(6, 20 )                                                   STG4C190
   20 FORMAT(1HO//14X,44HFUNDAMENTAL CASE 10 HAS BEEN REPLACED BY THE,STG4C200
     1    47H FOLLOWING CASE FOR DERIVATIVES DUE TO PITCHING)        STG4C210
      GO TO 40                                                        STG4C220
   30 NCASES = NCASES + 1                                             STG4C230
   40 CALL RCPICH(XCG,CREF,XI,DEL,EPS,BETA,CHORD,KK,THETA,THS,TST,HL, STG4C240
     1    NW,IW,NJ,IJ,NWT,NMAX,NROWS,NCASES)                         STG4C250
   50 CALL OUT2(NCASES)                                               STG4C260
      RETURN                                                          STG4C270
C                                                                     STG4C280
C  SAVE THE FIRST RUN SOLUTION ON UNIT 1                              STG4C290
   60 ISIZE = NEWMAX                                                  STG4C300
      CALL SAVECP(CP,CPREAD,NMAX,NJT,ISIZE,NCASES)                    STG4C310
C                                                                     STG4C320
C  DEFINE THE FUNDAMENTAL CASES FOR YAWING DERIVATIVES                STG4C330
      NCI = NCASES - 1                                                STG4C340
   70 CALL RCYAW(EPS,BETA,THETA,THS,Y,KK,NWT,NMAX,NROWS,NCI)          STG4C350
C  DEFINE THE LAST FUNDAMENTAL CASE FOR ROLLING DERIVATIVES           STG4C360
   80 CALL RCROLL(EPS,BETA,THETA,THS,TST,Y,NW,NWT,NMAX,NROWS,NCASES)  STG4C370
C                                                                     STG4C380
C  PRINT THE FUNDAMENTAL CASE GEOMETRY                                STG4C390
      IF(IPRINT .GE. 2) GO TO 100                                     STG4C400
      DO 90  N = 1,NCASES                                             STG4C410
      ICASE = N                                                       STG4C420
      CALL OUT2(ICASE)                                                STG4C430
   90 CONTINUE                                                        STG4C440
  100 CONTINUE                                                        STG4C450
      END                                                             STG4C460
```

```
*DECK OUT2                                                            OUT2C010
      SUBROUTINE OUT2(ICASE)                                          OUT2C020
C                                                                     OUT2C030
C  THIS SUBROUTINE PRINTS OUT THE GEOMETRIC DATA DERIVED FROM THE     OUT2C040
C  SECTIONAL METHOD INPUT                                             OUT2C050
C                                                                     OUT2C060
      COMMON/MATHEW/NCASES,ISYMM,IPRINT,JETFLG,IGTYPE,IHINGE         OUT2C070
      COMMON/MARK/NROWS,NROWSJ,NWT,NJT,NMAX,NW(40),NJ(40),IW(40),IJ(40)OUT2C075
      COMMON/LINE/ TITLE(8)                                           OUT2C080
      COMMON/JOHN/ AREA,SPAN,ARATIO,TR,SWEEP,CREF,CMAC,CMAP,XMC,XCG   OUT2C090
      COMMON/GEOM1/Y(40),CHORD(40),DELTA(40),XR(600),XI(600),DEL(600),OUT2C100
     1            D(40),KK(600),ITYPE(40)                           OUT2C110
      COMMON/GEOM2/XLEAD(40),XTRAIL(40),TANLE(40),TANTE(40)           OUT2C120
      COMMON/ECASE2/TST(40,10),HL(40,10),DJ(40),ACTE(40),AC(20,40),   OUT2C125
     1            XHR(40,40),RET(40,40),IFS(40,40),ICT(40),THT(40),NCT,NHTOUT2C130
      COMMON/ECASE3/EPS(ACC,10),BETA(ACC,10),THETA(40,10),THS(40,10)  OUT2C135
      COMMON/INDATA/ARE,SPA,CRE,XM,CMA,XC,NRO,NC,ISY,IPR,JET,IGT,IHI  OUT2C140
C                                                                     OUT2C150
C  PRINT CASE TITLE AND GENERAL GEOMETRIC PARAMETERS                  OUT2C160
      IF(ICASE .GT. 1) GO TO 60                                       OUT2C170
   10 WRITE(6, 20 ) TITLE                                             OUT2C180
   20 FORMAT(1H1,33X,13(4H****)//                                     OUT2C190
     1    47X,42HA  E  V  I  S  -  WING  COMPUTER  PROGRAM  */       OUT2C200
     2    47X,1(4H****)//22X,8A10)                                   OUT2C210
      CMA = CMAC * SPA / 2.0                                          OUT2C220
      WRITE(6, 40 ) AREA,ARE,SPAN,SPA,CREF,CRE,XMC,XM,CMAC,CMA,ARATIO,OUT2C230
     1            ARATIO,XCG,XC                                     OUT2C240
   40 FORMAT(1HO//54X,6HINPUT /                                       OUT2C250
     1       41X,6HAREA =,2F15.6 / 41X,6HSPAN =,2F15.6 /             OUT2C260
     2       41X,6HCREF =,2F15.6 / 42X,5HXMC =,2F15.6 /              OUT2C270
     3       41X,6HCMAC =,2F15.6 / 39X,9HARATIO =,2F15.6 /           OUT2C280
     4       42X,5HXCG =,2F15.6)                                     OUT2C290
      WRITE(6, 50 ) NROWS,NRO,NCASES,NC,ISYMM,ISY,IPRINT,IPR,JETFLG,JET,OUT2C300
     1            IGTYPE,IGT,IHINGE,IHI,NWT,NJT,NMAX                OUT2C310
   50 FORMAT(1HO/ 48X,7HNROWS =,I3,7X,13 / 47X,8HNCASES =,I3,7X,I3 /  OUT2C320
     1       48X,7HISYMM =,I3,7X,I3 / 47X,8HIPRINT =,I3,7X,I3 /      OUT2C330
     2       47X,8HJETFLG =,I3,7X,I3 / 47X,8HIGTYPE =,I3,7X,I3 /     OUT2C340
     3       47X,8HIHINGE =,I3,7X,I3 ///                             OUT2C350
     4       43X,25HNUMBER OF WING ELEMENTS =,I4 /                   OUT2C360
     5       43X,25HNUMBER OF JET ELEMENTS  =,I4 /                   OUT2C370
     6       42X,26HTOTAL NUMBER OF ELEMENTS =,I4)                   OUT2C380
      JJ = 0                                                          OUT2C390
      NWT                                                             OUT2C400
C                                                                     OUT2C410
C  PRINT FUNDAMENTAL CASE HEADER                                      OUT2C420
      WRITE(6, 70 ) ICASE                                             OUT2C430
   70 FORMAT(1H1,23X,1H*,19(4H****)//                                 OUT2C440
     1    24X,54H*  ELEMENT GEOMETRY DATA AND FUNDAMENTAL CASE DATA FOR,OUT2C450
     2    17H FUNDAMENTAL CASE,I3,3H  */24X,1H*,19(4H****))          OUT2C460
      ILINES = 3                                                      OUT2C470
      DO 260 K = 1,NROWS                                              OUT2C480
C                                                                     OUT2C490
C  PRINT SECTIONAL DATA                                               OUT2C500
      WRITE(6, 80 ) K,Y(K),DELTA(K),XLEAD(K),XTRAIL(K),CHORD(K),TANLE(K)OUT2C510
```

C24

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
   80 FORMAT(1HO,11H*** SECTION,I3,4H ***,2X,3HY =,F10.6,2X,7HDELTA =,  OUT20540
     1 F10.6,2X,7HXLEAD =,F10.6,2X,8HXTRAIL =,F10.6,2X,7HCHORD =,F10.6,  OUT20550
     2 2X,7HTANLE =,F10.6)                                               OUT20560
C                                                                        OUT20570
C  PRINT CHORDWISE DATA ON WING                                          OUT20580
      NWK = NW(K)                                                        OUT20590
      WRITE(6, 90 ) NWK,TST(K,LCASE),HL(K,LCASE),THS(K,LCASE)            OUT20600
   90 FORMAT(1HC,21HWING ELEMENTS    NW =,I3,5X, 7HTWIST =,F10.6,5X,      OUT20610
     1       4HHL =,F10.6,5X, 9HTHETA S =,F10.6)                         OUT20620
      WRITE(6, 100 ) (XB(J+L),L=1,NWK)                                   OUT20630
  100 FORMAT(1H ,14X,2HXB, 10F11.6/17X,10F11.6)                          OUT20640
      IF(LCASE .GT. 1) GO TO 130                                         OUT20650
      WRITE(6, 110 ) (XII(J+L),L=1,NWK)                                  OUT20660
  110 FORMAT(1H ,14X, 2HXI, 10F11.6/17X,10F11.6)                         OUT20670
      WRITE(6, 120 ) (DEL(J+L),L=1,NWK)                                  OUT20680
  120 FORMAT(1H ,13X,3HDEL,10F11.6/17X,10F11.6)                          OUT20690
  130 IF(ICT(K) .EQ. 0) GO TO 150                                        OUT20700
      ICK = ICT(K)                                                       OUT20710
      WRITE(6, 140 ) (AC(L,ICK),L=1,NWK)                                 OUT20720
  140 FORMAT(1H ,13X,6HCAMBER,10F11.6/17X,10F11.6)                       OUT20730
  150 WRITE(6, 160 ) (EPS(J+L,LCASE),L=1,NWK)                            OUT20740
  160 FORMAT(1H ,13X,3HEPS,10F11.6/17X,10F11.6)                          OUT20750
      WRITE(6, 170 ) (BETA(J+L,LCASE),L=1,NWK)                           OUT20760
  170 FORMAT(1H ,12X,4HBETA,10F11.6/17X,10F11.6)                         OUT20770
      WRITE(6, 180 ) (ITYPE(J+L),L=1,NWK)                                OUT20780
  180 FORMAT(1H ,12X,4HTYPE,10(3X,I2,6X)/17X,10(3X,I2,6X))               OUT20790
      J = J + NWK                                                        OUT20800
      IL = 1                                                             OUT20810
      IF(NWK .GT. 9) IL = 2                                              OUT20820
      ILINES = ILINES + 4 + 4*IL                                         OUT20830
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL                            OUT20840
C                                                                        OUT20850
C  PRINT CHORDWISE DATA ON JET                                           OUT20860
      NJK = NJ(K)                                                        OUT20870
      IF(NJK .GT. 0) GO TO 200                                           OUT20880
      WRITE(6, 190 )                                                     OUT20890
  190 FORMAT(1H ,9X,19HTHIS ROW HAS NO JET)                              OUT20900
      ILINES = ILINES + 1                                                OUT20910
      GO TO 230                                                          OUT20920
  200 WRITE(6, 210 ) NJK,D(K),DJ(K),ACTE(K),THETA(K,LCASE)               OUT20930
  210 FORMAT(1HO,1X,20HJET ELEMENTS     NJ =,I3,5X,3HD =,F10.6,5X,4HDJ =,  OUT20940
     1       F10.6,5X,4HACTE =,F10.6,5X,7HTHETA =,F10.6)                 OUT20950
      WRITE(6, 100 ) (XB(JJ+L),L=1,NJK)                                  OUT20960
      IF(LCASE .GT. 1) GO TO 220                                         OUT20970
      WRITE(6, 110 ) (XI(JJ+L),L=1,NJK)                                  OUT20980
      WRITE(6, 120 ) (DEL(JJ+L),L=1,NJK)                                 OUT20990
  220 WRITE(6, 170 ) (BETA(JJ+L,LCASE),L=1,NJK)                          OUT21000
      WRITE(6, 180 ) (ITYPE(JJ+L),L=1,NJK)                               OUT21010
      JJ = JJ + NJK                                                      OUT21020
      IL = 1                                                             OUT21030
      IF(NJK .EQ. 10) IL = 2                                             OUT21040
      ILINES = ILINES + 1 + 3 * IL                                       OUT21050
      IF(LCASE .EQ. 1) ILINES = ILINES + 2*IL                            OUT21060
  230 IF(K .EQ. NROWS) GO TO 260                                         OUT21070
      NWK1 = NW(K+1)                                                     OUT21080
      IL = 1                                                             OUT21090
      IF(NWK1 .GT. 9) IL = 2                                             OUT21100
      NEXT = 4 + 4*IL                                                    OUT21110
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL                                OUT21120
      NJK1 = NJ(K+1)                                                     OUT21130
      IL = 1                                                             OUT21140
      IF(NJK1 .EQ. 10) IL = 2                                            OUT21150
      NEXT = NEXT + 1                                                    OUT21160
      IF(NJK1 .EQ. 0) GO TO 240                                          OUT21170
      NEXT = NEXT + 1 + 3*IL                                             OUT21180
      IF(LCASE .EQ. 1) NEXT = NEXT + 2*IL                                OUT21190
  240 IF(55-ILINES) .GE. NEXT) GO TO 260                                 OUT21200
      WRITE(6, 250 )                                                     OUT21210
  250 FORMAT(1H1)                                                        OUT21220
      ILINES = 1                                                         OUT21230
  260 CONTINUE                                                           OUT21240
      RETURN                                                             OUT21250
      END                                                                OUT21260
```

```
*DECK SAVECP                                                             SAVC0010
      SUBROUTINE SAVECP(CP,DUMMY,NMAX,NJT,ISIZE,NCASES)                  SAVC0020
C                                                                        SAVC0030
C  THIS SURROUTINE SAVES THE CP SOLUTION FOR ALL FUNDAMENTAL CASES       SAVC0040
C  OF THE FIRST STABILITY DERIVATIVE RUN BY STORING ON DIRECT ACCESS     SAVC0050
C                                                                        SAVC0060
      DIMENSION CP(600,10),DUMMY(ISIZE)                                  SAVC0070
C                                                                        SAVC0080
C  FIND THE PROPER PLACE TO WRITE THE OLD SOLUTION                       SAVC0090
   10 IWRITE = NMAX * NJT                                                SAVC0100
C  DEFINE THE DUMMY ARRAY                                                SAVC0110
      DO 30 N = 1,NCASES                                                 SAVC0120
      IWRITE = IWRITE + 1                                                SAVC0130
      DO 20 I = 1,ISIZE                                                  SAVC0140
      DUMMY(I) = CP(I,N)                                                 SAVC0150
   20 CONTINUE                                                           SAVC0160
C                                                                        SAVC0170
C  SAVE THE DATA                                                         SAVC0180
      CALL WRITMS(1,DUMMY,ISIZE,IWRITE)                                  SAVC0190
   30 CONTINUE                                                           SAVC0200
      RETURN                                                             SAVC0210
      END                                                                SAVC0220
```

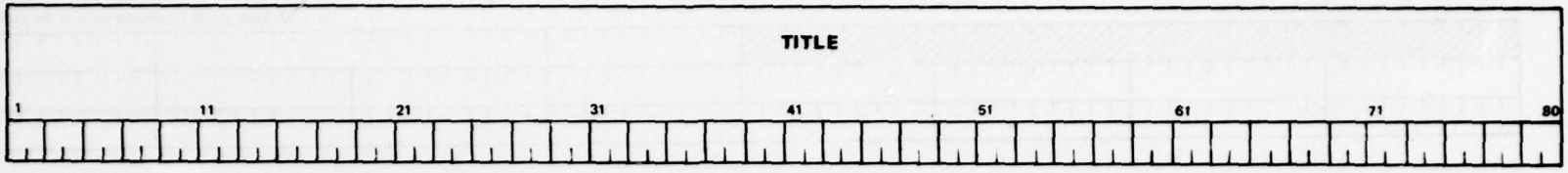
```
*DECK BCPICH                                                             BCP10010
      SUBROUTINE BCPICH(XCG,CREF,XI,DEL,EPS,BETA,C,KK,THETA,THS,TST,HL,  BCP10020
     1            NW,IW,NJ,IJ,NWT,NMAX,NROWS,NI)                         BCP10030
C                                                                        BCP10040
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR                   BCP10050
C  THE PITCHING RATE DERIVATIVE FUNDAMENTAL CASE                         BCP10060
C                                                                        BCP10070
      DIMENSION XI(600),DEL(600),C(4C),KK(600)                           BCP10080
      DIMENSION NW(40),IW(4C),NJ(40),IJ(40)                              BCP10090
      DIMENSION EPS(62C,1C),BETA(600,10),THETA(4C,10),THS(4C,1C)         BCP10100
      DIMENSION TST(4C,1C),HL(4C,1C)                                     BCP10110
C                                                                        BCP10120
C  DEFINE THE CAMBER ANGLES WHICH RESULT FROM PITCHING                   BCP10130
      DO 2C I = 1,NWT                                                    BCP10140
      KKI = KK(I)                                                        BCP10150
   1C EPS(I,N) = 2.CC * (XI(I)+(DEL(I)/2.0)*C(KKI)-XCG) / CREF           BCP10160
      BETA(I,N) = 0.CC                                                   BCP10170
   2C CONTINUE                                                           BCP10180
      IF(NWT .EQ. NMAX) GO TO 50                                         BCP10190
      NWT1 = NWT + 1                                                     BCP10200
      DO 4C I = NWT1,NMAX                                                BCP10210
      EPS(I,N) = C.CC                                                    BCP10220
      BETA(I,N) = 0.CC                                                   BCP10230
   4C CONTINUE                                                           BCP10240
C                                                                        BCP10250
C  DEFINE THE JET ANGLES WHICH RESULT FROM PITCHING                      BCP10260
   5C DO 7C K = 1,NROWS                                                  BCP10270
   6C THETA(K,N) = 0.00                                                  BCP10280
      IJK = IJ(K)                                                        BCP10290
      IF(NJ(K) .GT. 0) THETA(K,N) = 2.00 * (XI(IJK) - XCG) / CREF        BCP10300
      THS(K,N) = 0.00                                                    BCP10310
      TST(K,N) = 0.00                                                    BCP10320
      HL(K,N) = 0.00                                                     BCP10330
   7C CONTINUE                                                           BCP10340
      RETURN                                                             BCP10350
      END                                                                BCP10360
```

```
*DECK BCROLL                                                             BCR00010
      SUBROUTINE BCROLL(EPS,BETA,THETA,THS,TST,Y,                        BCR00020
     1            NW,NWT,NMAX,NROWS,NCASES)                              BCR00030
C                                                                        BCR00040
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR                   BCR00050
C  THE ROLLING RATE DERIVATIVE FUNDAMENTAL CASE                          BCR00060
C                                                                        BCR00070
      DIMENSION EPS(600,1C),BETA(600,1C),THETA(4C,10),THS(40,10),        BCR00080
     1      TST(4C,10)                                                   BCR00090
      DIMENSION Y(40),NW(40)                                             BCR00100
C                                                                        BCR00110
C  DEFINE THE TWIST AND CAMBER ANGLES WHICH RESULT FROM ROLLING          BCR00120
      N = NCASES                                                         BCR00130
      I = C                                                              BCR00140
      DO 4C K = 1,NROWS                                                  BCR00150
   1C TST(K,N) = Y(K)                                                    BCR00160
      THETA(K,N) = TST(K,N)                                              BCR00170
      THS(K,N) = 0.00                                                    BCR00180
      NWK = NW(K)                                                        BCR00190
      DO 3C L = 1,NWK                                                    BCR00200
      I = I + 1                                                          BCR00210
   2C EPS(I,N) = TST(K,N)                                                BCR00220
   3C CONTINUE                                                           BCR00230
   4C CONTINUE                                                           BCR00240
C                                                                        BCR00250
C  DEFINE THE ANGLES ON THE JET                                          BCR00260
      IF(NMAX .EQ. NWT) RETURN                                           BCR00270
      NWT1 = NWT + 1                                                     BCR00280
      DO 6C I = NWT1,NMAX                                                BCR00290
   5C EPS(I,N) = 0.00                                                    BCR00300
   6C CONTINUE                                                           BCR00310
      RETURN                                                             BCR00320
      END                                                                BCR00330
```

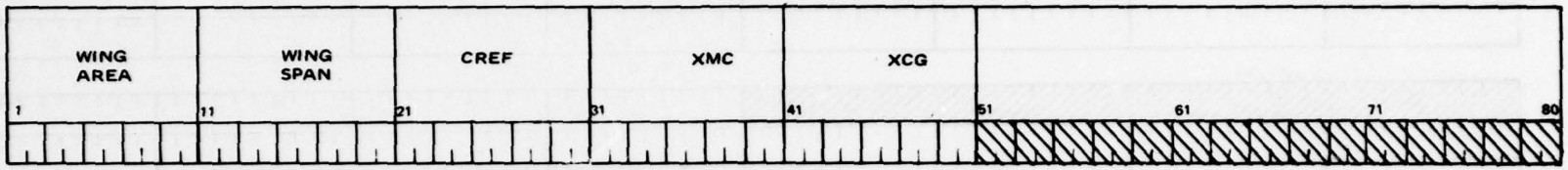
```
*DECK BCYAW                                                              BCY00010
      SUBROUTINE BCYAW (EPS,BETA,THETA,THS,Y,KK,                         BCY00020
     1            NWT,NMAX,NROWS,NCASES)                                 BCY00030
C                                                                        BCY00040
C  THIS SUBROUTINE DEFINES THE BOUNDARY CONDITIONS FOR ALL OF THE        BCY00050
C  YAWING RATE DERIVATIVE FUNDAMENTAL CASES                              BCY00060
C                                                                        BCY00070
      COMMON /DERIV/ UO(40),CLO,CMO,CNOMC                                BCY00080
      DIMENSION EPS(6CC,1C),BETA(600,10),THETA(4C,10),THS(40,10)         BCY00090
      DIMENSION Y(4C),KK(6CC)                                            BCY00100
C                                                                        BCY00110
C  DEFINE THE SECTIONAL NORMALIZED VELOCITY INDUCED BY YAWING            BCY00120
      DO 1C K = 1,NROWS                                                  BCY00130
      UO(K) = Y(K) / 57.295779                                           BCY00140
   1C CONTINUE                                                           BCY00150
C                                                                        BCY00160
C  DEFINE THE ANGLES FOR ALL FUNDAMENTAL CASES                           BCY00170
      DO 8C  N = 1,NCASES                                                BCY00180
      DO 3C I = 1,NWT                                                    BCY00190
      KKI = KK(I)                                                        BCY00200
   2C EPS(I,N) = -UO(KKI) + EPS(I,N)                                     BCY00210
      BETA(I,N) = 0.00                                                   BCY00220
   3C CONTINUE                                                           BCY00230
C                                                                        BCY00240
      DO 5C K = 1,NROWS                                                  BCY00250
   4C THETA(K,N) = -UO(K) + THETA(K,N)                                   BCY00260
      THS(K,N) = 0.00                                                    BCY00270
   5C CONTINUE                                                           BCY00280
      IF(NMAX .EQ. NWT) GO TO 8C                                         BCY00290
      NWT1 = NWT + 1                                                     BCY00300
      DO 7C I = NWT1,NMAX                                                BCY00310
   6C EPS(I,N) = 0.00                                                    BCY00320
      BETA(I,N) = 0.00                                                   BCY00330
   7C CONTINUE                                                           BCY00340
   8C CONTINUE                                                           BCY00350
      RETURN                                                             BCY00360
      END                                                                BCY00370
```


THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

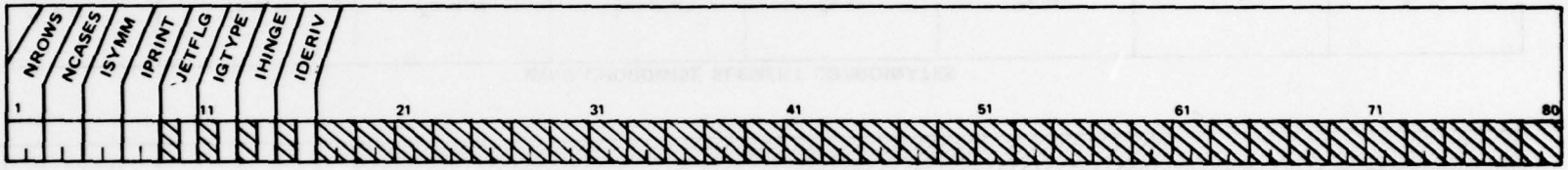
TITLE CARD



BASIC PLANFORM PARAMETER CARD



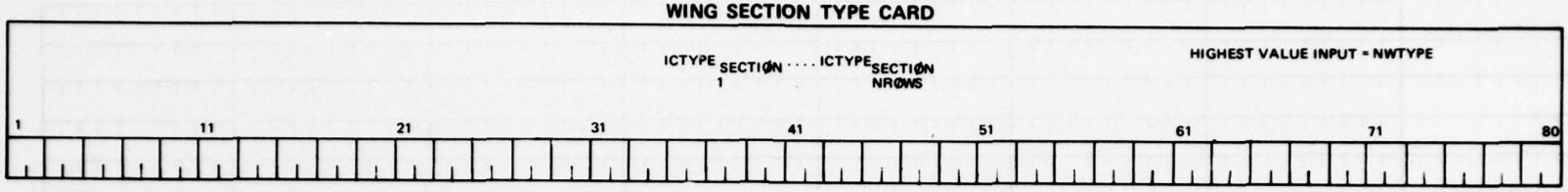
GENERAL CONTROL CARD



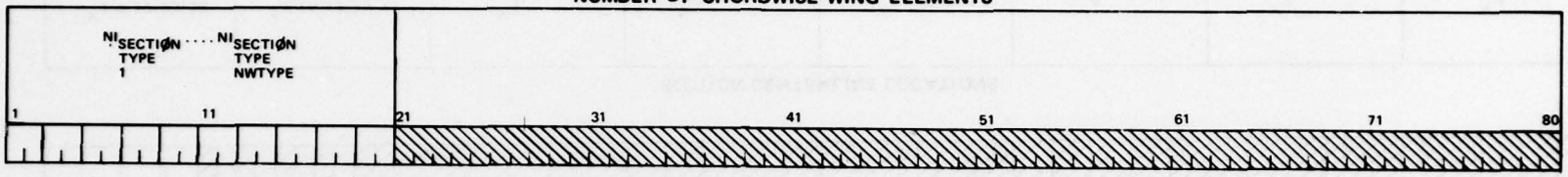
SECTION CENTERLINE LOCATIONS

$Y_{1,9,17,25,33}$	$Y_{2,10,18,26,34}$	$Y_{3,\ldots}$	$Y_{4,\ldots}$	$Y_{5,\ldots}$	$Y_{6,\ldots}$	$Y_{7,\ldots}$	$Y_{8,\ldots}$
1	11	21	31	41	51	61	71 80

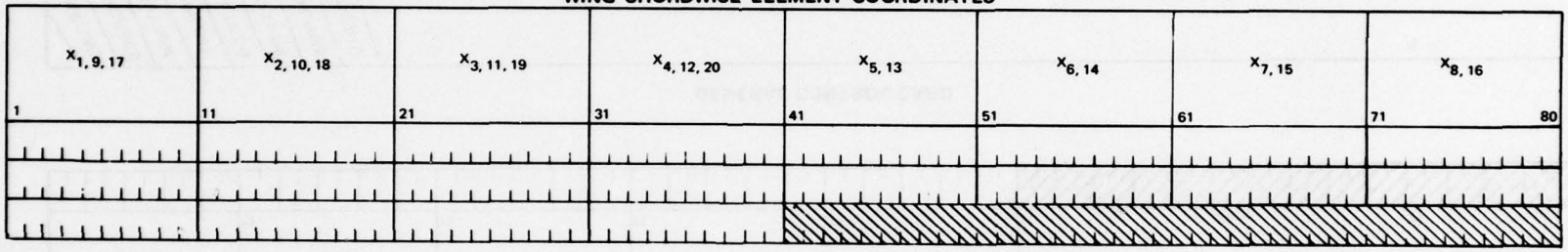
WING SECTION TYPE CARD

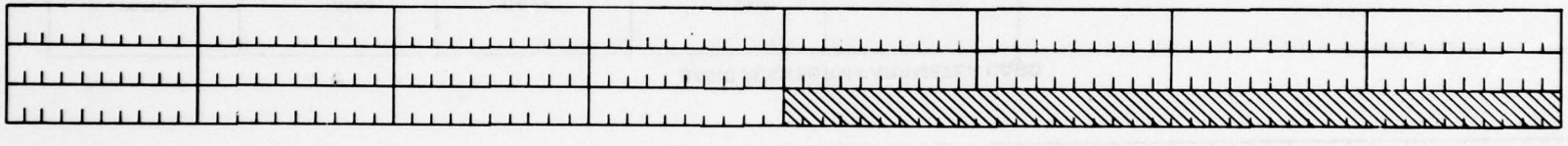


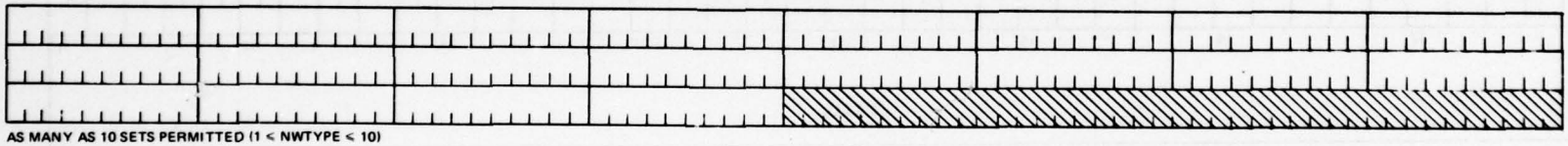
NUMBER OF CHORDWISE WING ELEMENTS



WING CHORDWISE ELEMENT COORDINATES







AS MANY AS 10 SETS PERMITTED (1 ≤ NWTYPE ≤ 10)

LEADING AND TRAILING EDGE COORDINATES

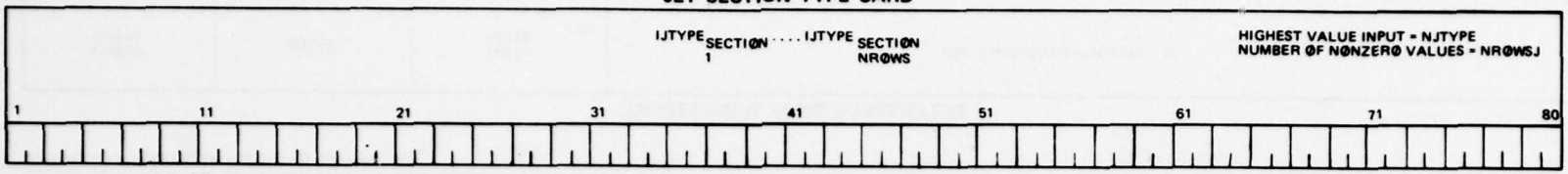
Y	X_{LEAD}	X_{TRAIL}	ØNLY REQUIRED FØR SECTIØNS ADJACENT TØ LEADING ØR TRAILING EDGE BREAKS (THIS INPUT REQUIRED ØNLY IF IGTYPE = 1)
1	11	21	31 41 51 61 71 80
9			

AS MANY AS NRØWS CARDS PERMITTED (NRØWS ≤ 40)

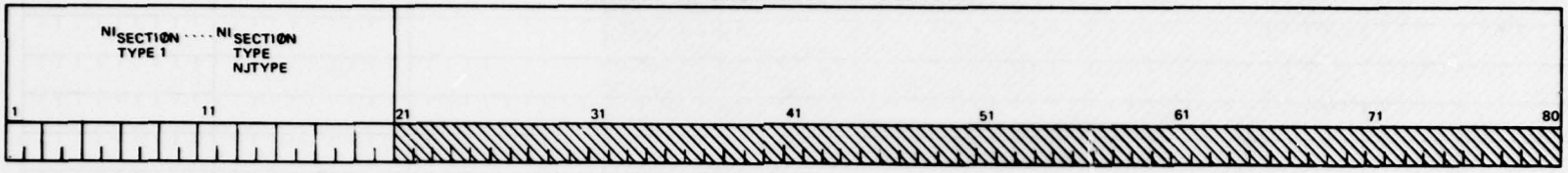
TRAPEZOIDAL WING PARAMETERS

ASPECT RATIØ	SWEEP	TAPER RATIØ	(ØNLY REQUIRED IF IGTYPE = 2)
1	11	21	31 41 51 61 71 80

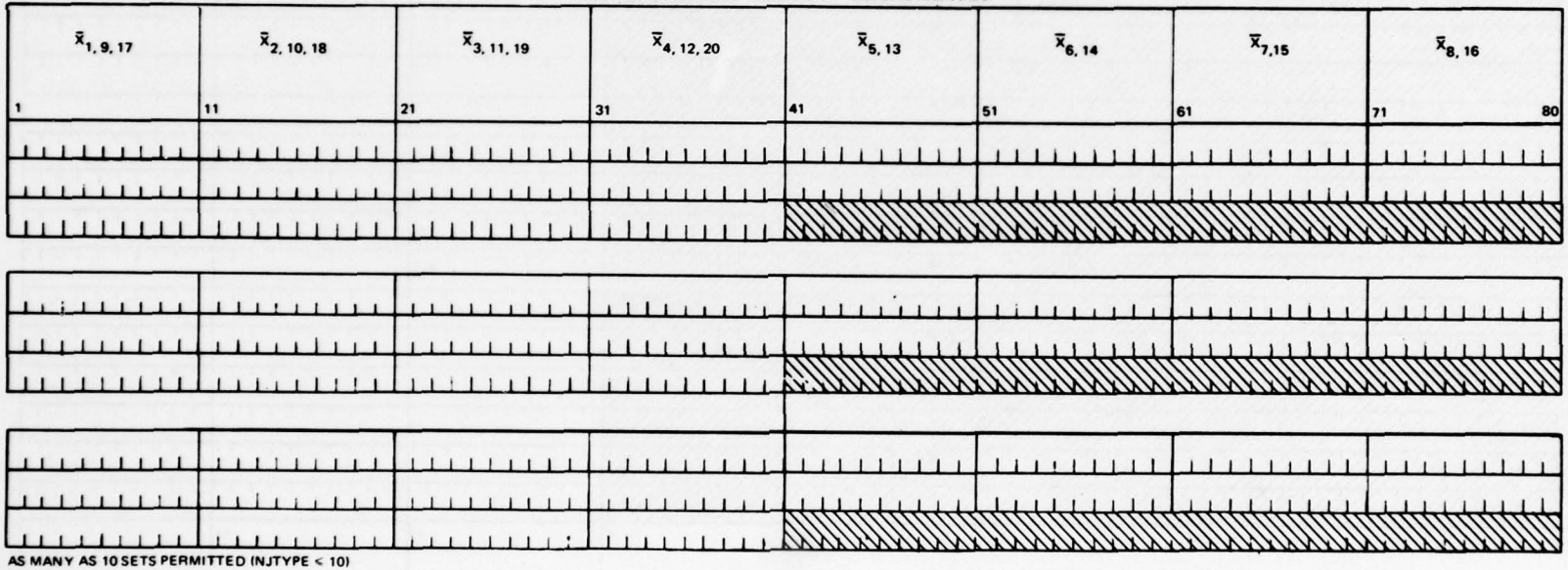
JET SECTION TYPE CARD



NUMBER OF CHORDWISE JET ELEMENTS



JET CHORDWISE ELEMENT COORDINATES



AS MANY AS 10 SETS PERMITTED (NJTYPE < 10)

FUNDAMENTAL CASE CONTROL CARD

INTWST	INHITE	INDELJ	INCAMB	INBETA	
1					11 21 31 41 51 61 71 80

TWIST DISTRIBUTION CARDS

$TWIST_{1,9,\ldots}$	$TWIST_{2,10,\ldots}$	$TWIST_{3,11,\ldots}$	$TWIST_{4,12,\ldots}$	$TWIST_{5,13,\ldots}$	$TWIST_{6,14,\ldots}$	$TWIST_{7,15,\ldots}$	$TWIST_{8,16,\ldots}$
1	11	21	31	41	51	61	71 80

LEADING EDGE HEIGHT CARDS

$HL_{1,9,\ldots}$	$HL_{2,10,\ldots}$	$HL_{3,11,\ldots}$	$HL_{4,12,\ldots}$	$HL_{5,13,\ldots}$	$HL_{6,14,\ldots}$	$HL_{7,15,\ldots}$	$HL_{8,16,\ldots}$
1	11	21	31	41	51	61	71 80

JET DEFLECTION CARDS

$DJ_{1,9,\ldots}$	$DJ_{2,10,\ldots}$	$DJ_{3,11,\ldots}$	$DJ_{4,12,\ldots}$	$DJ_{5,13,\ldots}$	$DJ_{6,14,\ldots}$	$DJ_{7,15,\ldots}$	$DJ_{8,16,\ldots}$
1	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

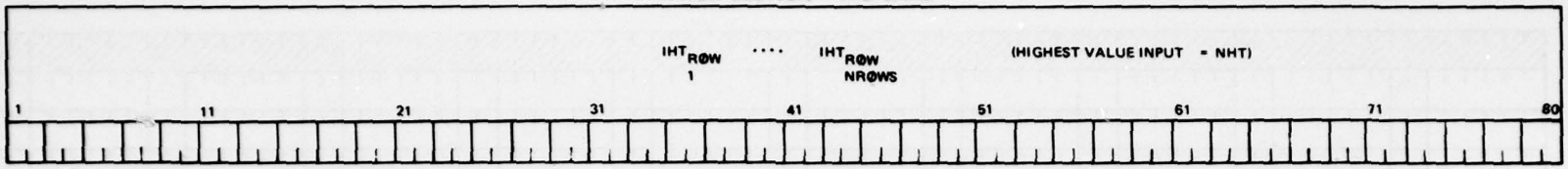
(AS MANY AS 10 SETS PERMITTED ($1 \leq NCT \leq 10$))

TRAILING EDGE CAMBER ANGLE CARDS

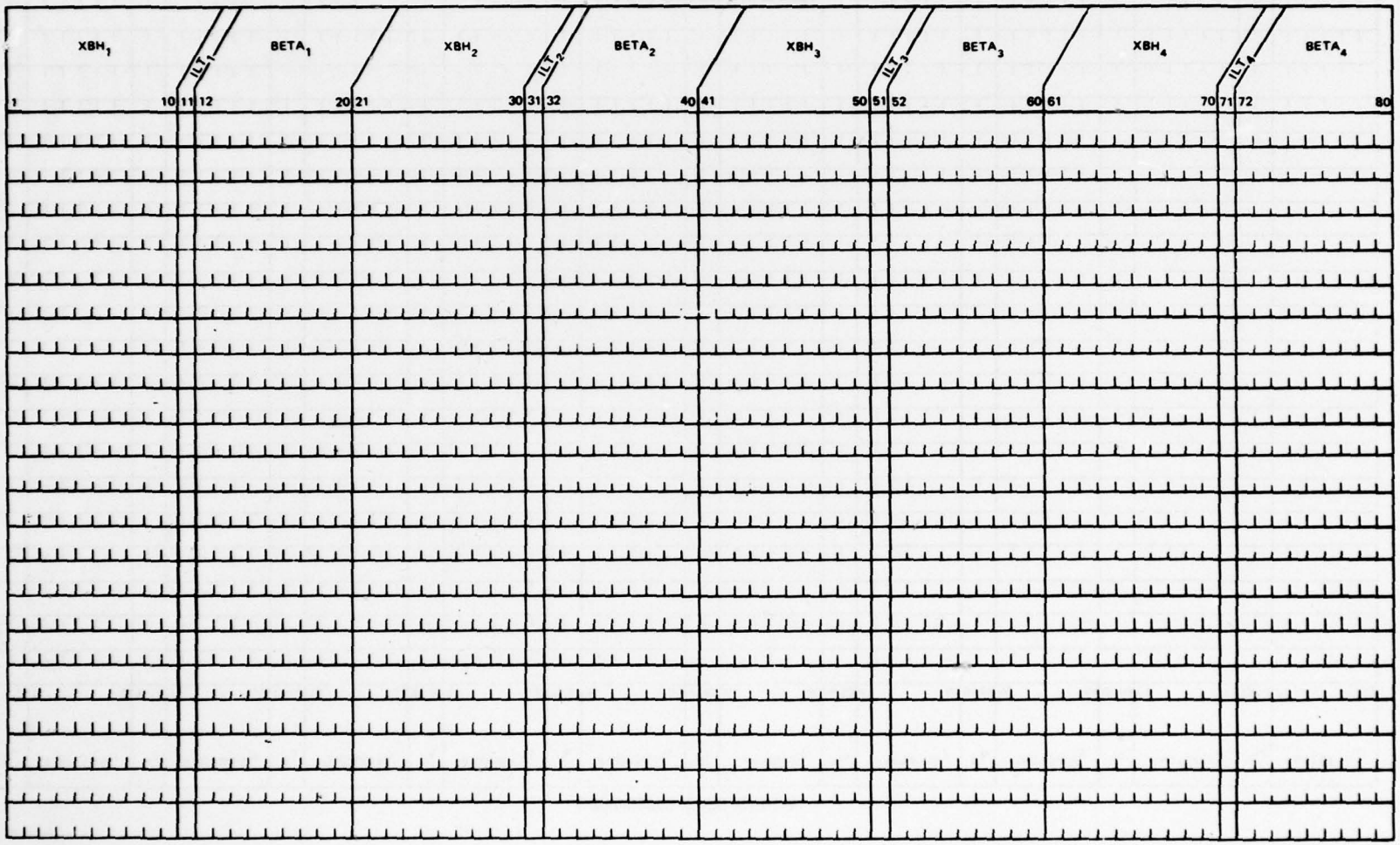
$ACTE_{1,9,...}$	$ACTE_{2,10,...}$	$ACTE_{3,11,...}$	$ACTE_{4,12,...}$	$ACTE_{5,13,...}$	$ACTE_{6,14,...}$	$ACTE_{7,15,...}$	$ACTE_{8,16,...}$
1	11	21	31	41	51	61	71 80

(NRØWSJ VALUES REQUIRED)

HINGE SECTION TYPE CARD



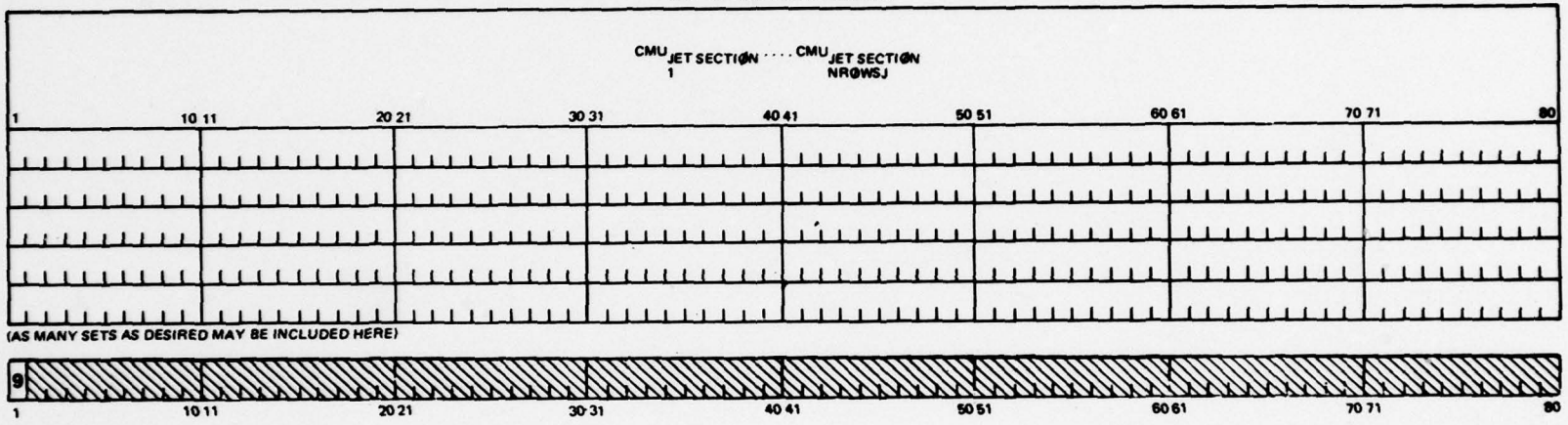
HINGE PARAMETER CARDS



COMPOSITE CASE CARDS

N_1	FACTØR N_1	N_2	FACTØR N_2	N_3	FACTØR N_3	N_4	FACTØR N_4	N_5	FACTØR N_5	N_6	FACTØR N_6	N_7	FACTØR N_7	N_8	FACTØR N_8	N_9	FACTØR N_9	N_{10}	FACTØR N_{10}
1 2	3 8	9 10	11 16	17 18	19 24	25 26	27 32	33 34	35 40	41 42	43 48	49 50	51 56	57 58	59 64	65 66	67 72	73 74	75 80
9																			

JET STRENGTH CARDS



DISTRIBUTION LIST - LIMITED

(VOLUME II ONLY)

Chief of Naval Research
Department of the Navy
Arlington, Virginia 22217
Attn: Aeronautics, Code 461 (TLW) (2 cys)

Naval Air Systems Command
Department of the Navy
Washington, D. C. 20360
Attn: NAVAIR 320 (1 cy)
NAVAIR 5301 (1 cy)

Naval Ship Reasearch & Development Center
Aviation and Surface Effects Department
Bethesda, Maryland 20034
Attn: Director, Aviation Department (2 cys)

Defense Documentation Center
Cameron Station, Building 5
Alexandria, Virginia 22314 (4 cys)

THIS PAGE INTENTIONALLY LEFT BLANK

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)	2a. REPORT SECURITY CLASSIFICATION
McDonnell Douglas Corporation Douglas Aircraft Company Long Beach, California 90846	UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

A THEORETICAL METHOD FOR CALCULATING THE AERODYNAMIC CHARACTERISITICS OF ARBITRARY JET-FLAPPED WINGS. VOL. II EVD JET-WING COMPUTER PROGRAM USER'S MANUAL

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

Michael L. Lopez, Cheng-Chung Shen, Norman F. Wasson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May, 1973	113	2

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-71-C-0250 b. PROJECT NO.	Report No. MDC J5519-02 Vol. II
c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

VOL. II - Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research Aeronautic, Code 461 800 N. Quincy Street Arlington, Virginia 22218

13. ABSTRACT

This report describes the EVD Jet-Wing Computer Program, which is based upon the Elementary Vortex Distribution (EVD) Jet-Wing Lifting Surface Theory described in Volume I of this report. This program provides a capability for determining the aerodynamic characteristics of wings of arbitrary planform, and includes the following:

1. Spanwise and chordwise loading
2. Spanwise variation of induced drag
3. A capability to investigate the effects of:
   a. Part span flaps
   b. Part span blowing
   c. Pitching, rolling, yawing, and sideslip
4. Total lift and induced drag (momentum method), pitching, yawing and rolling moments, etc.

The program has the capabilities for investigating the effects of a variation of leading and trailing flap deflection, camber, twist, jet deflection, and jet momentum.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
AERODYNAMIC STABILITY						
AERODYNAMICS						
BLOWING						
CIRCULATION CONTROL						
COMPUTER PROGRAM						
FINITE ELEMENT METHODS						
FLUID DYNAMICS						
HIGH LIFT SYSTEMS						
JET FLAP						
KUTTA CONDITION						
LIFTING SURFACE THEORY						
NUMERICAL ANALYSIS						
SHORT TAKEOFF AND LANDING AIRCRAFT						
VORTICITY						
WING						