

AD-779 737

DEVELOPMENT OF AN EMPIRICALLY BASED
COMPUTER PROGRAM TO PREDICT THE AERO-
DYNAMIC CHARACTERISTICS OF AIRCRAFT
VOLUME II. PROGRAM USER GUIDE

Roy T. Schemensky

General Dynamics

Prepared for:

Air Force Flight Dynamics Laboratory

November 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

Unclassified

Security Classification

AD 779737

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) General Dynamics Corporation Convair Aerospace Division Fort Worth, Texas		2a. REPORT SECURITY CLASSIFICATION Unclassified
		2b. GROUP
3. REPORT TITLE Development of an Empirically Based Computer Program to Predict the Aerodynamic Characteristics of Aircraft Volume II: Program User Guide		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Report - December 1972 through October 1973		
5. AUTHOR(S) (First name, middle initial, last name) Roy T. Schemensky		
6. REPORT DATE November 1973	7a. TOTAL NO. OF PAGES 223	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. F33615-73-C-3043	8a. ORIGINATOR'S REPORT NUMBER(S) AFFDL-TR-73-144 Volume II	
b. PROJECT NO. 147601		
c.	8b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		
10. DISTRIBUTION STATEMENT Approved for public release; Distribution unlimited.		
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY AFFDL/FXM
13. ABSTRACT This report (Volume II, Program User Guide) describes the computer program developed for the evaluation of the aerodynamic characteristics of large aircraft (bombers, tankers, and transports). The program calculates lift, moment, and drag characteristics of aircraft through the subsonic, transonic, and supersonic speed regimes, and includes the capability of calculating the effect of high-lift systems in both free air and in ground effect for landing and takeoff. The input to the computer program requires the configuration geometry and the aerodynamic conditions for which solutions are desired. The program includes the capability of analyzing both fixed-wing and variable-sweep-wing configurations as well as the aerodynamic characteristics of supercritical wing designs. Details of the methods, equations, and substantiating data for this program are contained in Volume I, Empirical Methods. Although this program was developed to handle the bomber, tanker, transport class of aircraft, it is also applicable to fighter type aircraft without maneuver devices.		

DD FORM 1 NOV 65 1473

Unclassified

Security Classification

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

~~Unclassified~~

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Aerodynamic Characteristics Drag Prediction Lift Prediction Moment Prediction High-Lift Prediction Computer Program Transport Aircraft Drag Due to Lift Vortex Lift						

ia

~~Unclassified~~

Security Classification

DEVELOPMENT OF AN EMPIRICALLY BASED COMPUTER PROGRAM TO PREDICT THE AERODYNAMIC CHARACTERISTICS OF AIRCRAFT

Volume II Program User Guide

R. T. SCHEMENSKY

Approved for public release; distribution unlimited.

NOTICE

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

ACCESSION for	
NTIS	White Section ☒
DDC	Buff Section ☐
UNANNOUNCED	☐
JUSTIFICATION.....	
BY.....	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	Avail. and/or SPECIAL
A	

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

ic

This report was prepared by the Convair Aerospace Division of General Dynamics Corporation, Fort Worth, Texas, for the Air Force Flight Dynamics Laboratory under Contract F33615-73-C-3043, Project 147601. The work reported here was performed in the period December 1972 through October 1973. This report was submitted by the author in October 1973.

The results of this work are documented in two volumes. Volume I presents the methodology developed in this study; Volume II contains the users manual for a computer program which automates these methods.

The work was accomplished under the direction of Mr. J. Kenneth Johnson of the Air Force Flight Dynamics Laboratory (FXM). The author wishes to acknowledge the valuable assistance of Mr. Eugene L. Crosthwait, Convair Aerospace Division, in the development of the empirical methods for the wing lift and wave drag characteristics.

This technical report has been reviewed and is approved.

Philip P. Antonatos

Philip P. Antonatos
Chief, Flight Mechanics Division
Air Force Flight Dynamics Laboratory

TABLE OF CONTENTS

	<u>Page</u>
1. COMPUTER PROGRAM DESCRIPTION	1
2. PROGRAM INPUT	5
3. SAMPLE PROBLEMS	32
4. PROGRAM AND SUBROUTINE DESCRIPTIONS	57
5. PROGRAM LISTING	89

1. COMPUTER PROGRAM DESCRIPTION

The Large Aircraft Aerodynamic Prediction Program has been coded in Fortran Extended Version 3.0 to operate on the CDC 6600 computer facility at W-PAFB. This program is also operational on the General Dynamics CDC 6600 computer (procedure code R1T).

The main program controls the calling of three primary overlay programs XINPT, GEOM, and SURVEY. In turn, XINPT calls two secondary overlay programs INPT and NINPT; while SURVEY calls four secondary overlay programs VGEOM, MCRIT, AERO, and LSHL. These programs bring the desired subroutines into the core and direct the calling of the subroutines to make the necessary computations. Figure 1 shows the overlay structure and the subroutines called from these programs.

Card input occurs on file name TAPE5 and data output occurs on file name TAPE6.

The deck structure for a run consists of a job card, system control cards, end-of-record card, binary object decks, end-of-record card, problem data deck, end-of-file card. The structure of the binary object decks and overlay control cards are listed in Table I. The problem data deck is described in the Program Input Section.

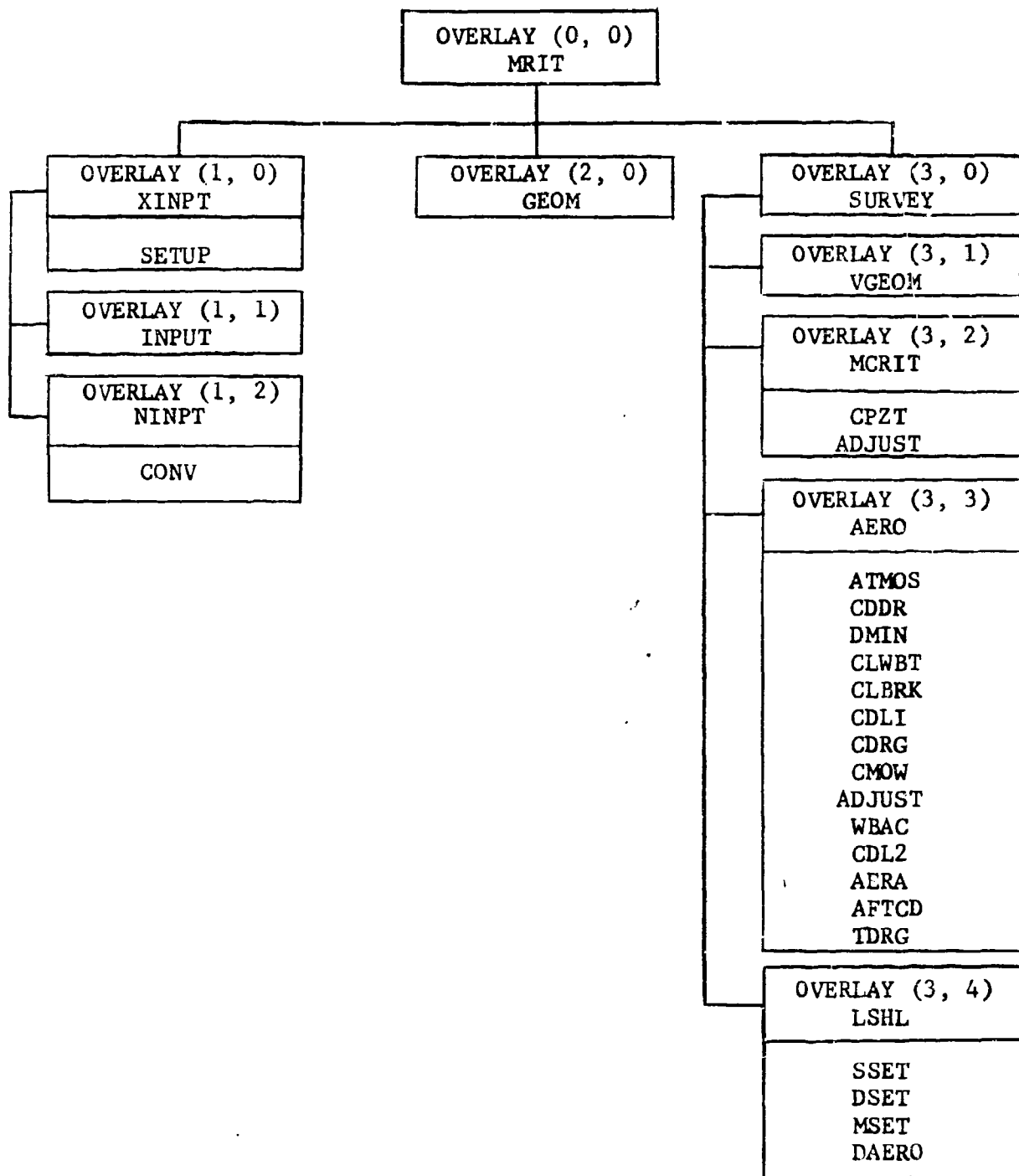


Figure 1 PROGRAM OVERLAY STRUCTURE

Table 1
DECK STRUCTURE FOR OVERLAYS

OVERLAY (OULY, 0, 0)
 PROGRAM MRIT
 SUBROUTINE LNTF
 FUNCTION DLNT
 SUBROUTINE TLNT

OVERLAY (1, 0)
 PROGRAM XINPT
 SUBROUTINE SETUP

OVERLAY (1, 1)
 PROGRAM INPT

OVERLAY (1, 2)
 PROGRAM NINPT
 SUBROUTINE CONV

OVERLAY (2, 0)
 PROGRAM GEOM

OVERLAY (3, 0)
 PROGRAM SURVEY
 SUBROUTINE ADJUST
 BLOCK DATA

OVERLAY (3, 1)
 PROGRAM VGEOM

OVERLAY (3, 2)
 PROGRAM MCRIT
 SUBROUTINE CPZT
 SECT
 CPUOV

OVERLAY (3, 3)
 PROGRAM AERO
 SUBROUTINE CDDR
 CLWBT
 AER2
 TAIL
 AALO
 CLBRK
 CDL1
 ADCL
 KGIN

Table 1 (Cont'd)

SUBROUTINE CDL2
AERA
AFTCD
TDRG
DMIN
FDRG
CFEQ
FFACT
IFACT
WDRG
CDWN
CDWT
CDRG
BDRG
CMOW
WBAC
ACCR
ATMOS
PLSQ
MTXEQ

OVERLAY (3, 4)

PROGRAM LSHL

SUBROUTINE SSET
DSET
MSET
DAERO

2. PROGRAM INPUT

The input to this program is described in the following section. Three data types are specified in the card descriptions. Type A indicates that alphabetic or numeric characters should be entered. Type I indicates integers which should be right-adjusted within the specified columns with no decimal point. Type F indicates a real number and may be tabulated anywhere within the specified columns and should include a decimal point. The program input is illustrated by the sample cases in the Sample Problem Section.

Card 1 - Title - Enter any alphanumeric characters to identify each problem. This data is printed out at the top of each aerodynamic survey problem. Columns 1-66.

Card 2 KPRINT(I), Printout Option Indicators - Setting a given KPRINT indicator to 1 causes the program to print out certain types of data. Columns 1-27. KPRINT(11) through (27) are used for diagnostic purposes to dump data generated in various sub-routines and are not intended to be used for normal computer runs.

<u>KRPINT and Column No.</u>	<u>Print Out</u>
1	Airfoil ordinates and pressure distribution
2 - 10	Not used
11	Dump Subroutine AER2
12	. GEOM
13	. AALO
14	. CDL2
15	BDRG
16	CLBRK
17	AERA
18	WBAC
19	CFEQ
20	CDWW
21	TAIL
22	CDL1
23	CDDR
24	ADJUST
25	CMOW
26	CPZT
27	Dump Subroutine SSET

Card 3 - Input Control Card - Write "FORMAT" or "NAMELIST" beginning in Column 1 if format or namelist input option is desired.

Cards 4, 5, ..., 27 - Configuration Definition Cards, Format Input Option (For NAMELIST input option go to card 43)

The aircraft geometry is represented as a series of bodies for the fuselage, canopy, and stores; open-nosed bodies for the nacelles; and a series of airfoil surfaces for the wing, tail surfaces, pylon and ventral fins. For cranked or complex wing planforms, the wing is represented as a series of surface panels. Figures 2 through 7 illustrate many of the geometric parameters which are used to define an aircraft configuration. In the

input to the program, those parameters that are marked with an asterisk (*) indicate that if zero or blank is entered the program will calculate that parameter using the available geometric data; otherwise the input value will be used.

Card 4 (Required, all Type I)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
4-5	NBODYS	Total number of body types used to represent the configuration
9-10	NNACS	Total number of nacelle types used to represent the configuration
14-15	NSURFS	Total number of airfoil surfaces used to represent the configuration
20	NHT	Enter 1 if a horizontal tail is present
25	NVT	Enter 1 if a single vertical tail is present, 2 for a twin vertical tail
30	ISWP	Variable sweep indicator 0 fixed wing geometry 1 variable sweep wing
34-35	NPNLS	Number of wing panels used to represent the wing surface (≤ 10)
40	IHLS	Enter 1 or 2 if geometry for the high-lift system is to be input. Enter 1 for single high-lift system, enter 2 if the high-lift system is defined by two segments (see Figure 6)
45	IREF	Angle of attack reference indicator 0 referenced to wing root chord plane 1 referenced to fuselage centerline

Card 5 (Required, all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	SREF	Reference area (ft ²)
11-20	*CMAC	Wing mean aerodynamic chord (in.)
21-30	*XCG	Fuselage station for moment reference point (in.). If zero is input the program will use the quarter-chord of the wing MAC.
31-40	ZCG	Height of moment reference point relative to wing root chord plane
41-50	ROUGHK	Surface roughness height for friction drag (in.)
51-60	FMISC	Miscellaneous drag factor as a percentage of total friction and form drag

Card 6 (Required if NBODYS > 0, all Type F)

1-10	BLN(I)	Body length (in.)
11-20	BWID(I)	Body width (in.)
21-30	BHGT(I)	Body height (in.)
31-40	*BAWET(I)	Body wetted area (ft ²)
41-50	BQ(I)	Interference factor
51-60	BNO(I)	Number of bodies of this type

Card 7 (Required if NBODYS > 0, all Type F)

1-10	*BAMX(I)	Body maximum cross-sectional area (in ²)
11-20	BABS(I)	Body base area (in ²)
21-30	BLNS(I)	Nose length (in.)
31-40	BLBT(I)	Boattail length (in.)

Card 7 (Continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
41-50	BASE(I)	Base drag area (in^2)
51-60	BFUS	Aft fuselage upsweep angle (deg.)
61-66	AB	Ratio of aft fuselage width to height in the upswept region

Repeat cards 6 and 7 for $I=1$ to NBODYS. Note the BFUS and AB are only entered for $I=1$ which corresponds to the fuselage input.

Card 8 (Required if NNACS > 0, all Type F)

1-10	ELEN(I)	Nacelle length (in.)
11-20	EWID(I)	Nacelle width (in.)
21-30	EHGT(I)	Nacelle height (in.)
31-40	*EAWET(I)	Total wetted area (ft^2)
41-50	EQF(I)	Interference factor
51-60	ENO(I)	Number of nacelles of this type

Card 9 (Required if NNACS > 0, all Type F)

1-10	*EAMX(I)	Nacelle maximum cross-sectional area (in^2)
11-20	EIN(I)	Nacelle inlet area (in^2)
21-30	EXIT(I)	Nacelle exit area (in^2)
31-40	ELNS(I)	Nose length (in.)
41-50	ELBT(I)	Boattail length (in.)

Repeat cards 8 and 9 for $I=1$ to NNACS.

Card 10 (Required only if NPNLS=1, otherwise go to card 13,
all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	AR	Aspect ratio
11-20	TAPR	Taper ratio
21-30	SWPLE	Leading-edge sweep (deg.)
31-40	SPLAN	Wing planform area (ft^2)
41-50	TWIST	Wing twist (deg.) negative for washout
51-60	WINC	Wing incidence (deg.) relative to fuselage centerline

Card 11 (NPNLS=1, all Type F except as noted)

1-10	TW (Type A)	Type wing section (begin in Col. 1) 64-210 (example 6-series) 00XX-64 (example 4-digit) SUPERCRT (example supercritical) BICONVEX (example biconvex)
11-20	CAM(1)	Wing camber design C_L (decimal value)
21-30	TOC(1)	Wing thickness (decimal value)
31-40	XLEW	X-position of point on wing leading edge (in.)
41-50	YWW	Y-position of point on wing leading edge (in.)
51-60	YB	Y-distance of intersection of wing with fuselage (in.)

Card 12 (NPNLS=1, all Type F)

1-10	SWMT	Maximum thickness sweep (deg.)
11-20	*CBAR(1)	Exposed wing mean aerodynamic chord (in.)

Card 12 (Continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
21-30	*AWET(1)	Wetted area (ft ²)
31-40	CONCL	Wing conical camber design C _L

Skip to card 17 after card 12.

Card 13 (NPNLS > 1, all Type F)

1-10	SPLAN	Planform area (ft ²). For fixed wing configurations if zero is input the program will calculate the theoretical planform area by extending the wing panel geometry that is input to the fuselage centerline. For variable sweep configurations the planform area of the movable panel extended to the centerline must be input. For wings with a highly swept inboard strake, the value of the theoretical planform area of the wing, ignoring the strake, should be input.
11-20	TAPR	Taper ratio of wing planform (required input for variable sweep)
21-30	SWPLE	Leading-edge sweep (deg.) of wing planform. (Required input for variable sweep)

Card 14 (NPNLS > 1, all Type F except as noted)

1-10	TW (Type A)	Type wing section
11-20	TWIST	Wing twist (deg.) negative for washout
21-30	WINC	Wing incidence (deg.)
31-40	SWMT	Average maximum thickness sweep for the entire wing planform (deg.)
41-50	CONCL	Wing conical camber design C _L

Card 15 (NPNLS > 1, all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	CAM(I)	Panel camber design C_L
11-20	TOC(I)	Panel thickness ratio
21-30	XLE(I)	Inboard X-position of panel leading edge (in.)
31-40	YW(I)	Inboard Y-position of panel leading edge (in.)
41-50	CRW(I)	Inboard chord length of panel (in.)
51-60	*CBAR(I)	Panel mean aerodynamic chord length (in.)
61-66	*AWET(I)	Panel wetted area (ft^2)

Repeat card 15 for I=1 to NPNLS

Card 16 (NPNLS > 1, all Type F)

1-10	XLE(NPNLS+1)	Outboard X-position of last panel's leading edge (in.)
11-20	YW(NPNLS+1)	Outboard Y-position of last panel's leading edge (in.)
21-30	CRW(NPNLS+1)	Outboard chord length of panel (in.)

Card 17 (TW = "SUPERCRIT", all Type F)

1-10	XMU	x/c position of maximum thickness on upper surface
11-20	ZMU	upper surface thickness (Z_u/c)
21-30	XML	x/c position of maximum thickness on lower surface
31-40	ZML	Lower surface thickness (must be input as a negative value)

Card 17 (Continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
41-50	ZPTE	Trailing edge slope (deg.)
51-60	ZTHICK	Thickness at trailing edge

Card 18 (ISWP=1, all Type F)

1-10	XPIVOT	X-location of wing pivot (in.)
11-20	YPIVOT	Y-location of wing pivot (in.)
21-30	XAPEX	X-location of centerline apex of movable panel (in.)
31-40	AFTSW	Maximum aft sweep (deg.)
41-50	*AFTCB	Mean aerodynamic chord of movable panel in aft sweep position (in.)
51-60	*AFTOC	Thickness ratio of movable panel in aft sweep position
61-66	*AFTAW	Wetted area of movable panel in aft sweep position

Card 19 (NSURFS $\geq$ 2, all Type F except as noted) Enter horizontal tail geometry first, then vertical tail followed by any additional airfoil surfaces.

1-10	TS(I) (Type A)	Type airfoil surface
11-20	SCAM(I)	Surface camber design C_L
21-30	STOC(I)	Thickness ratio
31-40	SMTSW(I)	Maximum thickness sweep (deg.)
41-50	SHF(I)	Hinge factor or interference factor

Card 20 (NSURFS $\geq$ 2, all Type F)

1-10	SWL(I)	Surface leading-edge sweep (deg.)
11-20	SWT(I)	Trailing-edge sweep (deg.)

Card 20 (Continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
21-30	STAPR(I)	Taper ratio
31-40	SCR(I)	Exposed root chord (in.)
41-50	*SBAR(I)	Exposed mean aerodynamic chord (in.)
51-60	*SAWET(I)	Wetted area (ft ²)

Card 21 (NSURFS $\geq$ 2, NHT=1, all Type F)

1-10	HTLE	X-location of horizontal tail exposed root chord (in.)
11-20	HTY	Y-location of exposed root chord (in.)
21-30	HTZ	Z-location relative to wing chord plane of horizontal tail (in.)
31-40	HTINC	Incidence of horizontal tail (deg.)

Repeat cards 19 and 20 for I=2 to NSURFS.

Card 22 (NPNLS $\geq$ 3, all Type F) Since the aerodynamic center calculation is restricted to one or two panel wings, an equivalent two-panel wing must be defined for moment calculation whenever 3 or more panels are used to define the main wing.

1-10	CLE(1)	X-location of leading edge at inboard span station of inboard panel (in.)
11-20	YC(1)	Y-location of leading edge at inboard span station of inboard panel (in.)
21-30	CCR(1)	Chord length of inboard panel at inboard span station (in.)
31-40	CLE(2)	X-location of leading edge at inboard span station of outboard panel (in.)
41-50	YC(2)	Y-location of leading edge at inboard span station of outboard panel (in.)
51-60	CCR(2)	Chord length of outboard panel at inboard span station (in.)

Card 23 (NPNLS $\geq$ 3, all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	CLE(3)	X-location of leading edge at outboard span station of outboard panel (in.)
11-20	YC(3)	Y-location of leading edge at outboard span station of outboard panel (in.)
21-30	CCR(3)	Chord length of outboard panel at outboard span station (in.)

Card 24 (Required if IHLS $>$ 0, otherwise go to card 30)

1-20	FLAP (Type A)	Type flap (begin in Col. 1) PLAIN FLAP S.S. FLAP (single slotted) D.S. FLAP (double slotted) T.S. FLAP (triple slotted) INPUT (user defined)
21-40	SLAT (Type A)	Type slat (begin in Col. 21) L.E. FLAP KRUGER SLAT SLAT (slotted type) INPUT
41-45	NF (Type I)	If flap type was "INPUT" indicate number of flap sectional data points to be entered, NF $\leq$ 5.
46-50	NS (Type I)	If slat type was "INPUT" indicate number of slat sectional data points to be entered, NS $\leq$ 5.

Card 25 (IHLS $>$ 0, all Type F)

1-10	BF1I	Inboard span station of first flap segment
11-20	BF1O	Outboard span station of first flap segment
21-30	CF1(1)	Flap chord to wing chord ratio of first flap segment

Card 25 (Continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
31-40	BS11	Inboard span station of first slat segment
41-50	BS10	Outboard span station of first slat segment
51-60	CS1	Slat chord to wing chord ratio of first slat segment

Card 26 (IHLS=2, all Type F)

1-10	BF2I	Inboard span station of second flap segment
11-20	BF2O	Outboard span station of second flap segment
21-30	CF2(1)	Flap chord ratio of second flap segment
31-40	BS2I	Inboard span station of second slat segment
41-50	BS2O	Outboard span station of second slat segment
51-60	CS2	Slat chord ratio

Card 27 (FLAP = "D.S." or "T.S.", all Type F) For double-slotted flaps (or triple-slotted) the additional flap chord ratios are then read in.

1-10	CF1(2)	Second flap chord of first flap segment for double- or triple-slotted flap
11-20	CF1(3)	Third flap chord of first flap segment for triple-slotted flap
21-30	CF2(2)	Second flap chord of second flap segment for double- or triple-slotted flap
31-40	CF2(3)	Third flap chord of second flap segment for triple-slotted flap

Card 28 (Required if "INPUT" was used for flap type on card 29,
all Type F)

Flap Section Data

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	DF(I)	Flap deflection (deg.)
11-20	CFOC(I)	Wing chord to wing chord ratio in extended position
21-30	DCLOF(I)	Change in lift at zero angle of attack due to flap deflection
31-40	DCLMF(I)	Change in maximum lift due to flap deflection
41-50	DCDF(I)	Change in profile drag due to flap deflection
51-60	DCMOF	Change in moment at zero angle of attack due to flap deflection

Repeat card 28 for I=1 to NF.

Card 29 (Required if "INPUT" was used for slat type on card 29,
all Type F)

1-10	DS(I)	Slat deflection (deg.)
11-20	CSOC(I)	Wing chord to wing chord ratio in extended position
21-30	DCLOS(I)	Change in lift at zero angle of attack due to slat deflection
31-40	DCLMS(I)	Change in maximum lift due to slat deflection
41-50	DCDS(I)	Change in profile drag due to slat deflection
51-60	DCMDS(I)	Change in moment at zero angle of attack due to slat deflection

Repeat card 29 for I=1 to NS.

Cards 30, 31, 32, and 33 - Problem Control Cards

Card 30 (Required, all Type I)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
4-5	NSURV	Number of high speed lift, drag, moment surveys to be run ($NSURV \leq 20$)
10	NHLSV	Number of low speed, high lift survey conditions to be run ($NHLSV \leq 5$)
14-15	NCLAS	Number of evenly spaced C_L values in the high speed survey
21	IT(1)	Transition location indicator, 1st survey
22	IT(2)	Transition location indicator, 2nd survey
:	:	:
40	IT(20)	Transition location indicator, 20th survey

For the ith survey condition, set IT=0 or blank if the flow is to be fully turbulent on all aircraft components, otherwise, set IT equal to the transition pattern number defined by the card 32 input.

41	ITRM(1)	Trim indicator, 1st survey
42	ITRM(2)	Trim indicator, 2nd survey
:	:	:
60	ITRM(20)	Trim indicator, 20th survey

For the ith survey condition, set ITRM=0 or blank if the horizontal tail is fixed, set ITRM=1 if the horizontal tail will vary to trim out the moment.

Card 31 (NSURV > 0, all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	FMSURV(I)	Mach number for the survey condition
11-20	ALT(I)	Altitude for the survey condition. Reynolds number can be input here instead of altitude by inputting a negative value of RN/ft multiplied by 10^6
21-30	DHSV(I)	Horizontal tail setting (If ITRM(I)=1, the program will calculate the tail setting)
31-40	SWPV(I)	Leading-edge sweep angle (deg.) (Required input only for variable- sweep configurations)
41-50	CLLO(I)	Low C_L value for the survey condition
51-60	CLHI(I)	High C_L value for the survey condition

Repeat card 31 for each survey condition, I=1, NSURV.

Card 32 (IT#0, 6F10.0 Format)

Each survey has a transition indicator assigned to it. If IT#0 for any survey, a boundary-layer transition pattern must be defined for each component of the aircraft. Up to five different transition patterns can be used in the surveys. Each transition pattern is numbered sequentially by the order it is input beginning with J=1 for the first transition pattern.

Transition location on a component is indicated by inputting the fraction of the component length where transition occurs.

TRB(J,K1)	Transition location on body K1 (K1 = 1, NBODYS)
TRN(J,K2)	Transition location on nacelle K2 (K2 = 1, NNACS)

Card 32 (Continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
	TRU(J,K3)	Transition location on upper surface of panel K3 (K3 = 1, NPNLS)
	TRL(J,K3)	Transition location on lower surface of panel K3
	TRS(J,K4)	Transition location on upper and lower surface of airfoil surface K4 (K4 = 2, NSURFS)

For each transition pattern J, first input the body transition location followed in turn by all the nacelle, wing panel, and airfoil surface transition locations. Repeat Card 32 input for each transition pattern J=1, to J_{MAX} where J_{MAX} = maximum value of IT(I=1, 20).

Card 33 (NHLSV > 0, all Type F)

1-10	DFI(1,I)	Flap deflection (deg.) of first flap element
11-20	CPF(I)	Wing chord with flap extended to wing chord with flap retracted ratio
21-30	SLI(I)	Slat deflection (deg.)
31-40	CPS(I)	Wing chord with slat extended to clean wing chord ratio
41-50	DELCD(I)	Landing gear drag
51-60	H(I)	Height of wing $\bar{c}/4$ above ground (ft.)

"PLAIN", "S.S.", or "INPUT" trailing-edge-flap types are single-element flaps. If FLAP = "D.S." or "T.S.", the deflection of each flap element must be input beginning with DFI(1,I) in Columns 1-10, DFI(2,I) in Columns 11-20 and for "T.S." flaps DFI(3,I) in Columns 21-30. The data for CPF(I), CPS(I), etc. are then shifted to the right on the card.

Repeat card 33 for I=1, to NHLSV.

Card 34 - Format Input Control Card - Write "ADJUST" beginning in Column 1 to indicate if adjustment factors are to be applied to some of the aerodynamic parameters predicted by the program. If no adjustment factors are to be read in, write "END OF INPUT" beginning in Column 1 to indicate that all Configuration Definition and Problem Control Cards have been read in and go to card 41.

Cards 35 through 39 - Adjustment Option Cards - The adjustment option allows certain predicted items in the computer procedure to be adjusted to match a desired value. Thus, the predictions can be adjusted to match wind tunnel data, for instance, so that perturbations in geometry for trade studies can be predicted from a firm baseline. An aerodynamic parameter of interest (A_{PRED}) can be adjusted to match an experimental value (A_{EXP}) by the equation

$$A_{EXP} = (A_{PRED}) \cdot Y_M + Y_A$$

where Y_M and Y_A are input correlation multiplier and adder that are a function of Mach number or lift coefficient.

Card 35 (Type I) Set a given IVAL indicator equal to a non-zero value to identify it as being an aerodynamic parameter to be adjusted. Begin with IVAL=1 for the first parameter to be identified, IVAL=2 for the second parameter, etc.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1	IVAL(1)	Adjust C_{M_0} as a function of Mach number
2	IVAL(2)	Adjust C_{DMISC} as a function of Mach number
3	IVAL(3)	Adjust α_{LO} as a function of Mach number
4	IVAL(4)	Adjust M_{CR} as a function of lift coefficient
24-25	NXVAR	Number of Mach values in the table of Mach function adjust factors (≤ 15)
29-30	NADJ	Number of parameters to be adjusted as a function of Mach number
34-35	NXCL	Number of C_L values in the table of lift function adjust factors (≤ 15)
39-40	NADJ2	Set equal to 1 if M_{CR} is to be adjusted

Card 36 (NXVAR > 0, all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	X(1)	Mach numbers for the table of Mach function adjust factors
11-20	X(2)	
21-30	X(3)	
31-40	X(4)	
41-50	X(5)	
51-60	X(6)	

Repeat card 36 until NXVAR values of X are read in.

Card 37 (NXVAR > 0, all Type F)

1-10	YM(J=1,I=1)	Multiplier factor
11-20	YA(J=1,I=1)	Adder factor
21-30	YM(J=2,I=1)	
31-40	YA(J=2,I=1)	
41-50	YM(J=3,I=1)	
51-60	YA(J=3,I=1)	

Repeat card 37 until NXVAR values are read in (J=NXVAR) for each aerodynamic parameter to be adjusted (I=1 to NADJ).

Card 38 (NXCL > 0, all Type F)

1-10	XCL(1)	C _L values for the table of lift function adjust factors
11-20	XCL(2)	
21-30	XCL(3)	
31-40	XCL(4)	
41-50	XCL(5)	
51-60	XCL(6)	

Repeat card 38 until NXCL values of XCL are read in.

Card 39 (NXCL > 0, all Type F)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	YM(J=1,I=NXVAR+1)	Multiplier factor
11-20	YA(J=1,I=NXVAR+1)	Adder factor
21-30	YM(J=2,I=NXVAR+1)	
31-40	YA(J=2,I=NXVAR+1)	
41-50	YM(J=3,I=NXVAR+1)	
51-60	YA(J=3,I=NXVAR+1)	

Repeat card 39 until NXCL values are read in (J=NXCL).

Card 40 - Format Input Control Card - Write "END OF INPUT" beginning in Column 1 to indicate that all Configuration Definition, Problem Control and Adjust Cards have been read in.

Card 41 - Repeat Problem Control Card - Write "END" to terminate a problem. This clears the input common blocks and reads card 1 as the next card. Write "SAVE" to repeat a problem with only small changes to the geometry or conditions to be run. Additional values will be read in on cards 42 through 50 which will update the previous problem; values not read in will remain unchanged.

Card 42 - New Title - Enter any alphanumeric characters to identify each problem. Columns 1-66.

Card 43 (Required, all Type I)

1	L(1)	BODYS Namelist data to be read in
2	L(2)	NACEL Namelist data to be read in
3	L(3)	WING Namelist data to be read in
4	L(4)	SURFS Namelist data to be read in
5	L(5)	SURV Namelist data to be read in
6	L(6)	STOL Namelist data to be read in
7	L(7)	ADJUST Namelist data to be read in

Set L(1), L(2), ... L(7) equal to 1 if that particular set of Namelist data is to be read in. Only the values that are read in will change the values in storage from the previous problem.

Use of NAMELIST in Input Data

Input data must be in a special form in order to read using a NAMELIST list. The first character (column 1) in each card to be read must be blank. The second character in the first card of a group of data cards must be an \$ (dollar sign), immediately followed by the NAMELIST name. The NAMELIST name must be followed by two blanks and must not contain any embedded blanks. This name is followed by data items separated by commas. (A comma after the last item is optional.) The end of a data group is signaled by \$. The form of the data items in an input card may be

(1) Variable name = constant

The variable name may be a subscripted array name or a single variable name. The constant may be integer, real, literal, complex, or logical.

(2) Array name = set of constants (separated by commas)

The array name is not subscripted. The set of constants consist of constants of the type integer, real, literal, complex, or logical. The number of constants must be less than or equal to the number of elements in the array. Successive occurrences of the same constant can be represented in the form K * constant.

The variable names and array names specified in the input data set must appear in the NAMELIST list, but the order is not significant.

Each data card must begin with a complete variable or array name or constant. Embedded blanks are not permitted in names or constants. Trailing blanks after integers or exponents are treated as zeros.

NOTE: All data items on a card are entered in Columns 9-66.

Card 44 - BODY'S NAMELIST - Namelist input variables

NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPMLS, SREF, CMAC,
IREF, ROUGHK, XCG, ZCG, CONCL, BLEN(I), BWID(I), BHGT(I),
BAWET(I), BQ(I), BNO(I), BAMX(I), BABS(I), BLNS(I), BLBT(I),
BASE(I), BFUS, AB, FMISC

Card 45 - NACEL NAMELIST

NNACS, SREF, ELEN(I), EWID(I), EHGT(I), EAWET(I), EQF(I),
ENO(I), EAMX(I), EIN(I), EXIT(I), ELNS(I), ELBT(I)

Card 46 - WING NAMELIST **

NSURFS, ISWP, NPNLS, AR, TAPR, SWPLE, SPLAN, TWIST, WING,
TW, CAM(I), TOC(I), XLE(I), YW(I), CRW(I), CBAR(I), AWET(I),
SWMT, XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC, AFTAW,
SREF, CMAC, CLE(I), YC(I), CCR(I), XLEW, YWW, YB, XCG,
CONCL

Card 47 - SURFS NAMELIST **

NSURFS, NHT, NVT, SREF, SBAR(I), TS(I), SCAM(I), STOC(I),
SAWET(I), SMTSW(I), SHF(I), SWL(I), SWT(I), STAPR(I), SCR(I),
HTLE, HTY, HTZ, HTINC

Card 48 - SURV NAMELIST

NSURV, NHLSV, NCLAS, IT(I), ITRM(I), FMSURV(I), ALT(I),
DHSV(I), SWPV(I), CLLO(I), CLHI(I), DFI(I), CPF(I), DSI(I),
CPS(I), DELCD(I), H(I), KPRINT(I), TRB(I), TRN(I), TRU(I),
TRL(I), TRS(I), DFI2(I), CPF2(I), DSI2(I), CPS2(I)

Card 49 - STOL NAMELIST ***

IHLS, NF, NS, FLAP, SLAT, DF(I), CFDC(I), DCLOF(I), DCLMF(I),
DCDF(I), DCMOF(I), DS(I), CSOC(I), DCLOS(I), DCLMS(I),
DCOS(I), DCMOS(I), BF1I, BF10, CF1, BF2I, BF20, CF2, BS1I,
BS10, CS1, BS2I

Card 50 - ADJUST NAMELIST

IVAL(I), NXVAR, NXCL, X(I), XCL(I), YM(J,I), YA(J,I)

** TW and TS indicate type of airfoil section; the following code
is used for NAMELIST input.

CODE	AIRFOIL TYPE	CODE	AIRFOIL TYPE
1	63-0XX	10	00XX-62
2	64-0XX	11	00XX-63
3	65-0XX	12	00XX-64
4	66-0XX	13	00XX-65
5	63A0XX	14	00XX-66
6	64A0XX	15	00XX-33
7	65A0XX	16	00XX-34
8	Supercritical	17	00XX-35
9	Biconvex	18	00XX-93
		19	00XX-94
		20	00XX-95

*** FLAT and SLAT indicate type of high lift system; the following code is used for NAMELIST input.

CODE	FLAP TYPE	CODE	SLAT TYPE
0	No flap	0	No slat
1	Plain flaps	1	Leading-edge flap
2	Single-slotted flap	2	Slotted slat
3	Double-slotted flap	3	Kruger slat
4	Triple-slotted flap	4	Input section data
5	Input section data		

After all the NAMELIST data is read in, card 41 is the next card to be read in.

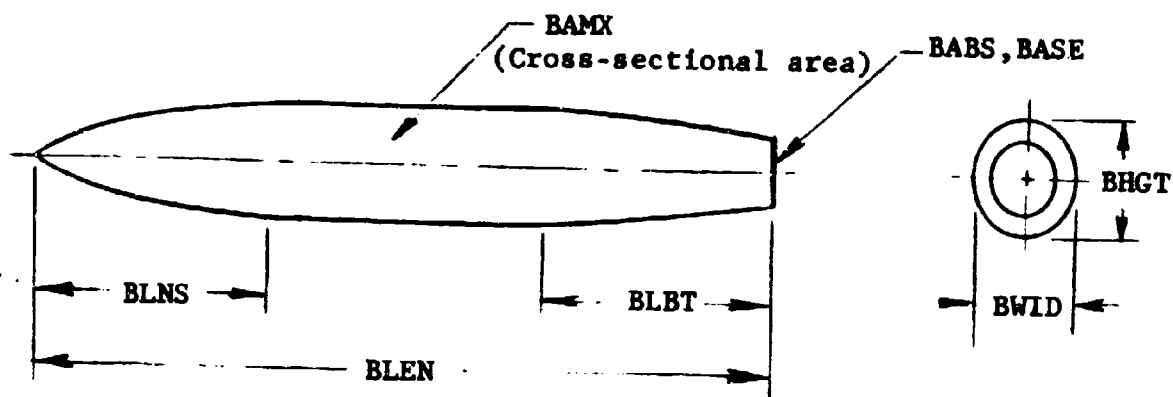


Figure 2 BODIES

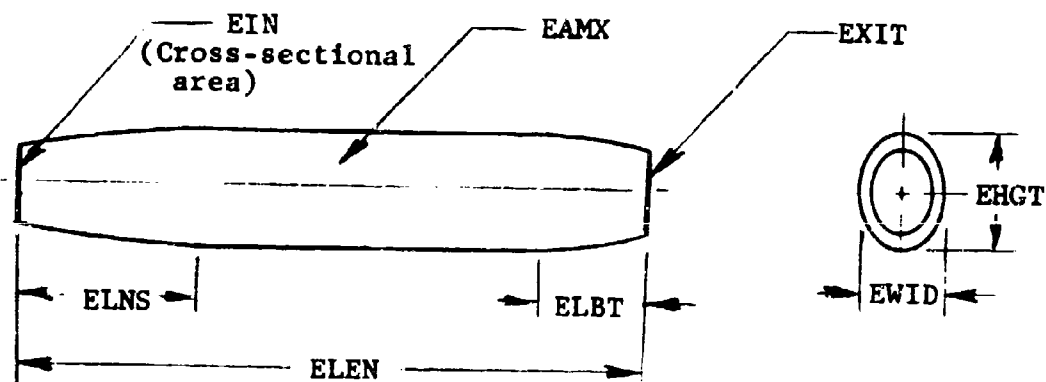


Figure 3 NACELLES

$$\text{STAPR} = \text{SCT}/\text{SCR}$$

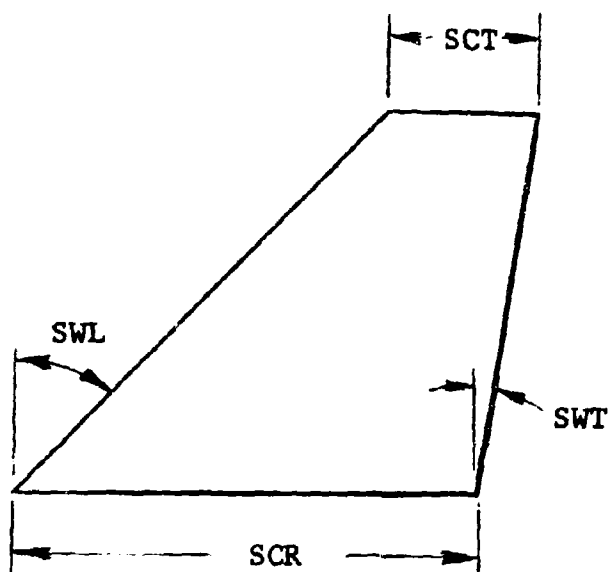
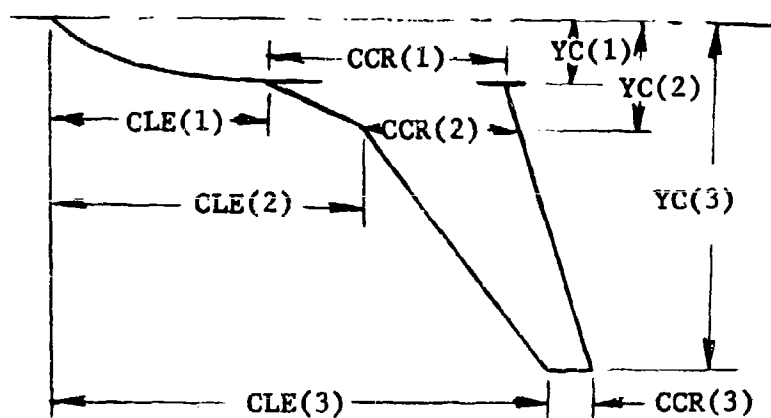
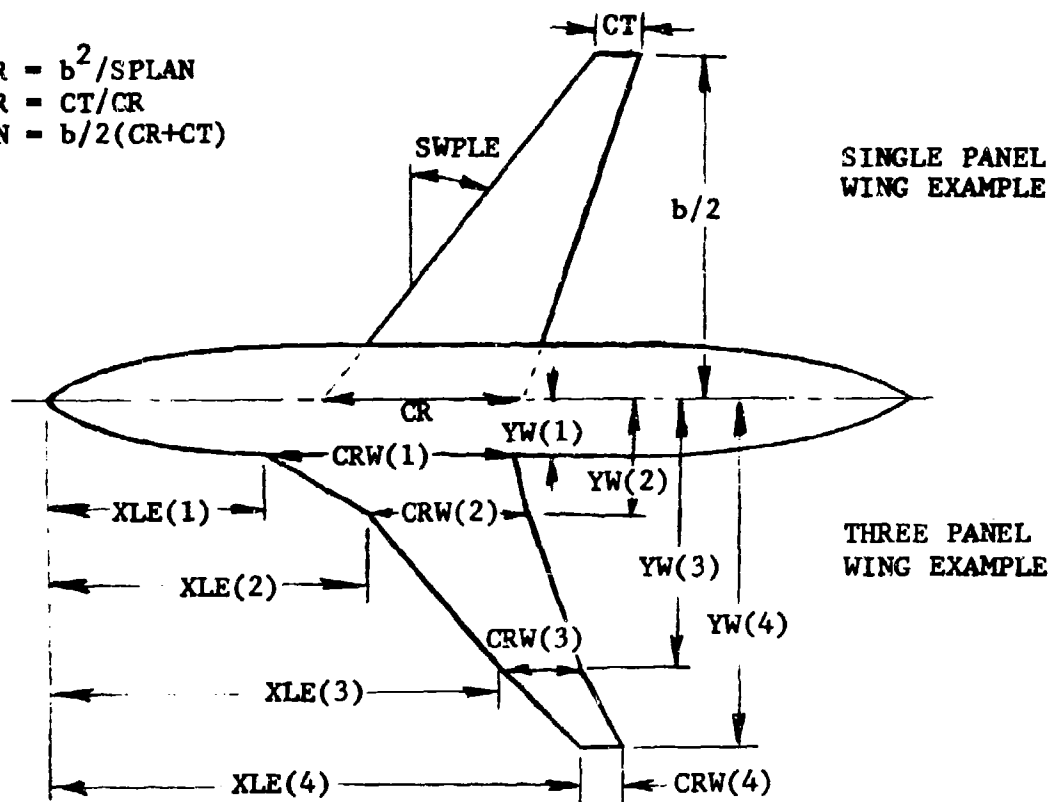


Figure 4 AIRFOIL SURFACE GEOMETRY
(OTHER THAN MAIN WING)

$$\begin{aligned} AR &= b^2 / \text{SPLAN} \\ \text{TAPR} &= CT / CR \\ \text{SPLAN} &= b/2 (CR + CT) \end{aligned}$$



WHEN MORE THAN TWO PANELS ARE USED TO DEFINE THE WING AN EQUIVALENT TWO-PANEL WING MUST ALSO BE DEFINED FOR MOMENT CALCULATIONS

Figure 5 WING SURFACE GEOMETRY

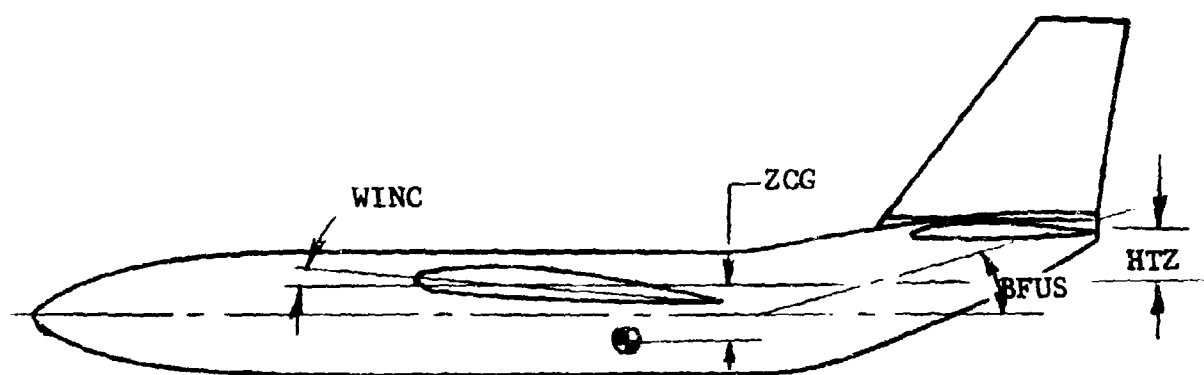
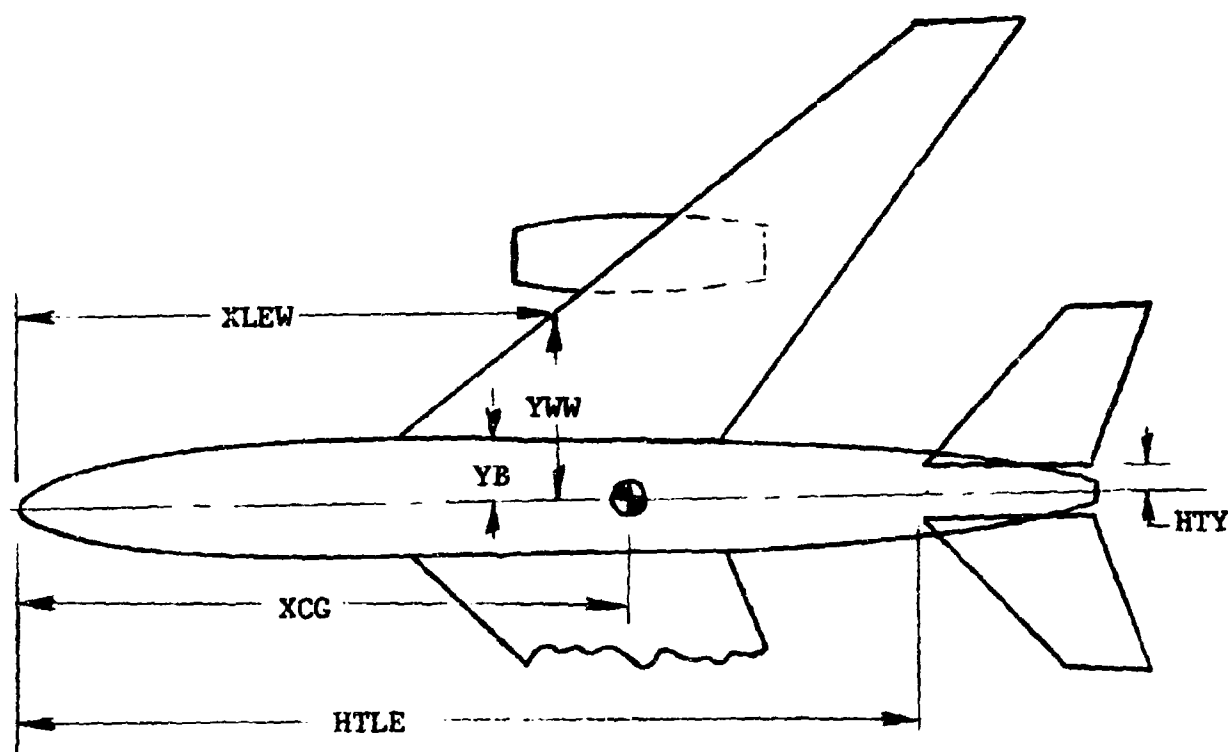


Figure 6 AIRCRAFT GEOMETRY REPRESENTATION

$BF1I = y_1/b/2$
 $BF1O = y_2/b/2$
 $BS1I = y_5/b/2$
 $BS1O = y_6/b/2$
 $CFI = C_f/C$
 $CFOC = C_f^f/C$

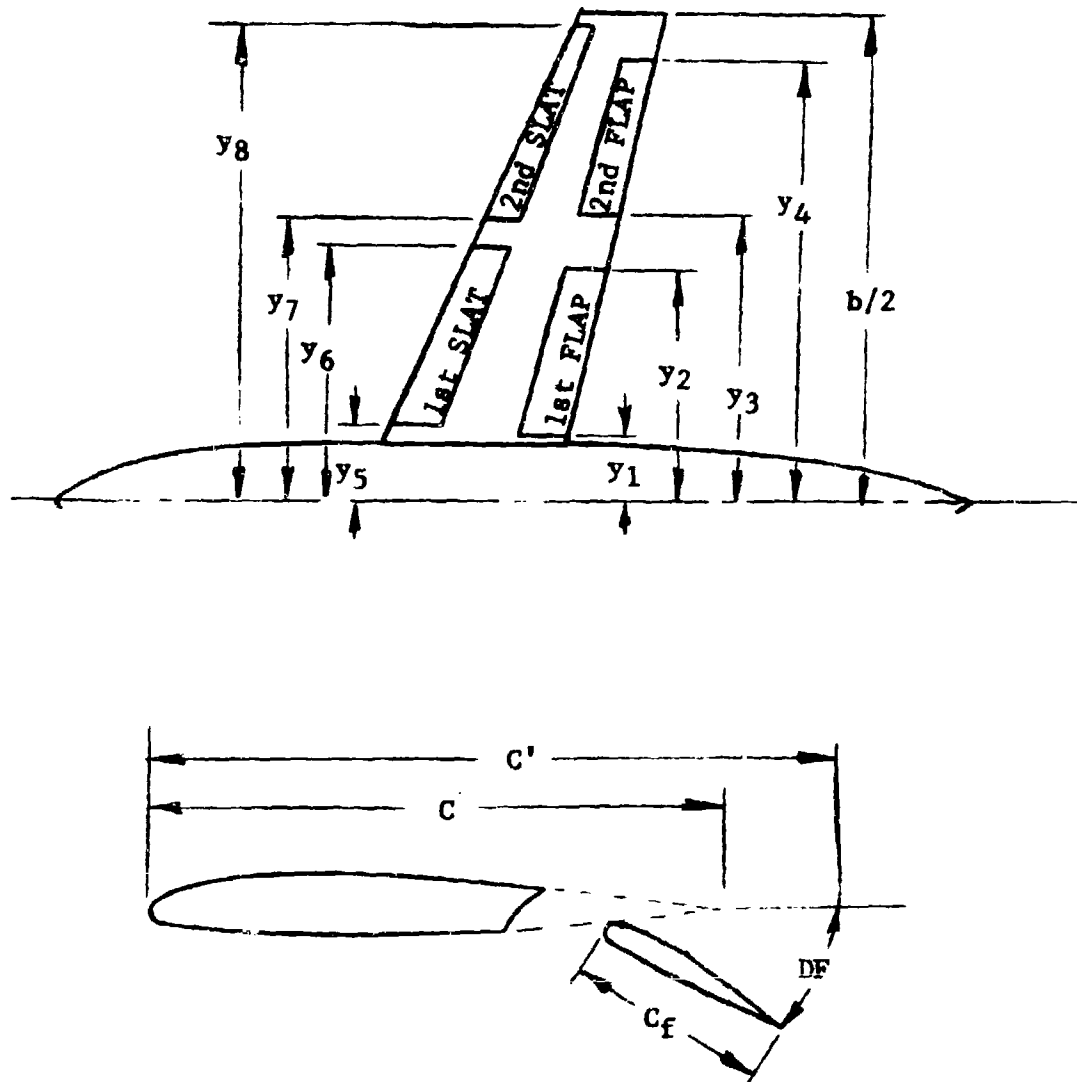


Figure 7 HIGH LIFT SYSTEM GEOMETRY

3. SAMPLE PROBLEMS

Three configurations are analyzed to illustrate the Large Aircraft input techniques and output formats. Problem 1 considers a transport configuration and demonstrates the Format input option and the high-speed lift and drag calculations. Problem 2 demonstrates the input to define the high-lift system for a transport configuration, some dump options and the high-lift system calculations in both free air and in ground effect. Problem 2 also demonstrates the use of the Repeat Problem Control Card to rerun the problem changing the defined configuration from a full-span flap to a partial-span flap high-lift system. Problem 3 demonstrates the Namelist input option for a wing-body configuration.

An abbreviated sample-problem output listing for Problem 1 is presented in pages 35 through 42 corresponding to the configuration defined by the input data in Table 2. The first page of output prints the problem title and the printout option indicators that were set. The next set of output is a listing of some of the input data which occurs when the FORMAT input option is used. The next set of output is a dump of the geometry calculated in Program GEOM which occurs when KPRINT(12) is set equal to 1. A summary of the configuration geometry used in friction drag calculations follows along with a table of the prediction for the critical Mach number versus lift and fuselage aft-end upsweep drag versus wing angle of attack. The last set of data consist of either a tabulation of the predicted drag, moment and angle of attack versus lift for a fixed horizontal tail setting, or a tabulation of drag, tail setting, and angle of attack versus lift for a trimmed ($C_M=0$) condition.

Several columns of drag versus lift are shown tabulated in the last set of data so that a breakdown of the items contributing to the total drag buildup can be seen. Basic drag due to lift of the wing-body combination is shown in one column, transonic-rise-plus-fuselage aft-end upsweep drag (if any) is shown in another column, while the lift and drag for a zero-horizontal-tail setting are shown in the last two columns on the page. A listing of the items contributing to minimum drag is also shown in the output, along with values of the lift-curve slope, zero-lift angle of attack, polar shape factors, tail-off zero-lift moment, tail-off moment curve slope, separation lift coefficient and maximum lift coefficient.

The input data for sample-problem 2 is shown in Table 3. The sample-problem output listing for problem 2 is presented in pages 44 through 51. Problem 2 set many of the printout option indicators to dump data generated in various subroutines. The dumped data is useful for checking out the program or evaluating the output in detail. The definition of the parameters that are dumped can be determined from the listing of the subroutine. For the high-lift survey problems the final output consists of a table of α , C_L , C_D , C_M for both free-air and in-ground effect up to the maximum lift coefficient.

Namelist input data for sample-problem 3 is shown in Table 4 and the output listing is presented in pages 53 through 56.

Table 2 SAMPLE PROBLEM 1

C-141A FLIGHT TEST ANALYSIS				NASA CR - 1558					
1								P010001	
FORMAT INPUT								P010002	
3	1	5	1	1	0	2	0	1	P010003
3228.0	266.47		732.88		0.0		0.0	5.00	P010004
1590.0	170.0		170.0		4347.52				P010005
22600.0	0.0		204.0		610.0		0.0	7.0	P010006
404.0	70.14		70.14		822.0		1.2	2.0	P010007
3860.0	0.0		202.0		202.0		0.0		P010008
299.84	43.8		43.8		136.8		1.3	1.0	P010009
1508.2	0.0		150.0		145.0		0.0		P010010
199.22	66.0		66.0		1045.96		1.3	4.0	P010011
3421.2	1800.0		1344.96		55.0		100.0		P010012
3228.0	0.373								P010013
0011-63	-5.58		4.89		25.0				P010014
0.4	0.1195		450.0		48.0		380.04	310.3	P010015
0.4	0.11		641.6		404.6		240.7	186.3	P010016
928.03	959.7		131.89					3134.2	P010017
64A010.5	0.0		0.105		25.0		1.1	2932.2	P010018
29.0	11.4		0.35		171.15		115.5	893.74	P010019
1512.0	0.0		276.0		0.0				P010020
64A013	0.0		0.13		35.0		1.1		P010021
38.5	22.5		0.60863		273.17		219.7	819.4	P010022
64A010	0.0		0.1		73.0		1.3		P010023
73.0	72.0		0.99		200.0		230.0	223.24	P010024
64A010	0.0		0.1		73.0		1.3		P010025
73.0	72.0		0.99		200.0		200.0	214.88	P010026
12	0	21							P010027
0.6	-2.47683	0.0			28.0		0.0	1.0	P010028
0.7	-2.47683	0.0			28.0		0.0	1.0	P010029
0.75	-2.47683	0.0			28.0		0.0	1.0	P010030
0.775	-2.47683	0.0			28.0		0.0	1.0	P010031
0.8	-2.47683	0.0			28.0		0.0	1.0	P010032
0.81	-2.47683	0.0			28.0		0.0	1.0	P010033
0.6	-2.47683				28.0		0.0	1.0	P010034
0.7	-2.47683				28.0		0.0	1.0	P010035
0.75	-2.47683				28.0		0.0	1.0	P010036
0.775	-2.47683				28.0		0.0	1.0	P010037
0.8	-2.47683				28.0		0.0	1.0	P010038
0.81	-2.47683				28.0		0.0	1.0	P010039
END OF INPUT									P010040
END									P010041
									P010042

GENERAL DYNAMICS
6607 PROCEEDURE #17

CONVAIR AEROSPACE DIVISION
PROBLEM 185829--1

FORT WORTH OPERATION
11/18/74 PAGE 0011

EMPIRICALLY BASED COMPUTER PROGRAM
TO PREDICT THE AERODYNAMIC CHARACTERISTICS
OF LARGE AIRCRAFT

KFEINT(12) = 1

C-141A FLIGHT TEST ANALYSIS NASA CR - 1558

GENERAL DIMENSIONS 6600 PROSECUTORY

CONVAID AEROSPACE DIVISION
PROBLEM IN 1589-11

FORT WORTH OPERATION
9/10/74 PAGE 002

PROBLEM INPUT PARAMETERS

NPOLYS = 7 NMCS = 1 NSURFS = 5 NMT = 1 NVT = 1 ISNP = 0 NPMLS = 2
SREF = 3228.000 SQ.FT. CMAC = 266.473 IN. FUS. STA. C.G. = 737.88 IN. ZCG = 0.000 IN.

NO.	GLEN	PMID	BHGT	BAWET	RAMX	BABS	BLNS	BLBT	BASE	BO	BNO
1	1590.20F	177.10F	175.00F	4347.52F	22677.000	0.00F	204.00F	618.00F	0.00F	1.00F	1.000
2	004.00F	78.14F	78.14F	822.00F	3863.000	0.00F	212.00F	202.00F	0.00F	1.00F	2.000
3	299.84F	43.84F	43.84F	136.81F	1595.300	0.00F	152.00F	145.00F	0.00F	1.00F	1.000
NO.	ELEN	EMID	ELGT	EAWET	CAHX	E1N	EXIT	ELNS	ELBT	EOF	ENO
1	199.22F	66.84F	66.84F	9045.94F	3421.200	1000.00F	1364.95F	99.00F	177.00F	1.30F	4.100

AIRFOIL TYPE = N11-53

TWIST = -5.581 DEG.

INCIDENCE = 4.89F DEG.

MAX. THICKNESS SMP. = 25.11F DEG.

WING PANEL GEOMETRY

NO.	CAMEFO	T/C	XLE	YH	CRM	GBAR	AWET
1	.40F	.110	451.000	48.00F	347.040	315.200	3134.21F
2	.40F	.110	641.640	404.60F	247.730	186.300	2932.20F
			928.030	959.77F	131.890		

GEOMETRY FOR ADDITIONAL AIRFOIL SURFACES

AIRFOIL	CAMEFO	T/C	SMTSH	SHP	L.E.SMP	T.E.SMP	TAPER	CR	SBAR	SAWET
64AF10	.600F	.105	25.000	1.10F	29.000	11.40F	.750	171.151	115.50F	897.74F
64AF13	.600F	.130	35.000	1.10F	36.50F	22.533	.609	273.17F	219.70F	819.41F
64AF16	.600F	.110	73.000	1.30F	73.000	72.03F	.990	28.000	20.000	225.24F
		.110	73.000	1.30F	73.000	72.03F	.990	20.000	20.000	214.88F

12 DRAG POLARS TO BE GENERATED AT THE FOLLOWING CONDITIONS
MACH NO. ALTITUDE TAIL SETTING L.E.SHEEP FROM CL TO CL

MACH NO.	ALTITUDE	TAIL SETTING	L.E.SHEEP	FROM CL	TO CL
.600F	-2.477	0.00F	29.000	0.00F	1.000
.700F	-2.477	0.00F	28.000	0.00F	1.000
.750F	-2.477	0.00F	28.000	0.00F	1.000
.775F	-2.477	0.00F	28.000	0.00F	1.000
.800F	-2.477	0.00F	28.000	0.00F	1.000
.810F	-2.477	0.00F	28.000	0.00F	1.000
.800F	-2.477	TRIMMED	28.000	0.00F	1.000
.700F	-2.477	TRIMMED	28.000	0.00F	1.000
.750F	-2.477	TRIMMED	28.000	0.00F	1.000
.775F	-2.477	TRIMMED	28.000	0.00F	1.000
.800F	-2.477	TRIMMED	28.000	0.00F	1.000
.810F	-2.477	TRIMMED	28.000	0.00F	1.000

GENERAL DYNAMICS
6600 FPOCELP 21T

COMVAIR AFOSPACE DIVISION
PROBLEM JAG929-01

FORT WORTH OPERATION
01/10/74 PAGE 003

FAC OF INPUT

GENERAL DYNAMICS
6807 FCGCCLCCT 217

CONVAIR AEROSPACE DIVISION
PROGRAM 165929-01

FORT WORTH OPERATION
01/17/74 PAGE 0005

CONFIGURATION SUMMARY

PCDY NUMBER	LENGTH (FT.)	WETTED AREA (SQ. FT.)	FR OR T/C	INTERFERENCE FACTOR	MAX. T/C SWEEP (DEG.)
PCDY NUMBER 1	132.5000	4747.5200	9.35294	1.0000	
PCDY NUMBER 2	73.6667	822.0000	5.75991	1.2000	
PCDY NUMBER 3	24.9000	136.0000	6.84566	1.3000	
NACELLE NO. 4	16.60167	1045.9500	3.01848	1.30000	
WING PANEL NO. 1	25.4500	2134.2000	.11950		25.0000
WING PANEL NO. 2	15.5250	2032.2000	.11900		25.0000
AIRFOIL SURFACE NO. 2	9.6250	893.7400	.10500	1.1000	35.0000
AIRFOIL SURFACE NO. 3	18.3000	819.4000	.13000	1.1000	73.0000
AIRFOIL SURFACE NO. 4	16.6667	223.2400	.10000	1.3000	73.0000
AIRFOIL SURFACE NO. 5	16.6667	214.8000	.10000	1.3000	73.0000

4-DIGIT AIRFOIL SECTION

REPORT OF RESULTS CONVOLUTIONAL ANALYSIS FOR 1/10/71 PREPARATION

C-141A FLIGHT TEST ANALYSIS
MACH NO. = .700
WING AREA = 2.4766276

L.E. SWEEP ANGLE = 0.0 DEG.
TAIL DEFL. (DH) = .19 DEG.

CL	TOTAL CD	CM	ALPHA	CD LIFT	CD RAFT	GL AT DH=1	CD AT DH=1
.000	.1176	.0670	-5.05	.0125	.0194	.140	.4176
.050	.1168	.0672	-4.61	.0112	.0178	.140	.4168
.100	.1156	.0674	-4.18	.0095	.0164	.140	.4156
.150	.1141	.0676	-3.74	.0073	.0150	.140	.4141
.200	.1123	.0678	-3.31	.0049	.0137	.140	.4123
.250	.1102	.0679	-2.87	.0025	.0125	.140	.4102
.300	.1078	.0680	-2.43	.0000	.0114	.140	.4078
.350	.1051	.0681	-2.00	.0000	.0104	.140	.4051
.400	.1021	.0682	-1.56	.0000	.0094	.140	.4021
.450	.0988	.0683	-1.12	.0000	.0085	.140	.3988
.500	.0952	.0684	-.69	.0000	.0077	.140	.3952
.550	.0913	.0685	-.25	.0000	.0069	.140	.3913
.600	.0871	.0686	.18	.0000	.0061	.140	.3871
.650	.0826	.0687	.62	.0000	.0055	.140	.3826
.700	.0778	.0688	1.06	.0000	.0048	.140	.3778
.750	.0727	.0689	1.49	.0000	.0042	.140	.3727
.800	.0673	.0690	1.93	.0000	.0037	.140	.3673
.850	.0616	.0691	2.37	.0000	.0032	.140	.3616
.900	.0557	.0692	2.80	.0000	.0028	.140	.3557
.950	.0495	.0693	3.24	.0000	.0024	.140	.3495
1.000	.0430	.0694	3.68	.0000	.0021	.140	.3430

CL = .11440 DEG.
K = .02106
CM = -.02042

CLDB = 1.29336
CLMAX = 1.28090

DRAG BREAKDOWN	FUSLAGE	BODIES	NACELLES	WING	HORIZ TAIL	VEPT TAIL	SURFACES
FRICITION	.0023	.0002	.0000	.0000	.0000	.0000	.0000
FORM	.0023	.0002	.0000	.0000	.0000	.0000	.0000
INTERF	.0023	.0002	.0000	.0000	.0000	.0000	.0000
WAVE	.0023	.0002	.0000	.0000	.0000	.0000	.0000
PAGE	.0023	.0002	.0000	.0000	.0000	.0000	.0000
CAPER	.0023	.0002	.0000	.0000	.0000	.0000	.0000
TRAG RISE	.0023	.0002	.0000	.0000	.0000	.0000	.0000
MISC	.0023	.0002	.0000	.0000	.0000	.0000	.0000

COMIN = .0144

Reproduced from
best available copy.

GENERAL CHARACTERISTICS

CONVERSION FACTORS DIVISION

1991-1992 PREPARATION

C-141A FLIGHT TEST ANALYSIS NASA CR - 1584
MACH NO. = .711 RA/FT = 2.4764806

L.E. SWEET ANGLE = 0.0 DEG.

TRIMMED CONDITION

CL	TOTAL CD	DM	ALPHA	CD LIFT	CD RAFT	CL AT DM=0	CU AT DM=0
.021	.11646	1.47892	-5.05	.30025	.00194	.00100	.01716
.065	.11629	1.47874	-4.61	.30072	.00178	.00000	.01660
.116	.11623	1.45997	-4.10	.30035	.00164	.00000	.01656
.164	.11649	.99966	-3.74	.30034	.00132	.00000	.01671
.212	.11609	.83482	-3.31	.30089	.00137	.00000	.01713
.259	.11775	.67748	-2.87	.30171	.00125	.00000	.01782
.307	.11878	.51565	-2.43	.30277	.00114	.00000	.01870
.355	.12011	.35737	-2.00	.30411	.00104	.00000	.02001
.403	.12151	.19565	-1.56	.30569	.00094	.00000	.02150
.451	.12326	.02752	-1.12	.30753	.00085	.00000	.02326
.498	.12525	.13401	-.69	.30964	.00077	.00000	.02528
.546	.12749	-.29990	-.25	.31200	.00069	.00000	.02756
.594	.12997	.44414	.18	.31463	.00061	.00000	.03012
.641	.13270	.62870	.62	.31751	.00055	.00000	.03293
.689	.13557	.79757	1.16	.32065	.00048	.00000	.03601
.737	.13855	.95873	1.49	.32400	.00042	.00000	.03936
.784	.14234	-1.12815	1.93	.32772	.00037	.00000	.04297
.832	.14647	-1.20981	2.37	.33164	.00068	.00000	.04720
.881	.15163	-1.45570	2.80	.33582	.00192	.00000	.05261
.927	.15808	-1.62178	3.24	.34026	.00412	.00000	.05925
.975	.16562	-1.78006	3.68	.34496	.00727	.00000	.06710

CLOB = 1.29036
CLMAX = 1.28090

ALC = -5.05055 DEG.
DELCL = -.6901
CM/CL = -.1154

CL = .11648 PER DEG.
K = .05106
CM = -.07012

DRAG BREAKDOWN	FUSLAGE	BODIES	NACELLES	WING	HORIZ TAIL	VERT TAIL	SURFACES
FRICTION = .01050	.00232	.00062	.00075	.00422	.00069	.00050	.00000
FORM = .00170	.00000	.00019	.00009	.00098	.00012	.00013	.00005
INTERF = .00128	.00000	.00017	.00025	.00105	.00023	.00019	.00013
WAVE = .00000	.00000	.00000	.00000	.00000	.00000	.00000	.00000
PAGE = .00000	.00000	.00000	.00000	.00000	.00000	.00000	.00000
CAPPER = .00000	.00000	.00000	.00000	.00000	.00000	.00000	.00000
CPAC RISER = .00000	.00000	.00000	.00000	.00000	.00000	.00000	.00000
MISC = .00067	.00000	.00000	.00000	.00000	.00000	.00000	.00000

CDMIN = .01607

Table 3 SAMPLE PROBLEM 2

NACA TN D-7034 4-ENGINE STOL TRANSPORT (HIGH WING)

1 1111 1111 1111

FORMAT INPUT

1	0	3	1	1	0	1	1
5.143	11.28	27.09	-3.73				
68.268	10.404	9.504					
	0.0	12.15	25.2	0.0			
7.0	0.3	29.0	5.143	-5.05	2.85		
66-012	0.0	0.12	24.27	14.7692	4.75		
25.0							
66-012	0.0	0.12	7.5	1.0			
12.5	-7.5	0.44	9.816				
58.255	0.0	17.67	0.0				
66-012	0.0	0.12	22.0	1.0			
27.0	6.0	0.52	14.628				
D.S. FLAP		SLAT					
0.134	1.0	0.16	0.134	1.0	0.15		
0.38							
0	1						
30.0	30.0	1.16	19.0	1.1			
1.14							

END OF INPUT

SAVE

NACA TN D-7034 PARTIAL SPAN FLAP

1

\$STOL BF10=0.75\$

END OF PROBLEM

005573P060001
005573P060002
005573P060003
005573P060004
005573P060005
005573P060006
005573P060007
005573P060008
005573P060009
005573P060010
005573P060011
005573P060012
005573P060013
005573P060014
005573P060015
005573P060016
005573P060017
005573P060018
005573P060019
005573P060020
005573P060021
005573P060022
005573P060023
005573P060024
005573P060025
005573P060026
005573P060027

CC = 00027

GENERAL DYNAMICS
666C PROCEDURE #17

CONVAIR AEROSPACE DIVISION
PROBLEM 805573-06

FORT WORTH OPERATION
10/30/73 PAGE 0JG2

EMPIRICALLY BASED COMPUTER PROGRAM
TO PREDICT THE AERODYNAMIC CHARACTERISTICS
OF LARGE AIRCRAFT

NACA TN D-7634 4-ENGINE STOL TRANSPORT (HIGH WING)

KPRINT(1) = 1
KPRINT(11) = 1
KPRINT(12) = 1
KPRINT(13) = 1
KPRINT(14) = 1
KPRINT(16) = 1
KPRINT(17) = 1
KPRINT(18) = 1
KPRINT(19) = 1
KPRINT(24) = 1
KPRINT(25) = 1
KPRINT(26) = 1
KPRINT(27) = 1

GENERAL DYNAMICS 6600 PROBLEMS

CONVAIR AEROSPACE DIVISION
PROBLEM 28553-00

50913098TH PREPARATION

PROBLEM INPUT PARAMETERS

NBODY5 = 1 NNACS = 0 NSURFS = 3 NHT = 1 NVT = 1 ISHP = 0 NPMLS = 1

SREF = 5.143 SQ.FT. CHAC = 11.280 IN. FUS. STA. C.G. = 27.090 IN. ZCG = -3.730 IN.

NO.	PLEN	BMID	BMGT	BAWET	BAMX	BAES	BLNS	BLBT	BASE	EQ	BMO
1	00.268	11.414	9.504	-0.000	-0.000	0.000	12.150	25.200	3.000	1.000	1.000

ASPECT RATIO = 7.000 TAPER RATIO = .300 L.E.SWEEP = 29.000 PLANFORM AREA = 5.14 SQ.FT.

WIST = -5.50 DEG. INCIDENCE = 2.850 DEG. THICKNESS = .1200
 AIRFOIL TYPE = 66-12 CAMBER = 0.000
 WING LOCATION (X,Y) = 24.270 14.7692 FUSLAGE INTERSECTION = 4.7500
 MAX. THICKNESS SMP. = 25.000 CDAR = -0.0000 ANET = -0.000

GEOMETRY FOR ADDITIONAL AIRFOIL SURFACES

AIRFOIL	CAMBER	T/C	SMTSH	SMF	L.E.SWP	T.E.SWP	TAPER	CR	SBAR	SAWET
66-1.2	0.00	.12	7.500	1.000	22.500	-7.000	.440	9.816	-0.000	-0.000
66-1.12	0.00	.120	22.000	1.000	27.000	6.000	.520	14.628	-0.000	-0.000

END OF INPUT

FORT WORTH OPERATION
10/30/73 PAGE 0006

0.0000
3.04894
0.22678

49

GENERAL DYNAMICS
6600 PROCEDURE R1T

CONVAIR AEROSPACE DIVISION
PROBLEM 005973-06

FORT WORTH OPERATION
10/30/73 PAGE 0808

HIGH LIFT CONSTANTS

ZKJ = .014+1J
CLVC = .824+55
DCLSE = -.746497
DCLLF = 2.21147J

ZKC = 1.106450
DCLF = 3.257277
DCLMS = .694746

ZKS = .798797
DCLMF = 1.529456
UCLVS = -.030626

SAPHL = .391+67
ZKF = .16+20J
AK = 7.0000J0
DCL = 0.0000J0

ZMD = .80J72J
DCL = 1.2662J0
TPR = .39J0J0

ZKA = 1.276285
SAPLE = .506145
AK = .160792

UCOMIN = .329376
DCLF = .23149
DCL = .061742

COI = .046185
DCL = 0.0000J0

DCOS = 0.000000
DCLF = .276820

UCHLF = -.874933
UCHCS = .138419
DCLC = -.031626

GENERAL DYNAMICS
6603 PROCEDURE R17

CONVAIR AEROSPACE DIVISION
PROBLEM 005573-06

FORI NORTH OPERATION
10/30/73 PAGE 0369

HIGH LIFT SURVEY

ALPHA	LL	CD	CM	AGRO	CLG	CCG	CMG
1.00	1.95979	.44972	-.72670	.201	1.96203	.39114	-.74092
1.00	2.17493	.46795	-.77305	1.291	2.18319	.40266	-.79293
2.00	2.17970	.49790	-.80234	2.380	2.19341	.42533	-.86027
3.00	2.20452	.53279	-.82847	3.457	2.20714	.45323	-.92137
4.00	2.34138	.57490	-.84740	4.516	2.30492	.48772	-.98297
5.00	2.34328	.62364	-.96709	5.559	2.42651	.52110	-1.04481
6.00	2.43923	.68031	-1.01750	6.579	2.47188	.56594	-1.10671
7.00	2.49422	.74934	-1.06857	7.576	2.50161	.62084	-1.16047
8.00	2.47326	.88962	-1.12027	8.548	2.51139	.77427	-1.22297
9.00	2.43634	1.08936	-1.17255	9.496	2.51752	.90095	-1.29114
NARELIST INPUT COMPLETED							
GEOMETRY DATA DUMP, COMMON BLKGE1							
6.8054	0.5000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4.4023	3.1234	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.5271	1.3474	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
.5001	.4454	.2416	.1319	.1200	.0000	.0000	.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
.3100	5.1430	1.2308	2.8225	1.1909	0.0000	0.0000	0.0000
4.1472	0.9400	6.5411	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	.5061	.2416	.2198	-.1303	.5061	2.6342	0.0000
0.0000	.4454			5.1343	2.6594	3.3047	.3549

Table 4 SAMPLE PROBLEM 3

WING 1	NASA TN D-5805	Q/C SWP = 25 DEG.	
1			
NAMELIST INPUT			P030001
\$BODY	NSBODY=1, XCG=25.013, BLEN(1)=48.8, BWID(1)=4.9, BHGT(1)=4.9,		P030002
	BLNS(1)=14.0, BLBT(1)=10.0, BASE(1)=0.0\$		P030003
\$NACEL	NNACS=0\$		P030004
\$WING	SREF=1.5, SPLAN=1.5, NPNLS=1, AR=6.0, TAPR=0.4, SWPLE=28.27,		P030005
	TH=5.0, LAN(1)=0.0, TOC(1)=0.08, XLEW=20.587, YHW=2.45, Yd=2.45,		P030006
	CBAR(1)=6.38, SWMT=25.0\$		P030007
\$SURFS	NSURFS=1, NHT=1, NVT=0\$		P030008
\$SURV	NSURV=7, NHLSV=1, FMSURV(1)=0.225, 0.456, 0.665, 0.717, 0.77,		P030009
	0.822, 0.883, ALT(1)=-1.5, -2.55, 5*-3.5, DHSV(1)=7*0.0,		P030010
	SWPV(1)=7*28.27, CLLO(1)=7*0.0, CLHI(1)=7*1.0, NCLAS=21\$		P030011
\$STOL	IHLS=0\$		P030012
\$ADJUST	IVAL(1)=20*0\$		P030013
SAVE			P030014
WING 2	NASA TN D-5805	Q/C SWEEP = 35 DEG.	P030015
1 1			P030016
\$BODY	XCG=26.418\$		P030017
\$WING	SWPLE=37.663, XLEW=20.6136, SWMT=35.0\$		P030018
SAVE			P030019
WING 3	NASA TN D-5805	Q/C SWEEP = 45 DEG.	P030020
1 1			P030021
\$BODY	XCG=28.2\$		P030022
\$WING	SWPLE=47.01, XLEW=21.964, SWMT=45.0\$		P030023
SAVE			P030024
WING 4	NASA TN D-5805	Q/C SWEEP = 35 DEG. T/C = 0.06	P030025
1 1 1			P030026
\$BODY	XCG=26.418\$		P030027
\$WING	SWPLE=37.66, XLEW=20.6136, SWMT=35.0, TOC(1)=0.06\$		P030028
\$SURV	FMSURV(1)=0.292, .5, .702, .755, .807, .863, .932,		P030029
	ALT(1)=-1.8, -2.7, 5*-3.6\$		P030030
END			P030031
			P030032
			P030033

GENERAL DYNAMICS
6600 PROCEJUNE RLT

CONVAIR AEROSPACE DIVISION
PROBLEM 186280-03

FORT WORTH OPERATION
10/16/73 PAGE 0702

CONFIGURATION SUMMARY

BODY NUMBER 1 HING PANEL NO. 1	LENGTH (FT.)	WETTED AREA (SQ. FT.)	FR OR I/C	INTERFERENCE FACTOR	MAX. T/C SWEEP (DEG.)
	.0667	4.36693	9.95918	1.00000	
	.53167	2.46988	.08000		25.00001

6A SERIES AIRFOIL SECTION

GENERAL DYNAMICS
6500 PROCEURE KIT

CONVEIR AEROSPACE DIVISION
PROBLEM 180280-03.

FOAT WORTH OPERATION
10/16/73 PAGE 0003

MACH CRITICAL TABLE

CL	MACH CRITICAL
0.000	.8675
.100	.8516
.200	.8351
.300	.8165
.400	.7932
.500	.7658
.600	.7342
.700	.6986
.800	.6578
.900	.6159
1.000	.5740

GENERAL DYNAMICS
6600 PROCEDURE R1T

CONVAIR AEROSPACE DIVISION
PROBLEM 106284-03

FORT WORTH OPERATION
 14/16/73 PAGE 364

WING 1 NASA TN D-5835 Q/C SWP = 25 DEG.
MACH NO. = .225 RN/FT = 1.5500E+06

L.S. SWEEP ANGLE = 28.27 DEG. TAIL DEFL. (UM) = 4.03 DEG.

CL	TOTAL CD	CM	ALPHA	CJ LEFT	CD RISE	CL AT DM=0	LO AT DM=0
0.003	.12549	.00000	.00	9.37007	0.00000	0.00000	.02949
.051	.12475	.01231	.61	.31826	0.00000	.05000	.00075
.100	.12153	.00661	1.22	.30114	0.00000	.00000	.00000
.150	.12283	.00992	1.82	.30234	0.00000	.15000	.00000
.200	.12405	.01123	2.43	.30416	0.00000	.00000	.00000
.250	.12639	.01653	3.04	.30651	0.00000	.25000	.32455
.300	.12985	.01984	3.65	.30937	0.00000	.30000	.02659
.350	.13324	.02315	4.26	.31275	0.00000	.35000	.02985
.400	.13714	.02645	4.86	.31665	0.00000	.40000	.03324
.450	.14156	.02976	5.47	.32110	0.00000	.45000	.03714
.500	.14651	.03307	6.08	.32622	0.00000	.50000	.04156
.550	.15203	.03596	6.74	.33214	0.00000	.55000	.04651
.600	.15838	.03885	7.62	.34291	0.00000	.60000	.05263
.650	.16593	.04249	8.76	.35844	0.00000	.65000	.05938
.700	.17426	.04630	10.13	.37878	0.00000	.70000	.07893
.750	.18339	.05051	11.71	.40391	0.00000	.75000	.09926
.800	.19430	.05509	13.63	.43381	0.00000	.80000	.12439
.850	.20814	.05986	15.75	.46851	0.00000	.85000	.15430
.900	.22649	.06445	20.12	.51861	0.00000	.90000	.18910
.950	.25277	.06926	25.73	.58229	0.00000	.95000	.22849
1.000	.28185	.07400	26.59	.66136	0.00000	1.00000	.27277
							.32185

CLA = .0023 PER DEG.
K = .1048
CMD = 6.66E0
ALO = 0.0000 DEG.
DELCL = 0.0000
CM/CL = .0661

```
ORAG BREAKDOWN -----
FRICION = .01678
FORM     = .00175
INTERF   = .00195
WAVE     = 0.00000
BASE     = 0.00000
CAMBER   = 0.00000
DRAG RISE= 0.00000
MISC     = 0.00000
```

FUELAGE	MOORIES
.24924	0.00000
.00079	0.00000
.00076	0.00000
0.00000	0.00000
0.00000	0.00000

ST7732YH
0.00000
0.00000
0.00000
0.00000

WING
-00750
-00096
-00119
000006

HORIZ TAIL
W. 02000
B. 00000
D. 00000
L. 00000

VERY TALL
0-01000
0-01000
0-02000
0-31000

SUBPAGE
0-00000
0-00000
0-00000
0-00000

SUBPAGE
0-00000
0-00000
0-00000
0-00000

COMIN = .02349

4. PROGRAM AND SUBROUTINE DESCRIPTIONS

(arranged in alphabetical order)

This section contains a brief outline of the purpose and use of each program or subroutine.

Subroutine AALO

Purpose

To compute zero lift angle of attack. The zero-lift angle of attack is computed as the sum of the increments due to camber, twist and incidence.

Use

CALL AALO (SPEED)

where speed is the Mach number, the computed value of zero-lift angle is contained in COMMON BLKCOL.

Subroutine ACCR

Purpose

ACCR computes the aerodynamic center at low lift and at stall for single panel wings. The subroutine also obtains the lift curve slope of the wing. The low lift aerodynamic center is obtained through use of triple-interpolation of the data presented in Figures 4.1.4.2-22 and -27 in the DATCOM.

Use

CALL ACCR (SPEED, AR, SWPLE, SWPMC, TR, SPLAN, TOC, TW, FMCRO, XACR, CLAX, XACS)

where input is

SPEED	Mach number
AR	Aspect ratio of exposed wing
SWPLE	Leading-edge sweep
SWPMC	Mid-chord sweep
TR	Taper ratio

SPAN	Exposed planform area
TOC	Thickness ratio
TW	Type airfoil section indicator
FMCRO	Zero-lift critical Mach number for complete configuration

and output is

XACR	Low lift aerodynamic center referenced to leading-edge of exposed root chord
CLAX	Lift-curve slope of exposed planform
XACS	Aerodynamic center at stall

Subroutines called

TLNT
AER2
LNTP

Subroutine ADCL

Purpose

To compute the effect of camber on the displacement of the drag polar. For Mach numbers less than 1.0, the lift coefficient for minimum profile drag is computed; for Mach numbers greater than or equal to 1.0, the lift coefficient for minimum drag is computed.

Use

CALL ADCL (SPEED, CLOPT)

where speed is the Mach number and CLOPT is the C_L for minimum profile drag subsonically. Supersonically, CLOPT is the polar displacement C_L for C_{Dmin} .

Subroutine ADJUST

Purpose

The subroutine adjusts an aerodynamic parameter, y_1 , to a new value, y_2 , by the equation

$$y_2 = y_1 \cdot V_M + V_A$$

where V_M and V_A are correlation multiplier and adder factors determined from input. The factors V_M and V_A are a function of either Mach number or C_L .

Use

CALL ADJUST (ID, ID2, XVAR, YVAR), where

ID	Parameter identification number for Mach number cases
ID2	Parameter identification number for C_L cases
XVAR	Value of Mach number if ID greater than 0, or value of C_L if ID = 0 and ID2 greater than 0
YVAR	Input value y_1 is changed to output value y_2

Subroutines called

LNTP

Subroutine AERA

Purpose

The subroutine calculates angle of attack for a given untrimmed C_L condition. For supersonic Mach numbers the linear relation

$$\alpha = C_L / C_{L_\alpha} + \alpha_{LO}$$

is used. For subsonic conditions the angle of attack is calculated by one of three different methods depending on whether the wing is a high aspect ratio, low aspect ratio or has a cranked leading edge. For low-aspect-ratio and cranked wings the effect of vortex lift is accounted for in the angle calculations.

Use

CALL AERA (SPEED, CL, ALPHA), where

SPEED	Mach number (I)
CL	Lift coefficient (I)
ALPHA	Angle of attack (0)

Subroutines called

DLNT
LNTP

Program AERO

Purpose

This program controls the sequence of calculations required to compute the trimmed lift, drag, moment, and angle of attack for a given set of conditions. The set of conditions are contained in COMMON BLKOV3 as one Mach number, altitude, trim indicator and untrimmed lift coefficient. The trim indicator is used to determine if a fixed horizontal tail setting is specified (in which case moment is calculated) or if the tail setting is calculated to produce a zero moment. The results of the aerodynamic calculations are contained in COMMON BLKC01.

The sequence of the calling of subroutines for aerodynamic calculations is controlled by the parameter JPASS, contained in COMMON BLKOV3, which is defined by program SURVEY. The parameter JPASS is used to prevent calling certain subroutines on repeat passes through AERO if the value they calculate remains fixed. An example would be where lift is the only change in the set of conditions received from COMMON BLKOV3; in this case it is unnecessary to recompute minimum drag for each CL.

Use

CALL OVERLAY (4HOVLY,3,3)

Subroutines called

ADJUST
AERA
AERA1 (second entry point in AERA)
AFTCD
ATMOS
CDDR
CDDR1 (second entry point in CDDR)
CDL1
CDL2
CDRG
CLBRK

(AERO continued)

Subroutines called

CLWBT
CMOW
DMIN
TDRG
WBAC

Subroutine AER2

Purpose

AER2 computes the lift-curve slope for a wing surface defined by the data in COMMON BLKCLA. The lift-curve slope is computed using a modified Polhamus expression in the subsonic range which is extended to match the two-dimensional linear-theory value at high supersonic Mach numbers.

Use

CALL AER2 (SPEED, CLA), where

SPEED	is the Mach number and
CLA	is the computed lift-curve value in per degree units

Subroutine AFTCD

Purpose

Subroutine AFTCD computes the drag increment, as a function of alpha, due to the fuselage aft-end upsweep.

Use

CALL AFTCD (ALPHA, CDAFT), where

ALPHA	is the angle of attack of the wing
CDAFT	is the aft-end drag increment due to upsweep angle

Subroutines called

DLNT

Subroutine ATMOS

Purpose

Given a geometric altitude H in feet the following quantities are computed:

- T - Temperature in degrees Rankin
- SIGMA - Ratio of density to that at sea level
- D - Density in $\text{Lb-sec}^2/\text{ft}^4$
- THETA - Ratio of temperature to that at sea level
- DELTA - Ratio of pressure to that at sea level
- A - Speed of sound in ft/sec
- V - Viscosity coefficient in $\text{Lb-sec}/\text{ft}^2$
- K - Error indicator

Data represents mean annual, mid-latitude, dry air conditions. Data below altitudes of -5000 meters or greater than 90 kilometers is invalid.

The equations and tables used in computation are essentially the same as those used in computing U.S. Standard Atmosphere, 1962, and this routine maintains the same degree of accuracy as the tables in U.S. Standard Atmosphere, 1962.

Use

CALL ATMOS (H, T, SIGMA, D, THETA, DELTA, A, V, K)

Reference

U.S. Standard Atmosphere, 1962; Government Printing Office

Subroutine BDRG

Purpose

To compute the base drag of an arbitrary body. The subroutine uses an empirical equation to determine base drag.

Use

CALL BDRG (SPEED, AB, SREF, CBD), where

SPEED	Mach number (I)
AB	Base area (I)
SREF	Reference area (I)
CDB	Base drag coefficient (O)

BLOCK DATA

Purpose

To define data in COMMON blocks which represent tables and charts that are used in various subroutines.

Subroutine CDDR

Purpose

Subroutine CDDR calculates the drag rise along with defining the two limit Mach numbers and the lift-curve slope at those Mach numbers. The limit Mach numbers and associated lift-curves are used later in subroutine CDLL to define the polar shape in the transonic region.

CDDR has two entry points, on the first pass through the constants in the drag rise equation are computed along with the two limit Mach numbers and their associated lift-curves. For subsequent passes only the drag rise is computed.

Use

CALL CDDR (CL, XMACH, RNOFT, CDR)
CALL CDDR1

Subroutines called

CLWBT
FDRG
LNTP
WDRG

Subroutine CDL1

Purpose

This subroutine calculates the constants which are used by subroutine CDL2 to determine the drag polar.

Use

CALL CDL1 (SPEED, RNOFT, FK, DELCL, PRIMEK, AKD, AKB), where

SPEED	Mach number (I)
RNOFT	Reynolds no./foot (I)
FK	Polar shape factor below polar break lift coefficient (0)
DELCL	Polar lift displacement (0)
PRIMEK	Additional drag factor for drag polar above polar break (0)
AKD	Theoretical drag-due-to-lift factor (0)
AKB	Separation drag factor used to calculate drag polar above separation lift coefficient (0)

Subroutines called

ADCL
DLNT
KGIN
LNTP

Subroutine CDL2

Purpose

This subroutine computes the drag-due-to-lift using the polar shape factors determined by subroutine CDL1 and the polar break and separation lift coefficients determined by subroutine CLBRK.

Use

CALL CDL2 (SPEED, CL, AEROK, DELCL, PRIMEK, AKD, AKB, CDL),
where

SPEED	Mach number (I)
CL	Lift coefficient (I)
AEROK	Polar shape factor below polar break lift coefficient (I)
DELCL	Polar lift displacement (I)
PRIMEK	Additional drag factor for drag polar above polar break (I)
AKD	Theoretical drag-due-to-lift factor (I)
AKB	Separation drag factor (I)
CDL	Drag due to lift (0)

Subroutine CDRG

Purpose

This subroutine calculates the drag increment due to wing camber. Wing camber causes a lift displacement in the drag polar; this displacement lift increment is related to the difference between the minimum profile drag and the minimum drag of the polar.

Use

CALL CDRG (SPEED, AEROK, DELCL, CDC), where

SPEED	Mach number (I)
AEROK	Polar shape factor (I)
DELCL	Polar lift displacement (I)
CDC	Camber drag (O)

Subroutine CDWN

Purpose

This subroutine calculates the nose wave drag of body and nacelle components.

Use

CALL CDWN (AMAX, XLNOS, RIN, CDW), where

AMAX	Maximum cross-sectional area (I)
XLNOS	Length of nose (I)
RIN	Radius of inlet area (I)
CDW	Wave drag of component based on maximum cross-sectional area (O)

Subroutine called

DLNT

Subroutine CDWT

Purpose

This subroutine calculates the boattail wave drag of body and nacelle components.

Use

CALL CDWT (AMAX, XLAFT, REX, CDW), where

AMAX	Maximum cross-sectional area (I)
XLAFT	Length of boattail (I)
REX	Exit or base area (I)
CDW	Wave drag of component based on maximum cross-sectional area (O)

Subroutine called

LNTP

Subroutine CDWW

Purpose

This subroutine calculates the wave drag for airfoil surface components.

Use

CALL CDWW (CDOSR), where CDOSR is the wing wave drag based on the configuration reference area.

Also the data in COMMON BLKWPB must be defined prior to calling CDWW. The data in BLKWPB is

AR	Aspect ratio of surface based on exposed planform area
ZLAM	Taper ratio of exposed planform
ZLE	Leading-edge sweep
ZTE	Trailing-edge sweep
ZM	Mach number
SOSR	Exposed area to reference area ratio
TYPE	Airfoil type indicator
CAM	Section camber
TOC	Section thickness to chord ratio

Subroutine CFEQ

Purpose

This subroutine calculates the flat-plate skin friction coefficient using the White-Christoph technique for turbulent

flow, the Blasius relation for laminar flow, and a momentum thickness matching technique for partial laminar-turbulent flow.

Use

CALL CFEQ (RNOFT, ZMACH, CBAR, XTR, CF), where

RNOFT	Reynolds number/foot
ZMACH	Mach number
CBAR	Length
XTR	Distance along CBAR where transition occurs
CF	Skin friction coefficient

Subroutine CLBRK

Purpose

Subroutine CLBRK calculates the lift coefficients for polar break, separation drag onset, and maximum lift. The subroutine also calculates other aerodynamic parameters used in subroutine AERA to compute angle of attack as a function of C_L .

Use

CALL CLBRK (SPEED, RE, RNOFT), where

SPEED	Mach number (I)
RE	Reynolds number parameter (I)
RNOFT	Reynolds number/foot (I)

The output, contained in COMMON BLKCOL, is defined as,

CLPB	Polar break lift coefficient
CLDB	Separation lift coefficient
CLMAX	Maximum lift coefficient
ABRK	Angle of attack for polar break
AMAX	Angle of attack at CLMAX
DAMAX	Increment in angle of attack between AMAX and a linear value of alpha at CLMAX
DEL	Tail lift increment to CLMAX
CLS	Lift coefficient where CL versus alpha becomes nonlinear
ARLO	Aspect ratio limit between low AR and high AR calculations

Subroutines called

DLNT
LNTP

Subroutine CLWBT

Purpose

CLWBT controls the sequence of calculations that compute the total wing-body-tail lift curve slope, zero lift angle of attack, and the factors used to compute drag and lift increments due to a horizontal tail deflection.

Use

CALL CLWBT (SPEED), where SPEED is the Mach number, the output is contained in COMMON BLKCOL.

Subroutines called

AALO
ADJUST
AER2
TAIL

Subroutine CMOW

Purpose

Subroutine CMOW computes the moment at zero-lift for the wing-body configuration.

Use

CALL CMOW (SPEED, CMO), where

SPEED	Mach number (I)
CMO	Wing-body C_{m0}

Subroutine called

TLNT

Subroutine CONV

Purpose

CONV converts input data to feet and radian units for the namelist input program NINPT. The subroutine makes the conversion only to the variables that are redefined in the NINPT program.

Use

CALL CONV
CALL CONV1

Subroutine CPUOV

Purpose

This subroutine calculates the pressure coefficient, C_p at x/c for an infinite sheared wing. The C_p is obtained by solving Equations (93) and (90) in the Royal Aero. Soc. TDM-6312. The arrays corresponding to $S(1)(x)$, $S(2)(x)$, $S(3)(x)$, $S(4)(x)$, and $S(5)(x)$ in Equation (93) are obtained from COMMON BLKCPI and were defined in subroutine CPZT.

Use

CALL CPUOV (S, A, SWP, IV, CPI, CP, XM), where

S	Sign indicator (+1 for upper surface C_p , and -1 for lower surface C_p) (I)
A	Angle of attack (I)
SWP	Sweep angle (I)
IV	Control point, $x/c = \frac{1}{2}(1 - \cos(\frac{IV\pi}{32}))$ (I)
CPI	Incompressible C_p (O)
CP	Compressible value of C_p (O)
XM	Mach number (I)

Subroutine CPZT

Purpose

This subroutine computes the critical Mach number using the local Mach number normal to the isobar in the mid-span region

of the wing. The mid-chord sweep and the aspect ratio of the wing are used to define an effective isobar sweep. The subroutine obtains the airfoil geometry from subroutine SECT and the pressure distribution around the airfoil from subroutine CPUOV. The subroutine then uses the incompressible pressure at the crest of the airfoil and isentropic flow relationships to calculate the critical Mach number on the wing. The predicted wing critical Mach number is prevented from exceeding the critical Mach number of the fuselage which is calculated as a function of fuselage fineness ratio.

Use

CALL CPZT (ID, XMACH, TOC, CLD, SWEEP), where

ID	Type identification of the airfoil (I)
XMACH	Mach number for compressible C_p solutions (I) (set equal to 0.6 in Program MCRIT)
TOC	Airfoil section thickness ratio (I)
CLD	Airfoil section camber (I)
SWEEP	Mid-chord sweep at mid-semi-span of the wing (I)

The output which consists of a series of critical Mach number and lift combinations, decreasing from a six-degree angle of attack are contained in COMMON BLKA05.

Subroutines called

CPUOV
LNTP
SECT

Subroutine DAERO

Purpose

This subroutine computes the lift, moment and drag in free air and in ground effect of the high-lift system. Subroutine DAERO uses the incremental effect of flap and leading-edge device computed in subroutines DSET and MSET along with the clean airplane aerodynamics computed by program AERO to compute the total lift, drag and moment versus angle of attack in both free air and ground effect.

Use

CALL DAERO (ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG, H, DF), where

ALPHA	Angle of attack (I)
CL	Lift coefficient (0)
CD	Drag coefficient (0)
CM	Moment coefficient (0)
AGRD	Angle of attack in ground effect (0)
CLG	Lift coefficient in ground effect (0)
CDG	Drag coefficient in ground effect (0)
CMG	Moment coefficient in ground effect (0)
H	Height of wing $\bar{c}/4$ point above ground (I)
DF	Flap deflection (I)

Subroutines called

DLNT
LNTP

Function DLNT

Purpose

DLNT is a two-dimensional, nth-order Lagrangian interpolation procedure.

Use

CALL DLNT (XBAR, YBAR, X, Y, F, NX, NY, NXMAX, LOX, LOY),
where

XBAR	The X value at which a value of the function is to be interpolated (I)
YBAR	The Y value at which a value of the function is to be interpolated (I)
X	The array of X values (I)
Y	The array of Y values (I)
F	The values of the function $f(x,y)$ (I)
NX	The size of the X array and the F array in the X direction (I)
NY	The size of the Y array and the F array in the Y direction (I)
NXMAX	The dimension of the F array in the X direction in the calling routine (I)
LOX, LOY	Number of points to be used in the X and Y directions, respectively, in the interpolations: 1 for step, 2 for linear, 3 for parabolic, 4 for cubic, etc. (I)

Subroutine called

LNTP

Subroutine DMIN

Purpose

Subroutine DMIN controls the sequence of calculations necessary to compute minimum drag.

Use

CALL DMIN (SPEED, RNOFT, CDMIN), where

SPEED	Mach number (I)
RNOFT	Reynolds number/ft (I)
CDMIN	Minimum drag (O)

Subroutines called

ADJUST
BDRG
FDRG
WDRG

Subroutine DSET

Purpose

Subroutine DSET computes the incremental lift and drag of flaps and leading-edge devices. This subroutine uses the two-dimensional section data calculated in subroutine SSET along with the geometry of the high lift system in order to compute the incremental effect on the airplane.

Use

CALL DSET (BFI, BFO, BSI, BSO, CF), where

BFI	Inboard span station of flap (I)
BFO	Outboard span station of flap (I)
BSI	Inboard span station of leading-edge device (I)
BSO	Outboard span station of leading-edge device (I)
CF	Chord ratio of trailing-edge flap (I)

The output is contained in COMMON BLKCO2 where

DCLOF	Increment in lift at zero angle of attack due to flap deflection
DCLOS	Increment in lift at zero angle of attack due to leading-edge device
DCLMF	Increment in maximum lift due to flap deflection
DCLMS	Increment in maximum lift due to leading-edge device
DCDMIN	Increment in minimum drag due to high-lift system
DCLF	Displacement in drag polar due to flap deflection
DCLS	Displacement in drag polar due to leading-edge device
SPLANX	Planform area with high-lift system deployed
RCLA	Ratio of lift curve slope with high-lift system to clean airplane lift-curve slope.

Subroutines called

DLNT
INTP

Subroutine FDRG

Purpose

This subroutine calculates friction, form and interference drag for all the components on the airplane.

Use

CALL FDRG (SPEED, RNOFT), where

SPEED	Mach number (I)
RNOFT	Reynolds number/ft (I)

The output is contained in COMMON BLKCO1.

Subroutines called

CFEQ
FFACT
IFACT

Subroutine FFACT

Purpose

Subroutine FFACT computes the form factors for each component.

Use

CALL FFACT (ID, GEOM, TYP, CLD, SPEED, CRITM, FF), where

ID	Identification for bodies, nacelles, or surface components (I)
GEOM	Fineness ratio for bodies and nacelle components; or thickness ratio for surface components (I)
TYP	Airfoil type identification number (I)
CLD	Airfoil camber (I)
SPEED	Mach number (I)
CRITM	Critical Mach number of configuration (I)
FF	Form factor (O)

Program GEOM

Purpose

Program GEOM computes many of the geometric parameters that are used in aerodynamic calculations. The geometry that was read in as input is used to calculate the additional geometry needed for aerodynamic calculations. The input geometry is contained in COMMON's BLKA01, BLKA02, and BLKA03 and the geometry calculated in program GEOM is output in COMMON BLKG01.

Use

CALL OVERLAY (4HOVLY, 2, 0)

Subroutine IFACT

Purpose

Subroutine IFACT computes the interference factors for each component.

Use

CALL IFACT (ID, PARM, CRITM, SPEED, FI), where

ID	Identification for bodies or surface components (I)
PARM	Fuselage Reynolds number for bodies, or maximum thickness sweep for surface components (I)
CRITM	Critical Mach number of configuration (I)
SPEED	Mach number (I)
FI	Interference factor (O)

Subroutine called

DLNT

Program INPT

Purpose

The configuration geometry and the aerodynamic conditions to be run are read in using formatted input statements. The program also converts the input data from inch and degree units to feet and radians.

Use

CALL OVERLAY (4HOVLY, 1, 1)

Subroutine KGIN

Purpose

This subroutine computes the polar shape factor and polar displacement for a drag polar with drag rise added. A least-squares, second-degree curve is fitted to the drag polar with drag rise. This is needed by subroutine CDL1 in order to interpolate the polar shape in the transonic region between M_{L1} and M_{L2} .

Use

CALL KGIN (CLDB, AKIN, DECLIN, SPEED, AKOUT, DCIOUT), where

CLDB	Upper C_L limit for polar calculation (I)
AKIN	Polar shape factor without drag rise (I)
DECLIN	Polar displacement C_L without drag rise (I)

SPEED	Mach number (I)
AKOUT	Equivalent polar shape factor with drag rise (O)
DCLOUT	Equivalent polar displacement with drag rise (O)

Subroutines called

CDDR1
LSPCF

Subroutine LNTP

Purpose

LNTP is a one-dimensional, nth-order Lagrangian interpolation procedure.

Use

CALL LNTP (XBAR, YBAR, X, Y, M, NO), where

XBAR	The abscissa value at which an ordinate is to be interpolated (I)
YBAR	The interpolated ordinate (O)
X	The array of abscissas (I)
Y	The array of ordinates (I)
M	The size of the arrays (I)
NO	The number of points to be used in the interpolation. NO=1 for step, NO=2 for linear, NO=3 for parabolic, NO=4 for cubic, etc. (I)

Program LSHL

Purpose

LSHL controls the sequence of calculations to produce lift, moment, and drag variations with angle of attack for each low-speed, high-lift survey condition specified by the input. For each high-lift survey, the program computes the incremental effect for an inboard and an outboard high-lift system segment. The program then sets up a DO LOOP to calculate the total lift, drag and moment for changing angle of attack up to CLMAX.

Use

CALL OVERLAY (4HOVLY, 3, 4)

Subroutines called

DAERO
DSET
MSET
SSET

Program MCRIT

Purpose

Program MCRIT constructs a table of critical Mach number versus C_L from either an input table or by using an empirical method.

Use

CALL OVERLAY (4HOVLY, 3, 2)

Subroutines called

ADJUST
CPZT
LNTP

Program MRIT (MAIN PROGRAM)

Purpose

This program controls the logic of the calling of the three primary overlays. The lengths of the principal COMMON blocks used in all three overlays are also specified in this program.

Programs called

OVERLAY(4HOVLY, 1, 0)	(XINPT)
OVERLAY(4HOVLY, 2, 0)	(GEOM)
OVERLAY(4HOVLY, 3, 0)	(SURVEY)

Subroutine MSET

Purpose

Subroutine MSET computes the incremental moment at zero lift of flaps and leading-edge devices. This subroutine uses the two-dimensional section data calculated in Subroutine SSET along with the geometry of the high-lift system in order to compute the incremental effect on the airplane.

Use

CALL MSET (EFI, BFI, BSI, BSO), where

BFI	Inboard span station of flap (I)
BFO	Outboard span station of flap (I)
BSI	Inboard span station of leading-edge device (I)
BSO	Outboard span station of leading-edge device (I)

The output is contained in COMMON BLK002 where,

DCMO	Increment in C_{m0} due to high-lift system
CDMCL	Increment in slope of the moment versus lift curve

Subroutines called

DLNT

Subroutine MTXEQ

Purpose

MTXEQ solves a set of simultaneous equations. This subroutine will solve the matrix equation $AX = B$ for the unknown matrix X. The dimensions of the matrices must be:

A:	NxN	B:	NxM
----	-----	----	-----

where M is the number of B vectors to be transformed into X vectors. At the same time, this subroutine computes a scaled version of the determinant of the matrix A.

The solution of the matrix equation $AX = B$ is accomplished by upper triangularizing the A matrix using a maximum row element as a pivot point for each stage in the reduction of the matrix A.

This entails searching in the first column of the reduced $(N-K) \times (N-K)$ A matrix for the element whose absolute value is the largest. A row interchange is then performed to bring this element into the A_{kk} position. After completion of the triangularization, back substitution is used to obtain the X matrix.

Use

CALL MTXEQ (A, X, B, N, K), where

A	The array containing the elements of the matrix A. Array A must have dimensions $N \times N$.
X	The array containing the elements of the matrix X. Array X must have dimensions $N \times K$.
B	The array containing the elements of the matrix B. Array B must have dimensions $N \times K$.
N	Number of columns and rows in the matrix A.
K	Number of columns in the matrix B and X.

Program NINPT

Purpose

The configuration geometry and the aerodynamic conditions to be run are read in using NAMELIST input statements. The program calls subroutine CONV to convert the input data to feet and radians units. For cases which are to be rerun with only a slight change in input from the previous case, an indicator is used to read only certain sets of namelist data.

Use

CALL OVERLAY (4HOVLY, 1, 2)

Subroutines called

CONV
CONV1 (second entry point in CONV)

Subroutine PLSQ

Purpose

To fit a polynomial of degree K to the set of points (X_i, Y_i) for $i = 1, \dots, N$, by the method of least squares.

The method of solution is as follows:

Given a set of N points, coefficients are found such that:

$$Y = C(K+1) + C(K) \cdot X + C(K-1) \cdot X^2 + \dots \\ + C(2) \cdot X^{K-1} + C(1) \cdot X^K$$

is the best least-square fit to Y_i for an Kth degree fit.

PLSQ solves the regression matrix equation by calling MTXEQ.

Use

CALL PLSQ (X, Y, N, K, C, LIST, EMAX, ERMS, EMEQ), where

X	Array of N independent variables
Y	Array of N dependent variables
N	Number of variables
K	Degree of the least-squares polynomial
C	Array of coefficients, high order to low order, of the least-squares polynomial
LIST	Error analysis output indicator =0 suppresses error analysis printout =1 prints error analysis
EMAX	Maximum absolute error obtained by using the least-square fit polynomial to approximate the dependent variable
ERMS	RMS error obtained by using the least-square fit polynomial
EMEQ	Maximum derivation from unity in the linear system check solution

Subroutine called

MTXEQ

Subroutine SECT

Purpose

Subroutine SECT calculates the thickness and camber airfoil ordinates which are used in subroutine CPZT to calculate pressure distributions. SECT can calculate, using internal methods, the section data for the standard NACA 6-series and 4-digit airfoils along with a biconvex and Whitcomb's supercritical airfoil. The subroutine can also obtain the airfoil ordinates at the control points, x/c , needed for pressure solutions by interpolation on a table of input ordinates.

Use

CALL SECT (ID, TOC, CLD), where

ID	Airfoil section identification number (I)
TOC	Airfoil thickness ratio (I)
CLD	Airfoil camber (I)

The output is contained in COMMON BLKR07.

Subroutine called

LNTP

Subroutine SETUP

Purpose

Places initial values of 0.0 in the input and geometry common blocks. This is done to prevent indeterminate values to be used in program calculations if input data did not specify values.

Use

CALL SETUP

Subroutine SSET

Purpose

Subroutine SSET computes the two-dimensional section data used in subroutines DSET and MSET to compute the incremental lift, drag and moment due to a high-lift system. SSET either computes the two-dimensional increments using an empirical modification of thin-airfoil theory, or SSET obtains the two-dimensional increments from input data if available.

Use

CALL SSET (DF, DS, CFOC, CSOC, OTE, ROT), where

DF	An array containing the flap deflection of each element of the trailing-edge flap (I)
DS	Deflection of the leading-edge device (I)
CFOC	An array containing the chord length ratios of each element of the trailing-edge flap (I)
CSOC	Chord length ratio of the leading-edge device (I)
OTE	Airfoil trailing-edge angle (I)
ROT	Airfoil leading-edge radius divided by thickness (I)

The output is contained in COMMON BLK001, where

DC1F	Two-dimensional increment in lift at zero angle of attack due to flap deflection
DC1MF	Two-dimensional increment in maximum lift due to flap deflection
DCDFS	Increment in profile drag due to flap deflection
DCMFS	Two-dimensional increment in momentum at zero alpha due to flap deflection
DC1S	Two-dimensional increment in lift at zero angle of attack due to leading-edge device
DC1MS	Two-dimensional increment in maximum lift due to leading-edge device
DCDSS	Increment in profile drag due to leading-edge device

DCDMSS

Two-dimensional increment in moment at
zero alpha due to leading-edge device

Subroutines called

DLNT
LNTP

Program SURVEY

Purpose

SURVEY controls the sequence of calculations to produce a lift, moment and drag variation for each high-speed survey condition specified by the input.

For variable sweep configurations the program will first call program VGEOM with the wing leading-edge sweep set at the forward position and then recall program VGEOM with the sweep set at the aft position. This is done in order to setup program VGEOM for geometry calculations at any arbitrary sweep position. SURVEY then enters a DO LOOP where the high sweep survey conditions are set up. SURVEY calls VGEOM and MCRIT to recalculate the geometry and the configuration critical Mach number each time the leading-edge sweep is changed in a survey. SURVEY then enters an inner DO LOOP where a sequence of untrimmed C_L are generated and program AERO is called to obtain the trimmed lift, moment and drag. SURVEY then prints out the results.

If a series of low-speed, high-lift surveys were specified by the input, SURVEY then calls program LSHL to perform those calculations.

Use

CALL OVERLAY (4HOVLY, 3, 0)

Programs called

OVERLAY(4HOVLY, 3, 1)	(VGEOM)
OVERLAY(4HOVLY, 3, 2)	(MCRIT)
OVERLAY(4HOVLY, 3, 3)	(AERO)
OVERLAY(4HOVLY, 3, 4)	(LSHL)

Subroutine TAIL

Purpose

Subroutine TAIL computes the lift curve slope contribution of the tail along with factors used to compute lift and drag increments due to a horizontal tail deflection. These factors are computed by first solving for the downwash, dynamic-pressure, exposed area lift-curve slope, carry-over lift factors and induced drag for the tail.

Use

CALL TAIL (SPEED), where SPEED is the input Mach number.

The output, contained in COMMON BLKCO1, is

CLAT	Lift-curve slope contribution of the tail
A	Trim drag factor
B	Trim drag factor
AOH	Angle of zero lift of the tail
CLDH	Change in lift due to tail deflection factor
DEDA	Change in downwash per change in angle of attack

Subroutines called

AER2
LNTP

Subroutine TDRG

Purpose

Subroutine TDRG calculates moment using the wing-body C_{m_Q} and aerodynamic center along with the tail lift and moment arm. Depending upon user options the moment can be calculated at a fixed horizontal tail setting, or a tail setting can be computed which will trim out the moment. Lift and drag increments due to tail deflection are also computed.

Use

CALL TDRG (ITRIM, SPEED, DCLT, DCDT), where

ITRIM	Trim indicator, =0 if tail setting specified by input, =1 if tail setting is to be computed to trim out moment
SPEED	Mach number
DCLT	Increment in lift due to tail deflection relative to zero tail setting
DCDT	Increment in drag due to tail deflection relative to zero tail setting

Subroutine TLNT

Purpose

TLNT is a triple-linear interpolation procedure.

Use

CALL TLNT (XBAR, YBAR, ZBAR, FBAR, X, Y, Z, F, NX, NY, NZ, NXMAX, NYMAX), where

XBAR	The X value at which a value of the function is to be interpolated (I)
YBAR	The Y value at which a value of the function is to be interpolated (I)
ZBAR	The Z value at which a value of the function is to be interpolated (I)
FBAR	The interpolated value of the function F(X,Y,X) (I)
X	The array of X values (I)
Y	The array of Y values (I)
Z	The array of Z values (I)
F	The three-dimension array F values (I)
NX	The size of the X array and the F array in the X direction (I)
NY	The size of the Y array and the F array in the Y direction (I)
NZ	The size of the Z array and the F array in the Z direction (I)
NXMAX	The dimension of the F array in the X direction in the calling routine (I)
NYMAX	The dimension of the F array in the Y direction in the calling routine (I)

Subroutines called

DLNT, LNTF

Program VGEOM

Purpose

Program VGEOM computes the geometry parameters that vary with wing sweep for variable-sweep configurations. The program is first called by program SURVEY at the forward and most aft sweep positions in order to set up VGEOM for any arbitrary sweep calculation.

Use

CALL OVERLAY (4HOVLY, 3, 1)

Subroutine WBAC

Purpose

Subroutine WBAC computes the aerodynamic center location for wing-body configurations. The subroutine calculates the aerodynamic center of the wing carry-over lift on the body, and the aerodynamic center of the forebody. A composite aerodynamic center is then computed using the aerodynamic center and lift curve slope of each component.

Use

CALL WBAC (SPEED, XACR), where

SPEED	Mach number (I)
XACR	Aerodynamic center of the wing-body configuration reference to the leading edge of the exposed root chord

Subroutines called

ACCR
DLNT
LNTP

Subroutine WDRG

Purpose

This subroutine calculates the wave drag for all the components on the airplane.

Use

CALL WDRG (FMACH), where

FMACH Mach number

The output is contained in COMMON BLKC01.

Subroutines called

CDWN
CDWT
CDWW

Program XINPT

Purpose

XINPT controls the sequence of input for new problems or allows the previous problem to be repeated with only a small change in input. For each new problem the title card, print dump card, and input control card are read from the input file. The subroutine SETUP is then called to zero out all the input common blocks, then depending upon the input control card either of the secondary overlay programs INPT or NINPT are then called to continue reading the input file. After the input file is read for a given problem, control is then transferred back to MRIT where the solution to the problem is obtained. After each solution is obtained control is then transferred back to XINPT and the repeat problem control card is then read. If end-of-file on tape 5 is read the program is terminated. If a perturbation of the previous problem is indicated, NINPT is called where the revised input is then read.

Use

CALL OVERLAY (4HOVLY, 1, 0)

Programs called

SETUP	
OVERLAY(4HOVLY, 1, 1)	(INPT)
OVERLAY(4HOVLY, 1, 2)	(NINPT)
EXIT	

5. PROGRAM LISTING


```

OVERLAY (OVLY,0,0)
PROGRAM MRIT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C   LARGE AIRCRAFT AERODYNAMIC PREDICTION PROCEDURE
C
COMMON /BLKA01/ N4(7)
COMMON /BLKA02/ A02(433)
COMMON /BLKA03/ A03(20)
COMMON /BLKA04/ N3(3), A04(78)
COMMON /BLKA05/ A05(131), NMCR, NXSET
COMMON /BLKSUR/ N1(42), SUR(370)
COMMON /BLKHLS/ N2, HLS(70)
COMMON /BLKG01/ G1(200)
COMMON /BLKPRT/ KPRINT(50)
COMMON /BLKADJ/ ADJ(322)
COMMON /BLKTIL/ TITLE(6)
COMMON /BLKOV1/ IJ
C
15 IJ = 0
20 CALL OVERLAY(4HOVLY,1,0)
C
CALL OVERLAY(4HOVLY,2,0)
CALL OVERLAY(4HOVLY,3,0)
C
GO TO 20
C
END

```

```

R1T0001
R1T0002
R1T0003
R1T0004
R1T0005
R1T0006
R1T0007
R1T0008
R1T0009
R1T0010
R1T0011
R1T0012
R1T0013
R1T0014
R1T0015
R1T0016
R1T0017
R1T0021
R1T0022
R1T0023
R1T0024
R1T0025
R1T0026
R1T0027
R1T0028
R1T0029

```

CC = 00026

	SUBROUTINE LNTP(XBAR,YBAR,X,Y,M,NO)	R1T0031
C		R1T0032
C	LAGRANGIAN INTERPOLATION	R1T0033
C	1) INCREASING OR DECREASING X ARRAY	R1T0034
C	2) LINEAR EXTRAPOLATION ONLY	R1T0035
C	3) NO = NUMBER OF POINTS USED IN INTERPOLATION	R1T0036
C	4) M = TOTAL NUMBER OF POINTS IN X ARRAY	R1T0037
C	5) IF(NO.LE.0.OR.M.LE.0) YBAR = 0.0	R1T0038
C	6) IF(M.EQ.1.AND.NO.GT.0) YBAR=Y(1)	R1T0039
C	7) IF(NO.EQ.1.AND.M.GT.0) YBAR=Y(NEAREST X(I))	R1T0040
C	8) IF(NO.GT.M) INTERPOLATION AS IF NO=M	R1T0041
C		R1T0042
C	DIMENSION X(M) , Y(M)	R1T0043
	YBAR=0.0	R1T0044
	N=NO	R1T0045
	IF(N.GT.M) N=M	R1T0046
	IF(M.LE.0.OR.N.LE.0) RETURN	R1T0047
	IF(M.GT.1) GO TO 10	R1T0048
	YBAR=Y(1)	R1T0049
	RETURN	R1T0050
C		R1T0051
	10 MDC=0	R1T0052
	IF(X(M).LT.X(1)) MDC=1	R1T0053
	DO 20 I=2,M	R1T0054
	I1=I-1+MDC	R1T0055
	I2=I-MDC	R1T0056
	IF(XBAR.GE.X(I1).AND.XBAR.LE.X(I2)) GO TO 40	R1T0057
	20 CONTINUE	R1T0058
	I1=1	R1T0059
	IF((MDC.EQ.0.AND.XBAR.GE.X(M)).OR.(MDC.NE.0.AND.XBAR.LE.X(M)))	R1T0060
	1 I1=M-1	R1T0061
	I2=I1+1	R1T0062
	GO TO 45	R1T0063
C		R1T0064
	40 IF(N.GT.2) GO TO 60	R1T0065
	45 IF(N.LT.2) GO TO 55	R1T0066
	50 YBAR=Y(I1)+(XBAR-X(I1))*(Y(I2)-Y(I1))/(X(I2)-X(I1))	R1T0067
	RETURN	R1T0068
C		R1T0069
	55 YBAR=Y(I1)	R1T0070
	IF(ABS(XBAR-X(I2)).LT.ABS(XBAR-X(I1))) YBAR=Y(I2)	R1T0071
	RETURN	R1T0072
C		R1T0073
	60 J=I-N/2	R1T0074
	IF(I.LE.N/2+1) J=1	R1T0075
	IF(I.GT.M-N/2) J=M-N+1	R1T0076
	NT=N+J-1	R1T0077
C		R1T0078
	DO 80 I=J,NT	R1T0079
	ELL=1.0	R1T0080
	DO 70 K=J,NT	R1T0081
	IF(K.NE.I) ELL=FLL*(XBAR-X(K))/(X(I)-X(K))	R1T0082
	70 CONTINUE	R1T0083
	80 YBAR=YBAR+ELL*Y(I)	R1T0084
	RETURN	R1T0085
		R1T0086

C

END

R1T0087
R1T0088

CC = 00058

C	FUNCTION DLNT (XBAR,YBAR,X,Y,F,NX,NY,NXMAX,LCX,LOY)	R1T0090
C	DOUBLE LAGRANGE INTERPOLATION	R1T0091
C		R1T0092
C	DIMENSION X(1),Y(1),F(NXMAX,1),FT(25)	R1T0093
C		R1T0094
	FBAR=0.0	R1T0095
	IF(LOY.GT.25) LOY=25	R1T0096
	LLX=LOX	R1T0097
	IF(LOX.GT.NX) LLX=NX	R1T0098
	IF(NY.GT.1) GO TO 10	R1T0099
	CALL LNTP (XBAR,FBAR,X,F(1,1),NX,LLX)	R1T0100
	GO TO 30	R1T0101
C		R1T0102
	10 LLY=LOY	R1T0103
	IF(LOY.GT.NY) LLY=NY	R1T0104
	DO 20 I=1,NY	R1T0105
	CALL LNTP (XBAR,FT(I),X,F(1,I),NX,LLX)	R1T0106
	20 CONTINUE	R1T0107
	CALL LNTP (YBAR,FBAR,Y,FT,NY,LLY)	R1T0108
C		R1T0109
	30 DLNT=FBAR	R1T0110
C		R1T0111
	RETURN	R1T0112
	END	R1T0113
		R1T0114

CC = 00025

	SUBROUTINE TLNT(XBAR,YBAR,ZBAR,FBAR,X,Y,Z,F,NX,NY,NZ,NXMAX,NYMAX)	R1T0116
C		R1T0117
C	TRIPLE LINEAR INTERPOLATOR	R1T0118
C		R1T0119
	DIMENSION X(1),Y(1),Z(1),F(NXMAX,NYMAX,1),FT(25)	R1T0120
C		R1T0121
	DO 10 I=1,NZ	R1T0122
C		R1T0123
	10 FT(I) = DLNT(XBAR,YBAR,X,Y,F(1,1,I),NX,NY,NXMAX,2,2)	R1T0124
C		R1T0125
	CALL LNTP(ZBAR,FBAR,Z,FT,NZ,2)	R1T0126
C		R1T0127
	RETURN	R1T0128
	END	R1T0129

CC = 00014

C	OVERLAY(1,0)	R1T0131
	PROGRAM XINPT	R1T0132
C	COMMON /BLKPRT/ KPRINT(50)	R1T0133
	COMMON /BLKTIL/ TITLE(6)	R1T0134
	COMMON /BLKQV1/ IJ	R1T0135
C	DIMENSION W(4), IS(12)	R1T0136
	DATA W / 4HFORM, 4HNAME, 4HSAVE, 4HEND /	R1T0137
C	IF(IJ.GE.1) GO TO 70	R1T0138
	20 CONTINUE	R1T0139
C		R1T0140
C	READ(5,1000) (TITLE(I), I = 1,6)	R1T0141
	IF (EOF(5).NE.0) CALL EXIT	R1T0142
	WRITE(6,2000) (TITLE(I), I= 1,6)	R1T0143
C		R1T0144
	READ(5,1001) (KPRINT(I), I=1,50)	R1T0145
C		R1T0146
	DO 50 I = 1, 50	R1T0147
	IF(KPRINT(I).GT.0) WRITE(6,2001) I, KPRINT(I)	R1T0148
	50 CONTINUE	R1T0149
	CALL SETUP	R1T0150
C		R1T0151
	IJ = 1	R1T0152
	70 READ(5,1002) WORD	R1T0153
	IF(EOF(5).NE.0) CALL EXIT	R1T0154
	IF(WORD.EQ.W(1)) CALL OVERLAY(4HOVLY,1,1)	R1T0155
	IF(WORD.EQ.W(2)) IJ = 1	R1T0156
	IF(WORD.EQ.W(2)) CALL OVERLAY(4HOVLY,1,2)	R1T0157
	IF(WORD.EQ.W(3)) IJ = 2	R1T0158
	IF(WORD.EQ.W(3)) CALL OVERLAY(4HOVLY,1,2)	R1T0159
	IF(WORD.EQ.W(4)) GO TO 20	R1T0160
C		R1T0161
	1000 FORMAT(6A10)	R1T0162
	1001 FORMAT(50I1)	R1T0163
	1002 FORMAT(A4)	R1T0164
	2000 FORMAT(1H1,/////, 27X,*EMPIRICALLY BASED COMPUTER PROGRAM* //	R1T0165
	1 23X,*TO PREDICT THE AERODYNAMIC CHARACTERISTICS*	R1T0166
	2 // 35X,*OF LARGE AIRCRAFT* (10(/)), 27X, 6A10)	R1T0167
	2001 FORMAT(5X,*KPRINT(*,I2,*) = *,I2)	R1T0168
C		R1T0169
	END	R1T0170
		R1T0171
		R1T0172
		R1T0173
		R1T0174
		R1T0175

CC = 00044

	SUBROUTINE SETUP	R1T0177
C		R1T0178
C	ZER0ES OUT INPUT COMMON BLOCKS	R1T0179
C		R1T0180
	COMMON /BLKA01/ N(7)	R1T0181
	COMMON /BLKA02/ A02(433)	R1T0182
	COMMON /BLKA03/ A03(20)	R1T0183
	COMMON /BLKA04/ N3(3), A04(78)	R1T0184
	COMMON /BLKSUR/ N1(42), SUR(370)	R1T0185
	COMMON /BLKHLS/ N2, HLS(70)	R1T0186
	COMMON /BLKADJ/ IVAL(20),X(15),YM(135),YA(135),XCL(15),NXVAR,NXCL	R1T0187
	COMMON /BLKG01/ G1(200)	R1T0188
C		R1T0189
	N2 = 0	R1T0190
	NXVAR = 1	R1T0191
	NXCL = 1	R1T0191
C		R1T0192
	DO 10 I = 1, 20	R1T0193
	A03(I) = 0.0	R1T0194
	IVAL(I) = 0	R1T0195
	IF(I.LE.3) N3(I) = 0	R1T0196
	IF (I.LE.7) N(I) = 0	R1T0197
	IF (I.LE.15) X(I) = 0.0	R1T0198
	IF(I.LE.15) XCL(I) = 0.0	R1T0199
10	CONTINUE	R1T0199
C		R1T0200
	DO 20 I = 1, 433	R1T0201
	A02(I) = 0.0	R1T0202
	IF(I.LE.41) N1(I) = 0	R1T0203
	IF(I.LE.50) HLS(I) = 0.0	R1T0204
	IF(I.LE.78) A04(I) = 0.0	R1T0205
	IF(I.LE.140) SUR(I) = 0.0	R1T0206
	IF(I.LE.200) G1(I) = 0.0	R1T0207
	IF(I.GT.135) GO TO 20	R1T0208
	YM(I) = 1.0	R1T0209
	YA(I) = 0.0	R1T0210
20	CONTINUE	R1T0211
C		R1T0212
	DO 30 I = 1, 20	R1T0213
	I1 = I + 44	R1T0214
	I2 = I + 204	R1T0215
	I3 = I + 379	R1T0216
	A02(I1) = 1.0	R1T0217
	A02(I2) = 1.0	R1T0218
	IF(I.LE.10) A02(I3) = 1.0	R1T0219
30	CONTINUE	R1T0220
C		R1T0221
	RETURN	R1T0222
	END	R1T0223

CC = 00049

	OVERLAY(1,1)	R1T0225
	PROGRAM INPT	R1T0226
C		R1T0227
C	PROBLEM DATA INPUT	R1T0228
C		R1T0229
	COMMON /BLKA01/ NBODY5, NNACS, NSURFS, NHT, NVT, ISWP, NPALS	R1T0230
	COMMON /BLKA02/ SKEF, AR, TAPR, SWPLE,	R1T0231
1	BLEN(10), BWID(10), BHGT(10), BAWET(10), BC(10),	R1T0232
2	BNJ(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),	R1T0233
3	BASE(10), ELEN(10), EWID(10), EHGT(10),	R1T0234
4	EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),	R1T0235
5	ELBT(10), EQF(10), ENQ(10), CBAR(10), TW,	R1T0236
5	XLEW, YHW, YB, CR, B02, BFUS, FMISC, AB, AFTAW,	R1T0237
6	CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CCNCL,	R1T0238
7	TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),	R1T0239
8	XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,	R1T0240
9	SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),	R1T0241
1	SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),	R1T0242
2	SCR(10), HTLE, HTY, HTZ, HTINC	R1T0243
	COMMON /BLKA03/ ROUGHK, CLE(3), CCR(3), YC(3), XCG, ZCG, CMAC, IREF, A1(6)	R1T0244
	COMMON /BLKA04/ IHLS, NF, NS, FLAP, SLAT, DF(5), CFQC(5),	R1T0245
1	DCLQF(5), DCLMF(5), DCLDF(5), DCMQF(5),	R1T0246
2	DS(5), CSOC(5), DCLOS(5), DCLMS(5), DCDS(5),	R1T0247
3	DCMS(5), BF11, BF10, CF1(3), BF21, BF20, CF2(3),	R1T0248
4	BS11, BS10, CS1, BS21, BS20, CS2	R1T0249
	COMMON /BLKA05/ XMU, ZMU, XML, ZML, ZPTE, ZTHICK,	R1T0250
1	RLE, OTE, DY, XMT, DYC,	R1T0251
2	CIMAX, CLMCR(10), XMCR(10),	R1T0252
3	XT(33), YT(33), XYC(33), NMCR, NXSET	R1T0253
C		R1T0254
	COMMON /BLKSUR/ NSURV, NCLAS, IT(20), ITRM(20), FMSURV(20),	R1T0255
1	ALT(20), DHSV(20), SWPV(20), CLLO(20), CLHI(20),	R1T0256
2	TRB(5,10), TRN(5,10), TRU(5,10), TRL(5,10),	R1T0257
3	TRS(5,10)	R1T0258
	COMMON /BLKHLS/ NHLSV, DFI(3,5), CPF(5), DSI(5), CPS(5),	R1T0259
1	DELC(5), H(5), DFI2(3,5), CPF2(5), DSI2(5),	R1T0260
2	CPS2(5)	R1T0261
C		R1T0262
	COMMON /BLKADJ/ IVAL(20), X(15), YM(15,9), YA(15,9), XCL(15),	R1T0263
1	NXVAR, NXCL	R1T0264
C		R1T0265
	DIMENSION W(10), TTT(22), TS1(10), TS3(10)	R1T0266
C		R1T0267
	DATA W / 4HPLAI, 4HS.S., 4HD.S., 4HT.S., 4HINPU, 4HL.E.,	R1T0268
1	4HSLAT, 4HKRUE, 4HINPU, 4HADJU/	R1T0269
	DATA TTT / 3H63-, 3H64-, 3H65-, 3H66-, 3H63A, 3H64A, 3H65A, 3HSUP, 3HBIC,	R1T0270
1	3H-62, 3H-63, 3H-64, 3H-65, 3H-66, 3H-33, 3H-34, 3H-35, 3H-93,	R1T0271
2	3H-94, 3H-95, 3HINP, 3H2IN /	R1T0272
	DATA TS1 / 10*0.0 /, TS3 / 10* 0.0 /	R1T0273
C		R1T0274
C	CONFIGURATION DEFINITION	R1T0275
C		R1T0276
	100 READ(5,1001) NBODY5, NNACS, NSURFS, NHT, NVT, ISWP, NPALS, IHLS,	R1T0277
1	IREF	R1T0278
	WRITE(6,2000) NBODY5, NNACS, NSURFS, NHT, NVT, ISWP, NPALS	R1T0279
C		R1T0280

	READ(5,1000) SREF, CMAC, XCG, ZCG, ROUGHK, FMISC	R1T0281
	WRITE(6,2001) SREF, CMAC, XCG, ZCG	R1T0282
C		R1T0283
	IF(NBDYS.EQ.0) GO TO 120	R1T0284
	DO 110 I = 1, NBDYS	R1T0285
	READ(5,1000) BLEN(I), BWID(I), BHGT(I), BAWET(I), BQ(I), BNO(I)	R1T0286
	IF(I.EQ.1) READ(5,1008) BAMX(I), BABS(I), BLNS(I), BLBT(I),	R1T0287
	1 BASE(I), BFUS, AB	R1T0288
	IF(I.NE.1) READ(5,1000) BAMX(I), BABS(I), BLNS(I), BLBT(I),	R1T0289
	1 BASE(I)	R1T0290
C		R1T0291
	IF(BQ(I).EQ.0.0) BQ(I) = 1.0	R1T0292
	IF(BNO(I).EQ.0.0) BNO(I) = 1.0	R1T0293
	IF(I.EQ.1) WRITE(6,2002)	R1T0294
	WRITE(6,2003) I, BLEN(I), BWID(I), BHGT(I), BAWET(I), BAMX(I), BABS(I),	R1T0295
	1 BLNS(I), BLBT(I), BASE(I), BQ(I), BNO(I)	R1T0296
C		R1T0297
	110 CONTINUE	R1T0298
C		R1T0299
	120 IF(NNACS.EQ.0) GO TO 140	R1T0300
	DO 130 I = 1, NNACS	R1T0301
	READ(5,1000) ELEN(I), EWID(I), EHGT(I), EAWET(I), EQF(I), ENO(I)	R1T0302
	READ(5,1000) EAMX(I), EIN(I), EXIT(I), ELNS(I), ELBT(I)	R1T0303
C		R1T0304
	IF(EQF(I).EQ.0.0) EQF(I) = 1.0	R1T0305
	IF(ENO(I).EQ.0.0) ENO(I) = 1.0	R1T0306
	IF(I.EQ.1) WRITE(6,2004)	R1T0307
	WRITE(6,2003) I, ELEN(I), EWID(I), EHGT(I), EAWET(I), EAMX(I), EIN(I),	R1T0308
	1 EXIT(I), ELNS(I), ELBT(I), EQF(I), ENO(I)	R1T0309
C		R1T0310
	130 CONTINUE	R1T0311
C		R1T0312
	140 IF(NPNLS.GT.1) GO TO 145	R1T0313
	READ(5,1000) AR, TAPR, SWPLE, SPLAN, TWIST, WINC	R1T0314
	WRITE(6,2006) AR, TAPR, SWPLE, SPLAN, TWIST, WINC	R1T0315
	READ(5,1004) TW1, TW2, TW3, CAM(1), TOC(1), XLEW, YWW, YB	R1T0316
	READ(5,1000) SWMT, CBAR(1), AWET(1), CONCL	R1T0317
C		R1T0318
	WRITE(6,2008) TW1, TW2, TW3, CAM(1), TOC(1), XLEW, YWW, YB,	R1T0319
	1 SWMT, CBAR(1), AWET(1)	R1T0320
	GO TO 160	R1T0321
C		R1T0322
	145 READ(5,1000) SPLAN, TAPR, SWPLE	R1T0323
	READ(5,1004) TW1, TW2, TW3, TWIST, WINC, SWMT, CONCL	R1T0324
	WRITE(6,2009) TW1, TW2, TW3, TWIST, WINC, SWMT	R1T0325
	IF(ISWP.GT.0) WRITE(6,2010) SPLAN, TAPR, SWPLE	R1T0326
C		R1T0327
	DO 150 I = 1, NPNLS	R1T0328
	IF(I.EQ.1) WRITE(6,2011)	R1T0329
	READ(5,1008) CAM(I), TOC(I), XLE(I), YW(I), CRW(I),	R1T0330
	1 CBAR(I), AWET(I)	R1T0331
	WRITE(6,2003) I, CAM(I), TOC(I), XLE(I), YW(I), CRW(I),	R1T0332
	1 CBAR(I), AWET(I)	R1T0333
C		R1T0334
	150 CONTINUE	R1T0335
	I = 1 + NPNLS	R1T0336

	READ(5,1000) XLE(I), YW(I), CRW(I)	R1T0337
	WRITE(6,2012) XLE(I), YW(I), CRW(I)	R1T0338
C		R1T0339
	160 IF(TW1.EQ.TTT(8)) READ(5,1000) XMJ,ZMJ,XPL,ZPL,ZPTE,ZTHICK	R1T0340
C		R1T0341
	IF(TW1.EQ.TTT(21).OR.TW1.EQ.TTT(22)) READ(5,1000)	R1T0342
1	RLE, OTE, DY, XMT, DYC, CLMAX	R1T0343
	IF(TW1.EQ.TTT(21)) READ(5,1009) NXSET, (XT(I), YT(I),	R1T0344
1	XYC(I), I = 1,NXSET)	R1T0345
	IF(TW1.EQ.TTT(22)) READ(5,1009) NMCR, (CLMCR(I),	R1T0346
1	XMCR(I), I = 1, NMCR)	R1T0347
	IF(ISWP.EQ.0) GO TO 170	R1T0348
	READ(5,1008) XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC, AFTAW	R1T0349
C		R1T0350
	170 IF(NSURFS.LE.1) GO TO 185	R1T0351
	DO 180 I = 2, NSURFS	R1T0352
	J = I - 1	R1T0353
	IF(J.EQ.1) WRITE(6,2013)	R1T0354
	READ(5,1004) TS1(J),TS2,TS3(J), SCAM(J), STOC(J), SMTSW(J),	R1T0355
1	SHF(J)	R1T0356
	READ(5,1000) SWL(J), SWT(J), STAPR(J), SCR(J), SBAK(J), SAWET(J)	R1T0357
	WRITE(6,2007) TS1(J),TS2,TS3(J), SCAM(J),STOC(J),SMTSW(J),	R1T0358
1	SHF(J),SWL(J),SWT(J),STAPR(J),SCR(J),SBAK(J),	R1T0359
2	SAWET(J)	R1T0360
	IF(NHT.NE.0.AND.J.EQ.1) READ(5,1000) HTLE, HTY, HTZ, HTINC	R1T0361
C		R1T0362
	180 CONTINUE	R1T0363
	185 IF(NPNLS.LE.2) GO TO 190	R1T0364
C		R1T0365
	READ(5,1000) ICLE(I), YC(I), CCR(I), I =1,3)	R1T0366
C		R1T0367
	190 CONTINUE	R1T0368
C		R1T0369
C		R1T0370
C	HIGH LIFT SYSTEM GEOMETRY IS READ IN IF IHLS GE 1	R1T0371
C		R1T0372
	200 IF(IHLS.EQ.0) GO TO 205	R1T0373
	READ(5,1007) FLAP, SLAT, DF, NS	R1T0374
	READ(5,1000) BF1I, BF1O, CF1(1), BS1I, BS1O, CS1	R1T0375
	IF(IHLS.EQ.2) READ(5,1000) BF2I,BF2O,CF2(1),BS2I,BS2O,CS2	R1T0376
	IF(FLAP.EQ.W(3).OR.FLAP.EQ.W(4)) READ(5,1000) CF1(2), CF1(3),	R1T0377
1	CF2(2), CF2(3)	R1T0378
	IF(FLAP.EQ.W(5)) READ(5,1000) (DF(I), CFCC(I), DCLOF(I),	R1T0379
1	DCLMF(I), DCDF(I), DCMOF(I), I=1,NF)	R1T0380
C		R1T0381
	IF(SLAT.EQ.W(5)) READ(5,1000) (DS(I), CSOC(I), DCLOS(I),	R1T0382
1	DCLMS(I), DCOS(I), DCMOS(I), I=1,NS)	R1T0383
C		R1T0384
C	PROBLEM CONTROL	R1T0385
C		R1T0386
	205 READ(5,1003) NSURV, NHLSV, NCLAS, (IT(I),I=1,20), (ITRM(I),I=1,20)	R1T0387
	IF(NSURV.EQ.0) GO TO 220	R1T0388
	WRITE(6,2014) NSURV	R1T0389
	DO 210 I =1, NSURV	R1T0390
	READ(5,1000) FMSURV(I), ALT(I), DHSV(I), SWPV(I), CLLO(I),CLHI(I)	R1T0391
	IF(ITRM(I).EQ.0) WRITE(6,2015) FMSURV(I), ALT(I), DHSV(I),	R1T0392

1	SWPV(I), CLLO(I), CLHI(I)	R1T0393
IF(ITRM(I).EQ.1)	WRITE(6,2016) FMSURV(I), ALT(I), SWPV(I),	R1T0394
1	CLLO(I), CLHI(I)	R1T0395
210	CONTINUE	R1T0396
C		R1T0397
C	B.L. TRANSITION LOCATIONS ARE READ IN FOR EACH COMPONENT	R1T0398
C		R1T0399
	J1 = 0	R1T0400
	DO 215 I = 1, NSURV	R1T0401
	IF(IT(I).EQ.0) GO TO 215	R1T0402
	J = IT(I)	R1T0403
	IF(J.LE.J1) GO TO 215	R1T0404
	IF(NBDYS.GT.0) READ(5,1000) (TRB(J,K), K=1,NBDYS)	R1T0405
	IF(NNACS.GT.0) READ(5,1000) (TRN(J,K), K=1,NNACS)	R1T0406
	IF(NPNLS.GT.0) READ(5,1000) (TRU(J,K),TRL(J,K),K=1,NPNLS)	R1T0407
	IF(NSURFS.GT.0) READ(5,1000) (TRS(J,K), K=2, NSURFS)	R1T0408
	J1 = J	R1T0409
215	CONTINUE	R1T0410
C		R1T0411
C		R1T0412
220	IF(NHLSV.EQ.0) GO TO 250	R1T0413
	DO 230 I = 1, NHLSV	R1T0414
	NDF = 1	R1T0415
	IF(FLAP.EQ.W(3)) NDF = 2	R1T0416
	IF(FLAP.EQ.W(4)) NDF = 3	R1T0417
C		R1T0418
	READ(5,1000) (DFI(J,I), J=1,NDF), CPF(I), DSI(I), CPS(I),	R1T0419
1	DELCD(I), H(I)	R1T0420
	IF(IHLS.EQ.2) READ(5,1000) (DFI2(J,I), J=1,NDF), CPF2(I),	R1T0421
1	DSI2(I), CPS2(I)	R1T0422
230	CONTINUE	R1T0423
C		R1T0424
C		R1T0425
250	READ(5,1005) WORD, WRD	R1T0426
	IF(WORD.EQ.W(10)) GO TO 260	R1T0427
	GO TO 270	R1T0428
260	READ(5,1006) (IVAL(I), I=1,20), NXVAR, NADJ, NXCL, NADJ2	R1T0429
	IF(NXVAR.EQ.0) GO TO 266	R1T0430
	READ(5,1000) (X(I), I = 1, NXVAR)	R1T0431
	DO 265 I = 1, NADJ	R1T0432
	READ(5,1000) (YM(J,I), YA(J,I), J= 1, NXVAR)	R1T0433
265	CONTINUE	R1T0434
266	IF(NXCL.EQ.0) GO TO 268	R1T0435
	READ(5,1000) (XCL(I), I = 1, NXCL)	R1T0436
	DO 267 I = 1, NADJ2	R1T0437
	READ(5,1000) (YM(J,NXVAR+I), YA(J,NXVAR+I), J=1,NXCL)	R1T0438
267	CONTINUE	R1T0439
268	CONTINUE	R1T0440
	READ(5,1005) WORD, WRD	R1T0441
C		R1T0442
270	WRITE(6,2005) WORD,WRD	R1T0443
C		R1T0444
C		R1T0445
	A = 1./57.2957796	R1T0446
	B = 1./12.	R1T0447
	C = 1./144.	R1T0448

C
 SWPLE = SWPLE * A
 SWMT = SWMT * A
 AFTSW = AFTSW * A
 DO 300 I = 1, 10
 SMTSW(I) = SMTSW(I) * A
 SWL(I) = SWL(I) * A
 SWT(I) = SWT(I) * A
 300 CONTINUE

C
 XPIVOT = XPIVOT * B
 YPIVOT = YPIVOT * B
 XAPEX = XAPEX * B
 AFTCB = AFTCB * B
 HTLE = HTLE * B
 HTY = HTY * B
 HTZ = HTZ * B
 CMAC = CMAC * B
 XCG = XCG * B
 ZCG = ZCG * B
 XLEW = XLEW * B
 YWW = YWW * B
 YB = YB * B
 CR = CR * B
 BOZ = BOZ * B
 DO 310 I = 1, 10
 BLEN(I) = BLEN(I) * B
 BWID(I) = BWID(I) * B
 BHGT(I) = BHGT(I) * B
 BLNS(I) = BLNS(I) * B
 BLBT(I) = BLBT(I) * B
 ELEN(I) = ELEN(I) * B
 EWID(I) = EWID(I) * B
 EHGT(I) = EHGT(I) * B
 ELNS(I) = ELNS(I) * B
 ELBT(I) = ELBT(I) * B
 CBAR(I) = CBAR(I) * B
 XLE(I) = XLE(I) * B
 CRW(I) = CRW(I) * B
 YW(I) = YW(I) * B
 SBAR(I) = SBAR(I) * B
 SCR(I) = SCR(I) * B
 310 CONTINUE
 XLE(11) = XLE(11) * B
 CRW(11) = CRW(11) * B
 YW(11) = YW(11) * B
 DO 315 I = 1, 3
 CLE(I) = CLE(I) * B
 CCR(I) = CCR(I) * B
 YC(I) = YC(I) * B
 315 CONTINUE
 C
 DO 320 I = 1, 10
 BAMX(I) = BAMX(I) * C
 BABS(I) = BABS(I) * C
 BASE(I) = BASE(I) * C

R1T0449
 R1T0450
 R1T0451
 R1T0452
 R1T0453
 R1T0454
 R1T0455
 R1T0456
 R1T0457
 R1T0458
 R1T0459
 R1T0460
 R1T0461
 R1T0462
 R1T0463
 R1T0464
 R1T0465
 R1T0466
 R1T0467
 R1T0468
 R1T0469
 R1T0470
 R1T0471
 R1T0472
 R1T0473
 R1T0474
 R1T0475
 R1T0476
 R1T0477
 R1T0478
 R1T0479
 R1T0480
 R1T0481
 R1T0482
 R1T0483
 R1T0484
 R1T0485
 R1T0486
 R1T0487
 R1T0488
 R1T0489
 R1T0490
 R1T0491
 R1T0492
 R1T0493
 R1T0494
 R1T0495
 R1T0496
 R1T0497
 R1T0498
 R1T0499
 R1T0500
 R1T0501
 R1T0502
 R1T0503
 R1T0504

EAMX(I) = EAMX(I) * C	R1T0505
EIN(I) = EIN(I) * C	R1T0506
EXIT(I) = EXIT(I) * C	R1T0507
320 CONTINUE	R1T0508
C	R1T0509
DO 340 I = 1, 17	R1T0510
DO 330 J = 1, 22	R1T0511
C	R1T0512
IF(TW1.EQ.TTT(J)) TW = J	R1T0513
IF(TW3.EQ.TTT(J)) TW = J	R1T0514
IF(TS1(I).EQ.TTT(J)) TS(I) = J	R1T0515
IF(TS3(I).EQ.TTT(J)) TS(I) = J	R1T0516
C	R1T0517
330 CONTINUE	R1T0518
340 CONTINUE	R1T0519
C	R1T0520
XFLAP = FLAP	R1T0521
XSLAT = SLAT	R1T0522
FLAP = 0.0	R1T0523
SLAT = 0.0	R1T0524
C	R1T0525
DO 350 I = 1, 5	R1T0526
IF(XFLAP.EQ.W(I)) FLAP = I	R1T0527
J = I + 5	R1T0528
IF(XSLAT.EQ.W(J)) SLAT = I	R1T0529
350 CONTINUE	R1T0530
C	R1T0531
1000 FORMAT(6F10.0)	R1T0532
1001 FORMAT(12I5)	R1T0533
1002 FORMAT(F10.0, A3, F3.0, F4.0, 4F10.0)	R1T0534
1003 FORMAT(3I5, 5X, 40I1)	R1T0535
1004 FORMAT(A3,A1,A3,3X, 5F10.0)	R1T0536
1005 FORMAT(A4,A10)	R1T0537
1006 FORMAT(20I1, 4I5)	R1T0538
1007 FORMAT(A4, 16X, A4, 16X, 4I5)	R1T0539
1008 FORMAT(6F10.0,F6.0)	R1T0540
1009 FORMAT(115/(6F10.0))	R1T0541
C	R1T0542
2000 FORMAT(1H1, * PROBLEM INPUT PARAMETERS*//5X, *NBDAYS =*,12,	R1T0543
1 5X,*NNACS =*,12,5X,*NSURFS =*,12,5X,*NHT =*,12,5X,	R1T0544
2 *NVT =*,12,5X,*ISWP =*,12,5X,*NPNLS =*,12 /)	R1T0545
2001 FORMAT(5X,*SREF =*,F9.3,* SQ.FT.*,5X,*CMAC =*,F9.3,* IN.*,	R1T0546
1 5X,*FUS. STA. C.G. =*F9.3,* IN.*,5X,*ZCG =*F8.3,* IN.*//)	R1T0547
2002 FORMAT(T3,*NO.*, T10,*BLEN*, T20,*BWID*,T30,*BHGT*, T40,*BAWET*,	R1T0548
1 T50,*BAMX*,T60,*BABS*,T70,*BLNS*,T80,*BLBT*,T90,*BASE*,	R1T0549
2 T100,*BQ*, T110,*BND* /)	R1T0550
2003 FORMAT(I5, 11F10.3)	R1T0551
2004 FORMAT(/T3,*N*, T10,*ELEN*, T20,*EWID*,T30,*EHGT*, T40,*EAWET*,	R1T0552
1 T50,*EAMX*,T60,*EIN*, T70,*EXIT*,T80,*ELNS*,T90,*ELBT*,	R1T0553
2 T100,*EQF*,T110,*ENO* /)	R1T0554
2005 FORMAT(/5X, A4, A10, 10X, 5H*****, /1H1)	R1T0555
2006 FORMAT(/** ASPECT RATIO =*F7.3, 5X,*TAPER RATIO =*F6.4,	R1T0556
1 5X,*L.E.SWEEP =*F7.3, 5X,*PLANFORM AREA =*F8.2,* SQ.FT.*//	R1T0557
2 5X,*TWIST =*F7.3,* DEG.*, 5X,*INCIDENCE =*F7.3,* DEG.*)	R1T0558
2007 FORMAT(2X,A3,A1,A3, 11F10.3)	R1T0559
2008 FOPMAT(5X, *AIRFOIL TYPE =*,2X,A3,A1,A3,5X,*CAMBER =*F7.4,5X,	R1T0560

1	*THICKNESS =*F8.4 /5X, *WING LOCATION (X,Y) = *,	R1T0561
2	2F12.4,*FUSLAGE INTERSECTION =*F9.4 /5X,	R1T0562
3	*MAX. THICKNESS SWP. =*F7.3,5X,*CBAR =*F8.4,5X,	R1T0563
4	*AWET =*F8.3 //)	R1T0564
2009	FORMAT(/ 5X,*AIRFOIL TYPE =*,2X,A3,A1,A3,5X,*TWIST =*F7.3,* DEG.*,	R1T0565
1	5X,*INCIDENCE =*F7.3,* DEG.*,5X,*MAX. THICKNESS SWP. =*,	R1T0566
2	F7.3,* DEG.*)	R1T0567
2010	FORMAT(5X,*VARIABLE GEOM. WING PLANFORM AREA =*F9.3,* SQ.FT.*,	R1T0568
1	5X,*TAPER RATIO =*F6.4,5X,*L.E.SWEEP =*F7.3,* DEG.* /)	R1T0569
2011	FORMAT(/5X,*WING PANEL GEOMETRY*/T3,*NO.*,T10,*CAMBER*, T20,	R1T0570
1	*T/C*, T30,*XLE*, T40,*YW*, T50,*CRW*, T60,*CBAR*,	R1T0571
2	T70,*AWET* /)	R1T0572
2012	FORMAT(25X,3F10.3)	R1T0573
2013	FORMAT(/5X,*GEOMETRY FOR ADDITIONAL AIRFOIL SURFACES* /	R1T0574
1	T2,*AIRFOIL *, T15,*CAMBER*, T25,*T/C*, T35,*SMTSW*,	R1T0575
2	T45,*SHF*, T55,*L.E.SWP*, T65,*T.E.SWP*, T75,*TAPER*,	R1T0576
3	T85,*CR*, T95,*SBAR*, T105,*SAWET* /)	R1T0577
2014	FORMAT(/5X, 15,* DRAG POLARS TO BE GENERATED AT THE FOLLOWING CON	R1T0578
1	DITIONS* /T5, *MACH NO.*, T20,*ALTITUDE*, T35,*TAIL SETTING*,	R1T0579
2	T53,*L.E.SWEEP*, T70,*FROM CL*, T85,*TO CL* /)	R1T0580
2015	FORMAT(F15.4, 5F15.3)	R1T0581
2016	FORMAT(F15.4, F15.3,15H TRIMMED, 3F15.3)	R1T0582
C		R1T0583
C		R1T0584
		R1T0585

END

CC = 00361

	OVERLAY(1,2)		R1T0587
	PROGRAM NINPT		R1T0588
C			R1T0589
C	READS INPUT WITH A NAMELIST FORMAT		R1T0590
C			R1T0591
	COMMON /BLKOV1/ IJ		R1T0592
	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS		R1T0593
	COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,		R1T0594
1	BLEN(10), BWID(10), BHGT(10), BAWET(10), BQ(10),		R1T0595
2	BND(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),		R1T0596
3	BASE(10), ELEN(10), EWID(10), EHGT(10),		R1T0597
4	EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),		R1T0598
5	ELBT(10), EQF(10), END(10), CBAR(10), TW,		R1T0599
5	XLEW, YWW, YB, CR, BO2, BFUS, FMISC, AB, AFTAW,		R1T0600
6	CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,		R1T0601
7	TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),		R1T0602
8	XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,		R1T0603
9	SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),		R1T0604
1	SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),		R1T0605
2	SCR(10), HTLE, HTY, HTZ, HTINC		R1T0606
	COMMON /BLKA03/ ROUGHK, CLE(3), CCR(3), YC(3), XCG, ZCG, CMAC, IREF, A1(6)		R1T0607
	COMMON /BLKA04/ IHLS, NF, NS, FLAP, SLAT, DF(5), CLOC(5),		R1T0608
1	DCLOF(5), DCLMF(5), DCDF(5), DCMDF(5),		R1T0609
2	DS(5), CSOC(5), DCLOS(5), DCLMS(5), DCDS(5),		R1T0610
3	DCMOS(5), BF11, BF10, CF1(3), BF21, BF20, CF2(3),		R1T0611
4	BS11, BS10, CS1, BS21, BS20, CS2		R1T0612
C			R1T0613
C			R1T0614
	COMMON /BLKSUR/ NSURV, NCLAS, IT(20), ITRM(20), FMSURV(20),		R1T0615
1	ALT(20), DHSV(20), SWPV(20), CLLO(20), CLHI(20),		R1T0616
2	TRB(5,10), TRN(5,10), TRU(5,10), TRL(5,10),		R1T0617
3	TRS(5,10)		R1T0618
	COMMON /BLKHLS/ NHLSV, DF1(3,5), CPF(5), DSI(5), CPS(5),		R1T0619
1	DELCD(5), H(5), DF12(3,5), CPF2(5), DSI2(5),		R1T0620
2	CPS2(5)		R1T0621
C			R1T0622
	COMMON /BLKADJ/ IVAL(20), X(15), YM(15,9), YA(15,9), XCL(15),		R1T0623
1	NXVAR, NXCL		R1T0624
	COMMON /BLKTIL/ TITLE(6)		R1T0625
	COMMON /BLKPRT/ KPRINT(50)		R1T0626
C			R1T0627
	DIMENSION L(9)		R1T0628
C			R1T0629
	NAMELIST /BDYS/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS, CMAC, IREF,		R1T0630
1	SREF, ROUGHK, XCG, ZCG, BFUS, AB, FMISC, BLEN, BWID, BHGT,		R1T0631
2	BAWET, BQ, BND, BAMX, BABS, BLNS, BLBT, BASE/		R1T0632
3	NACEL/ NNACS, SREF, ELEN, EWID, EHGT, EAWET, EQF, END,		R1T0633
4	EAMX, EIN, EXIT, ELNS, ELBT/		R1T0634
5	WING/ NSURFS, ISWP, NPNLS, AR, TAPR, SWPLE, SPLAN, TWIST,		R1T0635
6	WINC, TW, CAM, TOC, XLE, YW, CRW, CBAR, AWET, SWMT, CMAC,		R1T0636
7	XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC, SREF, AFTAW,		R1T0637
8	CLE, YC, CCR, XLEW, YWW, YB, CR, BO2, XCG, CONCL/		R1T0638
9	SURFS/ NSURFS, NHT, NVT, SREF, SBAR, TS, SCAM, STOC, SAWET,		R1T0639
10	SMTSW, SHF, SWL, SWT, STAPR, SCR, HTLE, HTY, HTZ, HTINC/		R1T0640
11	SURV/ NSURV, NCLAS, NHLSV, IT, ITRM, FMSURV, ALT, DHSV, SWPV,		R1T0641
A	CLLO, CLHI, YRB, TRN, TRU, TRL, TRS, DF1, CPF, DSI, CPS,		R1T0642

B	DELCO,H,DFI2,CPF2,DSI2,CPS2,KPRINT/	RIT0643
C	STOL/ IHLS,NF,NS,FLAP,SLAT,DF,CFOC,DCLOF,DCLMF,DCDF,	RIT0644
D	DCMUF,DS,CSOC,DCLOS,DCLMS,DCDS,DCMCS,BF11,	RIT0645
E	BF10,CF1,BF21,BF20,CF2,BS11,BS10,CS1,BS21,	RIT0646
F	BS20,CS2,H,DFI2,CPF2,DSI2,CPS2/	RIT0647
C	ADJUST/ IVAL, NXVAR, NXCL, X, XCL, YM, YA	RIT0648
C		RIT0649
	DO 100 I = 1,9	RIT0650
	L(I) = 1	RIT0651
	100 CONTINUE	RIT0652
C		RIT0653
	CALL CONV	RIT0654
	IF(IJ.EQ.1) GO TO 200	RIT0655
	READ(5,1000) (TITLE(I), I = 1,6)	RIT0656
	READ(5,1001) (L(I), I = 1,9)	RIT0657
C		RIT0658
	200 IF(L(1).EQ.1) READ(5,BODYS)	RIT0659
	IF(L(2).EQ.1) READ(5,NACEL)	RIT0660
	IF(L(3).EQ.1) READ(5,WING)	RIT0661
	IF(L(4).EQ.1) READ(5,SURFS)	RIT0662
	IF(L(5).EQ.1) READ(5,SURV)	RIT0663
	IF(L(6).EQ.1) READ(5,STOL)	RIT0664
	IF(L(7).EQ.1) READ(5,ADJUST)	RIT0665
C		RIT0666
	WRITE(6,2000)	RIT0667
C		RIT0668
	CALL CONV1	RIT0669
C		RIT0670
	350 CONTINUE	RIT0671
	1000 FORMAT(6A10)	RIT0672
	1001 FORMAT(50I1)	RIT0673
	2000 FORMAT(5X,*NAMELIST INPUT COMPLETED*)	RIT0674
C		RIT0675
	END	RIT0676

CC = 00090


```

SUBROUTINE CONV
C
C CONVERTS INPUT DATA TO FEET AND RADIAN UNITS
C
COMMON /BLKA02/ A1(433)
COMMON /BLKA03/ A2(20)
C
DIMENSION B1(433), B2(20)
C
DO 100 I = 1, 20
100 B2(I) = A2(I)
C
DO 200 I = 1, 433
200 B1(I) = A1(I)
C
RETURN
C
ENTRY CONV1
C
A      = 1./57.2957796
B      = 1./12.
C      = 1./144.
C
IF( A1(4).NE.B1(4) ) A1(4) = A1(4) * A
IF( A1(275).NE.B1(275) ) A1(275) = A1(275) * A
IF( A1(317).NE.B1(317) ) A1(317) = A1(317) * A
DO 300 I = 1, 10
IF( A1(369+I).NE.B1(369+I) ) A1(369+I) = A1(369+I)*A
IF( A1(389+I).NE.B1(389+I) ) A1(389+I) = A1(389+I)*A
IF( A1(399+I).NE.B1(399+I) ) A1(399+I) = A1(399+I)*A
300 CONTINUE
C
DO 400 I = 1, 10
IF( A1(224+I).NE.B1(224+I) ) A1(224+I) = A1(224+I) * B
IF( A1(319+I).NE.B1(319+I) ) A1(319+I) = A1(319+I) * B
IF( A1(419+I).NE.B1(419+I) ) A1(419+I) = A1(419+I) * B
IF( A1(104+I).NE.B1(104+I) ) A1(104+I) = A1(104+I) * C
IF( A2(1+I).NE.B2(1+I) ) A2(1+I) = A2(1+I) * B
400 CONTINUE
C
DO 500 I = 1, 20
IF( A1(84 +I).NE.B1(84 +I) ) A1(84 +I) = A1(84 +I) * B
IF( A1(184+I).NE.B1(184+I) ) A1(184+I) = A1(184+I) * B
IF( A1(64 +I).NE.B1(64 +I) ) A1(64 +I) = A1(64 +I) * C
500 CONTINUE
C
DO 600 I = 1, 30
IF( A1(4 + I).NE.B1(4 + I) ) A1(4 + I) = A1(4 + I) * B
IF( A1(114+I).NE.B1(114+I) ) A1(114+I) = A1(114+I) * B
IF( A1(280+I).NE.B1(280+I) ) A1(280+I) = A1(280+I) * B
IF( A1(154+I).NE.B1(154+I) ) A1(154+I) = A1(154+I) * C
600 CONTINUE
C
IF( A1(314).NE.B1(314) ) A1(314) = A1(314) * B
IF( A1(315).NE.B1(315) ) A1(315) = A1(315) * B
IF( A1(316).NE.B1(316) ) A1(316) = A1(316) * B

```

```

R1T0678
R1T0679
R1T0680
R1T0681
R1T0682
R1T0683
R1T0684
R1T0685
R1T0686
R1T0687
R1T0688
R1T0689
R1T0690
R1T0691
R1T0692
R1T0693
R1T0694
R1T0695
R1T0696
R1T0697
R1T0698
R1T0699
R1T0700
R1T0701
R1T0702
R1T0703
R1T0704
R1T0705
R1T0706
R1T0707
R1T0708
R1T0709
R1T0710
R1T0711
R1T0712
R1T0713
R1T0714
R1T0715
R1T0716
R1T0717
R1T0718
R1T0719
R1T0720
R1T0721
R1T0722
R1T0723
R1T0724
R1T0725
R1T0726
R1T0727
R1T0728
R1T0729
R1T0730
R1T0731
R1T0732
R1T0733

```

```

IF( A1(318).NE.B1(318) ) A1(318) = A1(318) * B
IF( A1(430).NE.B1(430) ) A1(430) = A1(430) * B
IF( A1(431).NE.B1(431) ) A1(431) = A1(431) * B
IF( A1(432).NE.B1(432) ) A1(432) = A1(432) * B
IF( A1(236).NE.B1(236) ) A1(236) = A1(236) * B
IF( A1(237).NE.B1(237) ) A1(237) = A1(237) * B
IF( A1(238).NE.B1(238) ) A1(238) = A1(238) * B
IF( A1(239).NE.B1(239) ) A1(239) = A1(239) * B
IF( A1(240).NE.B1(240) ) A1(240) = A1(240) * B
IF( A1(241).NE.B1(241) ) A1(241) = A1(241) * B
IF( A1(291).NE.B1(291) ) A1(291) = A1(291) * B
IF( A1(302).NE.B1(302) ) A1(302) = A1(302) * B
IF( A1(313).NE.B1(313) ) A1(313) = A1(313) * B
IF( A2(12).NE.B2(12) ) A2(12) = A2(12) * B
IF( A2(13).NE.B2(13) ) A2(13) = A2(13) * B

```

```

R1T0734
R1T0735
R1T0736
R1T0737
R1T0738
R1T0739
R1T0740
R1T0741
R1T0742
R1T0743
R1T0744
R1T0745
R1T0746
R1T0747
R1T0748
R1T0749
R1T0750
R1T0751

```

C

```

RETURN
END

```

CC = 00074

	OVERLAY(2,0)		R1T0753
	PROGRAM GEOM		R1T0754
C			R1T0755
C	COMPUTES GEOMETRIC PARAMETERS FOR AERO CALCULATIONS		R1T0756
C			R1T0757
	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS		R1T0758
	COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,		R1T0759
1	BLEN(10), BWID(10), BHGT(10), BAWET(10), BG(10),		R1T0760
2	BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),		R1T0761
3	BASE(10), ELEN(10), EWID(10), EHGT(10),		R1T0762
4	EAWET(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),		R1T0763
5	ELBT(10), EQF(10), END(10), CBAR(10), TW,		R1T0764
5	XLEN, YWW, YBOD, CR, BO2, A2(4),		R1T0765
6	CAM(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,		R1T0766
7	TWIST, ETWIST, INC, XLE(11), CRW(11), YW(11),		R1T0767
8	XPIVOT, YPIVOT, XAPEX, AFTSK, AFTCB, AFTOC,		R1T0768
9	SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),		R1T0769
1	SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),		R1T0770
2	SCR(10), HTLE, HTY, HTZ, HTINC		R1T0771
	COMMON /BLKA03/ ROUGHK, CLE(3),CCR(3),YC(3), XCG,ZCG,CMAC,A1(7)		R1T0772
C			R1T0773
	COMMON /BLKG01/ FRBOD(10), ARS(10), SSEX(10), FRNAC(10), SWP,		R1T0774
1	SWPQC, SWPMC, SWPTE, DOB, TOCW, CLO, SEXW, ESWMC,		R1T0775
2	SXX(10), XLFSWX(10), XMCSWX(10), ARW, TR, WPLAN,		R1T0776
3	CB, YB, XB,		R1T0777
4	CRX, CBX, CTX, YIX, VOX, SIX, SCX, ARI, ARXR,		R1T0778
5	CBXP, AROP, SOXP, SWPLEI, SWPLEO, SWPMCI, SWPMCO,		R1T0779
6	ESWQC, FSWLE, ESWTE, CROB, DXOB, XHT, XCRTE,		R1T0780
7	KH, OMEGA, FOC, TWST, DINC, G1(5), DXQC, G3(81)		R1T0781
C			R1T0782
	COMMON /BLKPRT/ KPRINT(50)		R1T0783
C			R1T0784
	DIMENSION FOCX(22)		R1T0785
	DIMENSION G2(200)		R1T0786
	EQUIVALENCE (G2(1), FRBOD(1))		R1T0787
	DATA FOCX / 4*0.05515,3*0.06651,15*0.079 /		R1T0788
C			R1T0789
	TWST = TWIST		R1T0790
	DINC = WINC		R1T0791
C			R1T0792
	DO 100 I = 1, NBDYS		R1T0793
	BDIA2 = BWID(I) * BHGT(I)		R1T0794
	FRBOD(I) = 0.0		R1T0795
	IF(BAMX(I).EQ.0.0) BAMX(I) = 0.7854 * BDIA2		R1T0796
	IF(BAWET(I).EQ.0.0) BAWET(I) = (2.8*BLNS(I) +2.5*BLBT(I)		R1T0797
1	*(1.+SQRT(BABS(I)/BAMX(I))) + (BLEN(I) -BLNS(I) -BLBT(I))		R1T0798
2	*4.) * SQRT(.7854 * BAMX(I)) * BNO(I)		R1T0799
	IF(BDIA2.GT.0.0) FRBOD(I) = BLEN(I)/SQRT(BDIA2)		R1T0800
100	CONTINUE		R1T0801
C			R1T0802
	DO 200 I = 1, NNACS		R1T0803
	EDIA2 = EWID(I) * EHGT(I)		R1T0804
	EPNAC(I) = 0.0		R1T0805
	IF(FAMX(I).EQ.0.0) FAMX(I) = 0.7854 * EDIA2		R1T0806
	IF(EAWET(I).EQ.0.0) EAWET(I) = (2.5*ELNS(I) * (1. +		R1T0807
1	SQRT(EIN(I)/EAMX(I))) + 2.5 *ELBT(I)		R1T0808

1	*(1.+SQRT(ENIT(I)/EAMX(I))) + (ELEN(I) - ELNS(I) - ELBT(I))	R1T0809
2	*4.) * SQRT(.7854 * BAMX(I)) * ENO(I)	R1T0810
	IF(EDIA2.GT.0.0) FRNAC(I) = ELEN(I)/SQRT(EDIA2)	R1T0811
200	CONTINUE	R1T0812
C		R1T0813
210	NP = NPNLS + 1	R1T0814
	IF(NPNLS.NE.1) GO TO 220	R1T0815
	TSWP = TAN(SWPLE)	R1T0816
	CO = 4./AR * (1.-TAPR)/(1.+TAPR)	R1T0817
C		R1T0818
C	GEOMETRY FOR SIMPLE TRAPEZOIDAL WINGS IS CALCULATED HERE	R1T0819
C		R1T0820
	SWP = SWPLE	R1T0821
	SWPQC = ATAN(TSWP - CO * 0.25)	R1T0822
	SWPMC = ATAN(TSWP - CO * 0.50)	R1T0823
	SWPTE = ATAN(TSWP - CO)	R1T0824
	BO2 = 0.5 * SQRT(SPLAN * AR)	R1T0825
	YW(1) = YBOD	R1T0826
	XLE(1) = XLEW + (YBOD - YWW) * TSWP	R1T0827
	DOB = YW(1)/BO2	R1T0828
C		R1T0829
	CR = SPLAN/(BO2*(1.+TAPR))	R1T0830
	XCRTF = XLEW - YWW * TSWP + CR	R1T0831
	CRX = CR * (1. - DOB*(1.-TAPR))	R1T0832
	CTX = CR * TAPR	R1T0833
	IF(CBAR(1).EQ.0.0) CBAR(1) = (CRX + CTX**2/(CRX+CTX))*0.66667	R1T0834
	YIX = BO2 * (1.-DOB)	R1T0835
	CB = 0.66667 * CR * (1. + TAPR*TAPR/(1. + TAPR))	R1T0836
	YB = BO2/3. * (1. + 2.*TAPR)/(1. + TAPR)	R1T0837
	XB = YB * TSWP + XLE(1) - YW(1)*TSWP	R1T0838
C		R1T0839
	ESWMC = SWPMC	R1T0840
	ESWQC = SWPQC	R1T0841
	ESWLE = SWPLE	R1T0842
	ESWTE = SWPTE	R1T0843
	TOCW = TOC(1)	R1T0844
	CLO = CAM(1)	R1T0845
	TR = TAPR	R1T0846
	SEXW = SPLAN/(1.+TR) * (2.-(1.-TR)*(YW(1)/BO2 + 1.))	R1T0847
1	* (1.-YW(1)/BO2)	R1T0848
	SXX(1) = SEXW	R1T0849
	IF(AWET(1).EQ.0.0) AWET(1) = SEXW * (2. + .1843*TOCW	R1T0850
1	+1.5268*TOCW**2 -.8395*TOCW**3)	R1T0851
	XLESWX(1) = SWPLE	R1T0852
	XMCSWX(1) = SWPMC	R1T0853
	ARW = AR	R1T0854
	WPLAN = SPLAN	R1T0855
	SIX = SEXW	R1T0856
	ARXR = 4.* YIX**2/SIX	R1T0857
	SWPLEI = SWP	R1T0858
	SWPMCI = SWPMC	R1T0859
C		R1T0860
	YW(2) = BO2	R1T0861
	XLE(2) = XLE(1) + (YW(2)-YW(1)) * TSWP	R1T0862
	CRW(1) = CRX	R1T0863
	CRW(2) = CTX	R1T0864

	CRQB = 0.5 * CR/BO2	R1T0865
	DXQB = CRQB - 0.75 * TAN(SWPQC)	R1T0866
C		R1T0867
220	IF(NPNLS.GT.2) GO TO 240	R1T0868
	DO 230 I = 1, NP	R1T0869
	CLE(I) = XLE(I)	R1T0870
	CCR(I) = CRW(I)	R1T0871
	YC(I) = YW(I)	R1T0872
230	CONTINUE	R1T0873
C		R1T0874
240	IF(NPNLS.LE.1) GO TO 310	R1T0875
C	GEOMETRY FOR MULTIPLE PANEL WINGS IS CALCULATED HERE	R1T0876
C		R1T0877
	WPLAN = SPLAN	R1T0878
	BO2 = YW(NP)	R1T0879
	TR = TAPR	R1T0880
	SWP = SWPLE	R1T0881
	ETA = YW(1)/YW(2)	R1T0882
	CRCL = (CRW(1) - ETA * CRW(2)) / (1.-ETA)	R1T0883
	CRQB = 0.5 * CRCL/BO2	R1T0884
	DOB = YW(1)/BO2	R1T0885
	SFXW = 0.0	R1T0886
	TDCW = 0.0	R1T0887
	CLD = 0.0	R1T0888
	COSMC = 0.0	R1T0889
	COSQC = 0.0	R1T0890
	TANLE = 0.0	R1T0891
	TANTE = 0.0	R1T0892
	SUMCB = 0.0	R1T0893
	SUMYB = 0.0	R1T0894
	SUMXB = 0.0	R1T0895
C		R1T0896
	DO 300 I = 1, NPNLS	R1T0897
	DYW = YW(I+1) - YW(I)	R1T0898
	SX = (CRW(I+1) + CRW(I)) * DYW	R1T0899
	DXLE = -XLE(I) + XLE(I+1)	R1T0900
	XLESW = ATAN(DXLE/DYW)	R1T0901
	DXTE = DXLE + (CRW(I+1) - CRW(I))	R1T0902
	IF(I.EQ.1) XCRTS = XLE(1) + YW(1)*TAN(XLESW) + CRCL	R1T0903
	DX = DXLE + 0.5 * (CRW(I+1) - CRW(I))	R1T0904
	DXQ = DXLE + 0.25 * (CRW(I+1) - CRW(I))	R1T0905
	XTESW = ATAN(DXTE/DYW)	R1T0906
	XMCSW = ATAN(DX/DYW)	R1T0907
	XQCSW = ATAN(DXQ/DYW)	R1T0908
C		R1T0909
	CR = CRW(I)	R1T0910
	CT = CRW(I+1)	R1T0911
	T = CT/CR	R1T0912
C		R1T0913
	SXX(I) = SX	R1T0914
	XLESWX(I) = XLESW	R1T0915
	XMCSWX(I) = XMCSW	R1T0916
C		R1T0917
	CB = 0.66667 * CR * (1. + T*T/(1.+T))	R1T0918
	IF(CBAR(I).EQ.0.0) CBAR(I) = CB	R1T0919
	YB = 0.33333 * DYW * (1.+2.*T)/(1.+T) + YW(I)	R1T0920

C	XB = XLE(I) - XLF(I) + (YB - YW(I))*TAN(XLESW)	R1T0921
		R1T0922
	DYX1 = 0.75*BO2 - YW(I)	R1T0923
	DYX2 = 0.75*BO2 - YW(I+1)	R1T0924
	IF(DYX1.GE.0.0.AND.DYX2.LT.0.0) X1 = XLE(I) + CRW(I)*0.25	R1T0925
1	+ DYX1 * TAN(XQCSW)	R1T0926
C		R1T0927
C		R1T0928
	IF(AWET(I).EQ.0.0) AWET(I) = SX * (2. +.1843*TOC(I)	R1T0929
1	+1.5268*TOC(I)**2 -.8345*TOC(I)**3)	R1T0930
	TOCW = TOCW + SX * TOC(I)**2	R1T0931
	COSMC = COSMC + COS(XMCSW) * SX	R1T0932
	COSQC = COSQC + COS(XQCSW) * SX	R1T0933
	TANLE = TANLE + TAN(XLESW) * SX	R1T0934
	TANTE = TANTE + TAN(XTESW) * SX	R1T0935
	CLD = CLD + CAM(I)**2 * SX	R1T0936
	SEFXW = SEXW + SX	R1T0937
C		R1T0938
	SUMCB = SUMCB + CB*SX	R1T0939
	SUMYB = SUMYB + YB*SX	R1T0940
	SUMXB = SUMXB + XB*SX	R1T0941
C		R1T0942
300	CONTINUE	R1T0943
C		R1T0944
	TOCW = SQRT(TOCW/SEXW)	R1T0945
	CLD = SQRT(CLD/SEXW)	R1T0946
	ESWMC = ACOS(COSMC/SEXW)	R1T0947
	ESWQC = ACOS(COSQC/SEXW)	R1T0948
	ESWLF = ATAN(TANLE/SEXW)	R1T0949
	ESWTE = ATAN(TANTE/SEXW)	R1T0950
C		R1T0951
	CB = SUMCB/SEXW	R1T0952
	YB = SUMYB/SEXW	R1T0953
	XB = SUMXB/SEXW + XLE(I)	R1T0954
	X2 = XLF(I) - YW(I)*TAN(XLESWX(I)) + 0.75 * CRCL	R1T0955
	DXOB = (X2 - X1)/BO2	R1T0956
C		R1T0957
	IF(WPLAN.EQ.0.0) WPLAN = SEXW + (CRCL + CRW(I)) * YW(I)	R1T0958
	ARW = 4. * BO2**2/WPLAN	R1T0959
	TRK = (TAN(ESWLE) - TAN(ESWTE)) * 0.25 * ARW	R1T0960
	IF(ISWP.EQ.0) TR = (1.-TRK)/(1.+TRK)	R1T0961
	TRLIM = CRW(NP)/CRCL	R1T0962
	IF(TR.LT.TRLIM) TR = TRLIM	R1T0963
C		T0964
310	IF(NSURFS.LE.1) GO TO 410	R1T0965
	XNVT = NVT	R1T0966
	DO 400 J = 2, NSURFS	R1T0967
	I = J-1	R1T0968
C		R1T0969
	SSFX(I) = SAWET(I)*0.5	R1T0970
	IF(SCR(I).LE.0.0) GO TO 400	R1T0971
	ARS(I) = SSFX(I)/SCR(I)**2	R1T0972
C		R1T0973
	IF(SWL(I).EQ.SWT(I)) GO TO 400	R1T0974
	ARS(I) = 4.*(1.-STAPR(I))/(1.+STAPR(I))/(TAN(SWL(I))-TAN(SWT(I)))	R1T0975
	SSFX(I) = (SCR(I) * (1.+STAPR(I)))**2 * ARS(I) * 0.25	R1T0976

	IF(NHT.EQ.0.AND.NVT.GT.0) SSEX(1) = 0.5 * XNVT * SSEX(1)	R1T0977
	IF(NHT.EQ.1.AND.NVT.GT.0) SSEX(2) = 0.5 * XNVT * SSEX(2)	R1T0978
C		R1T0979
	IF(SAWET(1).EQ.0.0) SAWET(1) = SSEX(1) * (2. +.1843*STOC(1)	R1T0980
1	+1.5268*STOC(1)**2 -.8395*STOC(1)**3)	R1T0981
	IF(SBAR(1).EQ.0.0) SBAR(1) = 0.66667 * SCR(1) *	R1T0982
1	(1.+STAPR(1)**2/(1.+STAPR(1)))	R1T0983
C		R1T0984
	400 CONTINUE	R1T0985
C		R1T0986
	410 IF(XCG.EQ.0.0) XCG = XB + 0.25 * CB	R1T0987
	IF(CMAC.EQ.0.0) CMAC = CB	R1T0988
	NI = TW	R1T0989
	FOC = CLD * FOCX(NI)	R1T0990
	DXQC = XLE(1+NPNLS)+CRW +NPNLS)*0.25 -XLE(1)-CRW(1)*0.25	R1T0991
C		R1T0992
C	GEOMETRY FOR THE HORIZONTAL TAIL IS CALCULATED	R1T0993
C		R1T0994
	XH = 0.0	R1T0995
	OMEGA = 0.0	R1T0996
C		R1T0997
	IF(NHT.EQ.0) GO TO 415	R1T0998
	BHT = SQRT(ARS(1) * SSEX(1))	R1T0999
	XHT = BHT/6. * (1. + 2.*STAPR(1))/(1.+ STAPR(1))*TAN(SWL(1))	R1T1000
1	+ SBAR(1) * 0.25 + HTLE	R1T1001
C		R1T1002
	XH = SQRT((HTZ-ZCG)**2 + (XHT-XCG)**2)	R1T1003
	OMEGA = ATAN((HTZ-XCG)/(XHT-XCG))	R1T1004
C		R1T1005
	415 IF(NPNLS.LE.1) GO TO 420	R1T1006
C		R1T1007
C	CRANKED WING GEOMETRY IS CALCULATED	R1T1008
C		R1T1009
	CRX = CCR(1)	R1T1010
	CBX = CCR(2)	R1T1011
	CTX = CCR(3)	R1T1012
	YIX = YC(2) - YC(1)	R1T1013
	YOX = YC(3) - YC(2)	R1T1014
C		R1T1015
	SIX = (CRX + CBX) * YIX	R1T1016
	SOX = (CBX + CTX) * YOX	R1T1017
	API = 4. * YIX**2 / SIX	R1T1018
	ARXR = 4. * (YIX+YOX)**2 / (SIX+SOX)	R1T1019
C		R1T1020
	DYP = YIX * 0.5	R1T1021
	CHXP = CTX + (YOX + DYP) * (CBX-CTX)/YOX	R1T1022
	SOXP = (CBXP + CTX) * (YOX + DYP)	R1T1023
	AROP = 4. * (YOX + DYP)**2 / SOXP	R1T1024
C		R1T1025
	SWPLFI = ATAN((CLE(2)-CLE(1))/YIX)	R1T1026
	SWPLFO = ATAN((CLE(3)-CLE(2))/YOX)	R1T1027
	SWPMCI = ATAN((CLE(2)-CLE(1)+(CBX-CRX)*0.5)/YIX)	R1T1028
	SWPMCO = ATAN((CLE(3)-CLE(2)+(CTX-CBX)*0.5)/YOX)	R1T1029
C		R1T1030
	420 IF(KPRINT(12).EQ.0) GO TO 421	R1T1031
	WRITE(6,1000) (G2(I), I = 1, 111)	R1T1032

C	421 WRITE(6,2000)	RIT1033
	IF(NBDYS.GT.0) WRITE(6,2001) (I,BLEN(I),BAWET(I),FRBCD(I),	RIT1034
	1 BQ(I), I = 1, NBDYS)	RIT1035
C		RIT1036
	IF(NNACS.GT.0) WRITE(6,2002) (I,ELEN(I),EAWET(I),FRNAC(I),	RIT1037
	1 EQF(I), I = 1,NNACS)	RIT1038
C		RIT1039
	IF(NSURFS.EQ.0) GO TO 450	RIT1040
	X = SWMT * 57.2958	RIT1041
	DO 425 I = 1, NPNLS	RIT1042
	IF(I.NE.NPNLS) WRITE(6,2003) I, CBAR(I), AWT(I), TOC(I)	RIT1043
	425 IF(I.EQ.NPNLS) WRITE(6,2003) I, CBAR(I), AWT(I), TOC(I), X	RIT1044
C		RIT1045
	NS = NSURFS - 1	RIT1046
	IF(NS.LE.0) GO TO 450	RIT1047
	DO 430 I = 1, NS	RIT1048
	J = I + 1	RIT1049
	X = SMTSW(I) * 57.2958	RIT1050
	WRITE(6,2004) J, SBAR(I), SAWET(I), STOC(I), SHF(I), X	RIT1051
	430 CONTINUE	RIT1052
	450 CONTINUE	RIT1053
	1000 FORMAT(5X, *GEOMETRY DATA DUMP, COMMON BLKG01*/(1X,10F12.4))	RIT1054
	2000 FORMAT(1H1, 5X, *CONFIGURATION SUMMARY* ///	RIT1055
	1 31X, *LENGTH*, 7X, *WETTED AREA*, 5X, *FR CR T/C*,	RIT1056
	2 4X, *INTERFERENCE*, 4X, *MAX. T/C SWEEP* /	RIT1057
	3 30X, * (FT.)*, 8X, * (SQ. FT.) *, 20X,	RIT1058
	4 *FACTOR*, 8X, * (DEG.) * /)	RIT1059
	2001 FORMAT(5X, *BODY NUMBER* I2, 5X, 4F15.5)	RIT1060
	2002 FORMAT(5X, *NACELLE NO.* I2, 5X, 4F15.5)	RIT1061
	2003 FORMAT(5X, *WING PANEL NO.*I2, 2X, 3F15.5, 15X, F15.5)	RIT1062
	2004 FORMAT(5X, *AIRFOIL SURFACE NO.*I2, F12.5, 4F15.5)	RIT1063
C		RIT1064
	END	RIT1065
		RIT1066

CC = 00314

	OVERLAY(3,0)	R1T1068
	PROGRAM SURVEY	R1T1069
C		R1T1070
C	AERODYNAMIC SURVEY	R1T1071
C		R1T1072
C	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NMT, NVT, ISWP, NPNLS	R1T1073
		R1T1074
	COMMON/BLKSUR/ NSURV, NCLAS, IT(20), ITRM(20), FMSURV(20),	R1T1075
1	ALT(20), DHSV(20), SWPV(20), CLLO(20), CLHI(20),	R1T1076
2	TRBNWS(250)	R1T1077
C		R1T1078
C	COMMON /BLKHLS/ NHLSV, HLS(70)	R1T1079
C		R1T1080
	COMMON /BLKTIL/ TITLE(6)	R1T1081
	COMMON /BLKOV3/ ITYP, XMACH, TOC, CLD, SWPMC, H, FM, ITL, JPASS,	R1T1082
1	ROT, OTE, ITRIM	R1T1083
C		R1T1084
	COMMON /BLKC01/ CL, CD, CM, ALPHA, CDM, CDL, CDR, CDRO, CLT, CDT,	R1T1085
1	DM, FK, DELCL, CMO, DCMCL, XACWB, CLA, ALO,	R1T1086
2	TCO(5), CDFUS(5), CDBOD(5), CONAC(4), CDWING(4),	R1T1087
3	CDHT(4), CDVT(4), CDSURF(4),	R1T1088
4	C1(9), CLPB, CLDB, CLMAX, C2(9), CDC, C3(10),	R1T1089
5	C4(7), CDMISC, DEDA, CDAFT, C5(5)	R1T1090
	COMMON /BLKA02/ A1(433)	R1T1091
	COMMON /BLKA03/ A2(13), IREF, A3(6)	R1T1092
	COMMON /BLKVGM/ KPASS, SWP, VGM(27)	R1T1093
	COMMON /BLKC03/ CO3(22)	R1T1094
C		R1T1095
	COMMON /BLKC01/ G1(200)	R1T1096
C		R1T1097
	COMMON /BLKBD2/ AA(22), BB(22), CC(22), DD(22), XT(22)	R1T1098
C		R1T1099
	DIMENSION XOTE(20), YOTE(20)	R1T1100
	DATA XOTE / 34.6,38.4,46.4,60.2,57.5,59.5,66.5,30.0,95.0,50.0,	R1T1101
1	63.0,82.8,113.0,153.0,63.0,82.8,113.0,63.0,82.8,113.0, /	R1T1102
2	YOTE / 4*14.8,3*14.05,40.0,0.0,11*13.8 /	R1T1103
C		R1T1104
	IF(ISWP.EQ.0) GO TO 100	R1T1105
	KPASS = 1	R1T1106
	SWP = A1(4)	R1T1107
	CALL OVERLAY(4HOVLY,3,1)	R1T1108
	KPASS = 2	R1T1109
	SWP = A1(317)	R1T1110
	CALL OVERLAY(4HOVLY,3,1)	R1T1111
100	KPASS = 0	R1T1112
C		R1T1113
	TOC = G1(46)	R1T1114
	CLD = G1(47)	R1T1115
	ITYP = A1(235)	R1T1116
	WINC = A1(280)	R1T1117
	IF(NSURV.EQ.0) GO TO 510	R1T1118
C		R1T1119
	SWP2 = SWPV(1)	R1T1120
C		R1T1121
	JPASS = 0	R1T1122
	DO 500 L = 1, NSURV	R1T1123

FM	= FMSURV(L)	R1T1124
H	= ALT(L)	R1T1125
DH	= DHSV(L)	R1T1126
SWEEP	= SWPV(L)	R1T1127
CLLOW	= CLLO(L)	R1T1128
ITL	= IT(L)	R1T1129
ITRIM	= ITRM(L)	R1T1130
C		R1T1131
XNCL	= NCLAS - 1	R1T1132
DCL	= 0.0	R1T1133
IF(XNCL.GT.0.0) DCL = (CLHI(L) - CLLO(L))/XNCL		R1T1134
C		R1T1135
SWP	= SWEEP/57.2957796	R1T1136
IF(ISWP.GT.0.AND.SWEEP.NE.SWP2) JPASS = 0		R1T1137
SWP2	= SWEEP	R1T1138
C		R1T1139
IF(ISWP.GT.0.AND.JPASS.EQ.0) CALL OVERLAY(4HOVLY,3,1)		R1T1140
EO2	= 0.25 * SQRT(G1(80)*G1(82))	R1T1141
DO 200 IP = 1, NPNLS		R1T1142
SWPMW	= G1(69+IP)	R1T1143
IF(ISWP.GT.0.AND.IP.EQ.NPNLS) GO 0 190		R1T1144
IF(A1(302+IP).LE.EO2.AND.A1(303+IP).GT.EO2) GO TO 210		R1T1145
GO TO 200		R1T1146
190 IF(A1(302+NPNLS).LE.EO2) SWPMW = G1(69+NPNLS)		R1T1147
200 CONTINUE		R1T1148
C		R1T1149
210 IF(SWPMW.GT.G1(49)) SWPMW = G1(49)		R1T1150
SWPMC	= SWPMW * 57.2958	R1T1151
TOE	= G1(46)	R1T1152
CLD	= G1(47)	R1T1153
ITYP	= A1(235)	R1T1154
XMACH	= 0.6	R1T1155
C		R1T1156
IF(JPASS.EQ.0) CALL OVERLAY(4HOVLY,3,2)		R1T1157
C		R1T1158
IF(JPASS.EQ.2) JPASS = 1		R1T1159
C		R1T1160
DO 300 M = 1, NCLAS		R1T1161
XMI	= M-1	R1T1162
CL	= CLLOW + DCL * XMI	R1T1163
C		R1T1164
CALL OVERLAY(4HOVLY,3,3,6HRECALL)		R1T1165
C		R1T1166
SWEEP	= G1(41) * 57.29578	R1T1167
IF(M.NE.1) GO TO 220		R1T1168
WRITE(6,1005) (TITLE(K), K=1,6)		R1T1169
IF(H.GE.0.0.AND.ITRIM.EQ.0) WRITE(6,1000) FM,H,SWEEP,DH		R1T1170
RN	= -H * 1.0E6	R1T1171
IF(H.LT.0.0.AND.ITRIM.EQ.0) WRITE(6,1006) FM,RN,SWEEP,DH		R1T1172
IF(H.GE.0.0.AND.ITRIM.NE.0) WRITE(6,1008) FM, H, SWEEP		R1T1173
IF(H.LT.0.0.AND.ITRIM.NE.0) WRITE(6,1009) FM,RN, SWEEP		R1T1174
220 ANGLE = ALPHA		R1T1175
IF(IREF.EQ.1) ANGLE = ALPHA - WINC		R1T1176
C		R1T1177
CDRA	= CDR + CDAFT	R1T11771
IF(ITRIM.EQ.0) WRITE(6,1001) CLT, CDT, CM, ANGLE, CDL, CDRA,		R1T1178

1	CL, CD	R1T1179
1	IF(ITRIM.EQ.1) WRITE(6,1001) CLT, CDT, CH, ANGLE, CDL, CDRA,	R1T1180
1	CL, CD	R1T1181
C		R1T1182
300	CONTINUE	R1T1183
C		R1T1184
	ALOR = ALO	R1T1185
	IF(IREF.EQ.1) ALOR = ALO - WINC	R1T1186
	WRITE(6,1002) CLA, ALOR, CLDB, FK, DELCL, CLMAX, CM0, DCMCL	R1T1187
C		R1T1188
	WRITE(6,1003)(TCD(J), CDFUS(J), CDBOD(J), CDMAC(J), CDWING(J),	R1T1189
1	CDHT(J), CDVT(J), CDSURF(J), J = 1, 4)	R1T1190
C		R1T1191
	WRITE(6,1004) TCD(5), CDFUS(5), CDBOD(5), CDC, CDRO, COMISC	R1T1192
	WRITE(6,1007) CDM	R1T1193
C		R1T1194
500	CONTINUE	R1T1195
C		R1T1196
510	IF(NHLSV.EQ.0) GO TO 700	R1T1197
	ROT = AA(ITYP)/TOC + BB(ITYP) + CC(ITYP)*TOC + DD(ITYP)*TOC**2	R1T1198
	IF(ITYP.EQ.8) ROT = 0.88216 * TOC**0.806	R1T1199
	OTE = XOTE(ITYP)*TOC + YOTE(ITYP)*CLD	R1T1200
C		R1T1201
		R1T1202
C		R1T1203
	IF(NSURV.GT.0) GO TO 600	R1T1204
	CO3(1) = 0.0	R1T1205
	DO 520 I = 1, 11	R1T1206
	IF(I.GT.1) CO3(I) = CO3(I-1) + 0.1	R1T1207
	CO3(11+1) = 0.9	R1T1208
520	CONTINUE	R1T1209
C		R1T1210
600	DH = 0.0	R1T1211
	FM = 0.2	R1T1212
	H = 0.0	R1T1213
	CL = 0.0	R1T1214
	JPASS = 0	R1T1215
	ITL = 0	R1T1216
	ITRIM = 0	R1T1217
C		R1T1218
	CALL OVERLAY(4HOVLY,3,3)	R1T1219
	CALL OVERLAY(4HOVLY,3,4)	R1T1220
C		R1T1221
700	CONTINUE	R1T1222
C		R1T1223
1000	FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* FT.*, 5X,	R1T1224
1	*SWEEP ANGLE =*F6.2,* DEG.*, 5X,*TAIL DEFL. (DH) =*F6.2,	R1T1225
2	* DEG.* //T10,*CL*, T22,*TOTAL CD*, T38,*CM*, T53,*ALPHA*,	R1T1226
3	T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =0*,	R1T1227
4	T113,*CD AT DH=0*, /)	R1T1228
1001	FORMAT(5X,F8.3,2F15.5, F15.2, 4F15.5)	R1T1229
1002	FORMAT(/10H CLA =,F8.5,1X,*PER DEG.*,10H ALN =,F8.5,1X,	R1T1230
1	*DEG.*, 29X,10H CLDB =,F8.5 /	R1T1231
1	10H K =,F8.5, 9X, 10H DELCL =,F8.5 ,	R1T1232
2	34X,10H CLMAX =,F8.5 /	R1T1233
2	10H CM0 =,F8.5, 9X, 10H CM/CL =,F8.4 //	R1T1234

```

3 * DRAG BREAKDOWN ----- *, 8X,*FUSLAGE*, 9X,*BODIES*, 7X,*NACELLES* R1T1235
4*,11X,*WING*, 5X,*HORIZ TAIL*, 6X,*VERT TAIL*, 7X,*SURFACES* ) R1T1236
1003 FORMAT(5X,10HFRICITION =,F7.5,7F15.5 / R1T1237
1      5X,10HFORM =,F7.5,7F15.5 / R1T1238
2      5X,10HINTERF =,F7.5,7F15.5 / R1T1239
3      5X,10HWAVE =,F7.5,7F15.5 ) R1T1240
1004 FORMAT(5X,10HBASE =,F7.5, 2F15.5 / R1T1241
1      5X,10HCAMBER =,F7.5 / R1T1242
2      5X,10HDRAG RISE=,F7.5 / R1T1243
3      5X,10HMISC =,F7.5 ) R1T1244
1005 FORMAT(1H1, /1X, 6A10 ) R1T1245
1006 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/FT =*1PE15.4,5X, R1T1246
1      *L.E. SWEEP ANGLE =* OPF6.2,* DEG.*,5X,*TAIL DEFL. (DH) =* R1T1247
2      F6.2,* DEG.* R1T1248
2      //T10,*CL*, T22,*TOTAL CD*, T38,*CM*, T53,*ALPHA*, R1T1249
3      T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH=0*, R1T1250
4      T113,*CD AT DH=0*, / ) R1T1251
1007 FORMAT(/5X, 10H COMIN =,F7.5 //) R1T1252
1008 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* FT.*, 5X, R1T1253
1      *L.E. SWEEP ANGLE =*F6.2,* DEG.*,5X,*TRIMMED CONDITION* R1T1254
2      //T10,*CL*, T22,*TOTAL CD*, T38,*DH*, T53,*ALPHA*, R1T1255
3      T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =0*, R1T1256
4      T113,*CD AT DH=0*, / ) R1T1257
1009 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/FY =*1PE15.4,5X, R1T1258
1      *L.E. SWEEP ANGLE =* OPF6.2,* DEG.*,5X,*TRIMMED CONDITION* R1T1259
2      //T10,*CL*, T22,*TOTAL CD*, T38,*DH*, T53,*ALPHA*, R1T1260
3      T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH=0*, R1T1261
4      T113,*CD AT DH=0*, / ) R1T1262
C      R1T1263
      R1T1264
      END

```

CC = 00198

	SUBROUTINE ADJUST(ID, ID2, XVAR, YVAR)	R1T1266
C		R1T1267
C	ADJUST Y-VARIABLE USING CORRELATION FACTORS	R1T1268
C	COMMON /BLKADJ/ IVAL(20), X(15), YM(15,9), YA(15,9), XCL(15),	R1T1269
	1 NXVAR, NXCL	R1T1270
C		R1T1271
C	COMMON /BLKPRT/ KPRINT(50)	R1T1272
C		R1T1273
	YVAR1 = YVAR	R1T1274
	IF(ID.EQ.0) GO TO 300	R1T1275
	50 NV = IVAL(ID)	R1T1276
C		R1T1277
	IF(NV.EQ.0) RETURN	R1T1278
	100 CALL Lntp(XVAR, VM, X, YM(1,NV), NXVAR, 2)	R1T1279
	200 CALL Lntp(XVAR, VA, X, YA(1,NV), NXVAR, 2)	R1T1280
	GO TO 500	R1T1281
C		R1T1282
	300 NV = IVAL(ID2)	R1T1283
	IF(NV.EQ.0) RETURN	R1T1284
	400 CALL Lntp(XVAR, VM, XCL, YM(1,NV), NXCL, 2)	R1T1285
	CALL Lntp(XVAR, VA, XCL, YA(1,NV), NXCL, 2)	R1T1286
C		R1T1287
	500 YVAR = YVAR1 * VM + VA	R1T1288
C		R1T1289
	IF(KPRINT(24).EQ.0) RETURN	R1T1290
	WRITE(6,1000) ID, YVAR1, XVAR, VM, VA, YVAR	R1T1291
1000	FORMAT(5X,*ADJUST DUMP*, 5X, *ID =*,15/1X,5F15.5)	R1T1292
	RETURN	R1T1293
	END	R1T1294
		R1T1295

CC = 00030

	BLOCK DATA	RIT1297
C	COMMON /BLKBD1/ XSWPL(11), YRMIN(11)	RIT1298
C	COMMON /BLKBD2/ AAA(22), BBB(22), CCC(22), DDD(22), XTT(22)	RIT1299
C	COMMON /BLKBD3/ BOT(12), ATSW(6), TPRT(6), FXAC1(216), FXAC2(216)	RIT1300
C	COMMON /BLKBD4/ XBD4(5), YBD4(5), ZBD4(3), FBD4(5,5,3)	RIT1301
C	COMMON /BLKBD5/ XDY0(6), YXMN(6), ZLBD5(6,6)	RIT1302
C	COMMON /BLKBD6/ XBD6(4), YBD6(4), ZBD6(6), FBD6(4,4,6),	RIT1303
C	1 FBD62(4,4,6), FBD63(4,4,6)	RIT1304
C	COMMON /BLKMAP/ MAP, TRANS, DY, AMAP(22), BMAP(11)	RIT1305
	COMMON /BLKT22/ XIN13(10), YIN13(7), ZOUT13(10,7), ZOUI14(10,7)	RIT1306
	COMMON /BLKA15/ NXIN15, XIN15(6), NYIN15, YIN15(5),	RIT1307
	1 FOUT15(6,5), FOUT16(6,5)	RIT1308
	COMMON /BLKCLB/ X1(6), Y1(6), X9(8), Y9(8), XAR(5), YDCL(5)	RIT1309
	COMMON /BLKA16/ XSWP(7), YTR(4), FEP35(7,4), FEP7(7,4),	RIT1310
	1 XCLDB(7), YAKB(7)	RIT1311
C	COMMON /BLKMAX/ XTR(6), YC1(6), YC2(6), XSWP1(4), YA(4),	RIT1312
1	YB(4), XDY(8), XM(4), CTAB(8,4), DTAB(8,4),	RIT1313
2	XXCLM(13), YYDY(6), FCLMX(13,6), XXC2(9),	RIT1314
3	YMHACH(5), FOCCLMX(9,5), XDY1(9), YXMT(4),	RIT1315
4	ZC1MAX(9,4), XDY2(8), YFOC(6), ZOC1M(8,6),	RIT1316
5	ZFOC1M(8,6)	RIT1317
1	COMMON /BLK12/ XSP(8), YDYA(6), FDA(8,6), XAB(6), YCO(8),	RIT1318
1	PKVOFM(6,8), XANG(10), YRTOC(7), FRA(10,7)	RIT1319
	COMMON /BLKMX3/ SPAN(6), YTPR(4), FKB(6,4), FKS(6), FKD(6,4)	RIT1320
	COMMON /BLK15/ XC2(6), YAST(9), FDAM(6,9), XCT(6), YH(5), FDAM2(6,5)	RIT1321
	COMMON /BLK15/ XXXCLM(9), FCLMXX(9,6)	RIT1322
	COMMON /BLKMX6/ XX(12), XY(7), XF(12,7)	RIT1323
	COMMON /BLKMX7/ SPAN2(11), YTPR2(5), FKSW(11,5)	RIT1324
	COMMON /BLKMX8/ SPAN3(6), YTPR3(4), FKM(6,4)	RIT1325
	COMMON /BLKMX9/ SPAN4(5), YBF11(5), FKA(5,5),	RIT1326
1	SPAN5(5), YBF12(3), FKF(5,3)	RIT1327
C		RIT1328
C		RIT1329
C		RIT1330
	DATA XSWPL / 0.0, .17453, .34906, .5236, .65813, .87266, 1.04719,	RIT1331
1	1.13446, 1.22173, 1.34390, 1.57079 /,	RIT1332
2	YRMIN / 0.53, 0.53, 0.52, 0.515, 0.505, 0.49, 0.45,	RIT1333
3	0.39, 0.30, 0.12, 0.10 /	RIT1334
C		RIT1335
	DATA BOT / 0.0, .4, .8, 1.2, 1.6, 1.8, 2., 2.4, 2.8, 3.2, 3.6, 4.,	RIT1336
1	ATSW / 1., 2., 3., 4., 5., 6. /,	RIT1337
2	TPRT / 0., 0.2, 0.25, 0.33, 0.5, 1.0 /	RIT1338
C		RIT1339
C	FIG. 4.1.4.2-22 DATCOM ****	RIT1340
	DATA FXAC1/.25,.24,.23,.22,.20,.18,.17,.25,.32,.37,.40,.42,	RIT1341
1	7*.335, .39, .44, .48, .495, .5,	RIT1342
2	.42, .44, .45, .46, .465, .475, .50, .56, .56, .57, 2*.58,	RIT1343
		RIT1344
		RIT1345
		RIT1346
		RIT1347
		RIT1348
		RIT1349
		RIT1350
		RIT1351
		RIT1352

3	.50, .53, .55, .575, .60, .62, 6*0.67,	RIT1353
4	.58, .62, .66, .695, .73, .755, .83, .78, .78, .765, .75, .75,	RIT1354
5	.68, .72, .76, .79, .83, .765, .98, .90, .90, .87, .85, .84,	RIT1355
6	.29, .27, .25, .25, .23, .215, .20, .28, .33, .39, .44, .46,	RIT1356
7	.40, .41, .41, .405, .40, .40, .40, .48, .52, .54, .56, .57,	RIT1357
8	.51, .53, .54, .555, .57, .58, .60, .65, .665, .68, .69, .69,	RIT1358
9	.64, .66, .68, .70, .73, .70, .795, .80, .81, .815, .805, .80,	RIT1359
A	.75, .77, .80, .82, .87, .91, .97, .955, .95, .96, .93, .92,	RIT1360
B	.87, .895, .915, .945, 1.0, 1.05, 1.15, 1.12, 1.1, 1.1, 1.05, 1.03,	RIT1361
C	.30, .29, .275, .26, .245, .24, .23, .30, .35, .40, .45, .475,	RIT1362
D	.41, .42, .425, .425, .42, .415, .41, .50, .54, .57, .595, .60,	RIT1363
E	.54, .555, .57, .58, .60, .61, .63, .68, .69, .71, .72, .72,	RIT1364
F	.67, .69, .71, .73, .76, .79, .82, .83, .845, .865, .85, .845,	RIT1365
G	.80, .82, .84, .86, .90, .95, 1.02, 1.01, 1.0, 1.01, .98, .97,	RIT1366
H	.93, .95, .97, .99, 1.05, 1.1, 1.25, 1.2, 1.16, 1.17, 1.11, 1.1,	RIT1367
DATA FXAC2 /		RIT1368
I	.32, .31, .30, .28, .25, .24, .23, .31, .37, .42, .475, .50,	RIT1369
J	.45, .46, .46, .455, .45, .445, .44, .54, .57, .60, .63, .64,	RIT1370
K	.595, .60, .61, .62, .64, .65, .665, .72, .74, .76, .78, .775,	RIT1371
L	.73, .745, .765, .78, .80, .83, .88, .89, .90, .93, .92, .91,	RIT1372
M	.89, .90, .91, .93, .96, 1.0, 1.07, 1.08, 1.08, 1.1, 1.06, 1.05,	RIT1373
N	1.04, 1.05, 1.06, 1.09, 1.12, 1.17, 1.27, 1.26, 1.25, 1.26, 1.21, 1.2,	RIT1374
O	.36, .34, .325, .31, .30, .275, .25, .34, .41, .48, .52, .55,	RIT1375
P	.52, .52, .52, .52, .51, .505, .50, .60, .63, .67, .71, .72,	RIT1376
Q	.70, .70, .705, .71, .72, .735, .75, .80, .82, .87, .89, .89,	RIT1377
R	.88, .885, .89, .90, .915, .94, .98, 1.0, 1.02, 1.07, 1.05, 1.05,	RIT1378
S	1.04, 1.05, 1.05, 1.06, 1.09, 1.12, 1.19, 1.2, 1.22, 1.26, 1.22, 1.21,	RIT1379
T	1.2, 1.21, 1.22, 1.24, 1.27, 1.3, 1.36, 1.4, 1.42, 1.45, 1.4, 1.38,	RIT1380
U	.50, .48, .46, .44, .40, .38, .34, .42, .50, .60, .70, .76,	RIT1381
V	.75, .75, .745, .74, .72, .70, .68, .73, .80, .88, .97, 1.0,	RIT1382
W	5*1.0, .98, .96, 1.01, 1.08, 1.19, 1.22, 1.23,	RIT1383
X	5*1.24, 1.23, 1.2, 1.29, 1.38, 1.48, 1.48, 1.49,	RIT1384
Y	5*1.49, 1.47, 1.44, 1.55, 1.66, 1.77, 1.73, 1.75,	RIT1385
Z	3*1.74, 1.73, 1.71, 1.70, 1.68, 1.81, 1.95, 2.06, 1.98, 2.0 /	RIT1386

C
C ** FIGURE 4.1.4.1-2 , DATCOM *****

DATA XBD4 / 0.0, .34907, .69813, .87267, 1.0472 /,		RIT1387
1	YBD4 / 0.0, 1.5, 3.5, 6.0, 10.0 /,	RIT1388
2	ZBD4 / 0.0, 0.5, 1.0 /,	RIT1389
3	FBD4 / 5*0.0, 0.0, 0.0, 0.0, -.0002, -.0005,	RIT1390
4	0.0, -.0007, -.002, -.0027, -.0037,	RIT1391
5	0.0, -.0019, -.0044, -.006, -.008,	RIT1392
6	0.0, -.008, -.016, -.021, -.0265,	RIT1393
7	5*0.0, 0.0, 0.0, -.0005, -.0008, -.0011,	RIT1394
8	0.0, -.001, -.0029, -.0045, -.0068,	RIT1395
9	0.0, -.003, -.007, -.0093, -.013,	RIT1396
1	0.0, -.008, -.016, -.021, -.0265,	RIT1397
2	5*0.0, 0.0, -.0002, -.0006, -.001, -.0012,	RIT1398
3	0.0, -.0013, -.003, -.004, -.0052,	RIT1399
4	0.0, -.0031, -.0066, -.0088, -.012,	RIT1400
5	0.0, -.008, -.016, -.0207, -.026 /	RIT1401
DATA XDYO / 0.8, 1.1, 1.4, 1.6, 2.0, 2.4 /,		RIT1402
1	YXMN / 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 /,	RIT1403
2	Z1BD5/ 3.714, 4.395, 5.432, 6.348, 8.545, 11.231, 4.029, 4.517,	RIT1404
3	5.249, 5.982, 7.874, 10.254, 4.395, 4.639, 5.127, 5.616,	RIT1405

4	7.081,8.973, 4.7,4.834,5.127,5.432,6.287,7.691,	RIT1409
5	5.371,5.213,5.188,5.31,5.799,6.836,	RIT1410
6	6.226,5.738,5.432,5.335,5.616,6.409 /	RIT1411

C
C
C

FIGURE 4.1.4.2-25 DATCOM *****

DATA XBD6 / 0., 1., 2., 3. /,	RIT1412
1 YBD6 / -2., -1., 0., 1. /,	RIT1413
2 ZBD6 / 0., 1., 2., 3., 4., 5. /,	RIT1414
3 FBD61 / 0.,.125,.16,.17,0.,.115,.17,.2,0.,.15,.195,.195,0.,	RIT1415
4 .165,.2,.2,.18,.26,.28,.3,.18,.27,.305,.32,.18,.285,.32,.32,.18,	RIT1416
5 .3,2*.32,.34,.39,.4,.412,.34,.405,.435,.45,.34,.425,2*.445,.34,	RIT1417
6.445,.45,.45,.5,.505,.515,.53,.5,.525,.54,.55,.5,.545,2*.55,.5,3*	RIT1418
7.55,.665,.62,.623,.626,.665,.653,.66,.665,8*.665,	RIT1419
8 .815,.73,.728,.722,.815,.775,.778,.78,.815,.79,.78,.78,.815,	RIT1420
9 3*.775 /	RIT1421
DATA FBD62 / 0.,.135,.17,.186,0.,.12,.19,.215,0.,.175,2*.215,0.,.2	RIT1422
1,2*.235,.2,.28,.298,.314,.2,.3,.35,.365,.2,.32,2*.36,.2,.365,2*.38	RIT1423
2,.4,.43,.44,.45,.4,.47,.495,.508,.4,.49,2*.51,.4,3*.52,	RIT1424
3 .6,.575,.578,.58,.6,.61,.63,.65,.6,.64,2*.65,.6,3*.65,	RIT1425
4 .8,3*.72,.8,.77,.78,.792,.8,.795,.792,.79,.8,3*.78,	RIT1426
5 .98,.84,.83,.82,.98,3*.92,.98,.925,2*.915,.98,3*.885 /,	RIT1427
6 FBD63 / 0.,.15,.18,.2,0.,.135,.23,.28,0.,.2,2*.275,0.,.24,2*.28,	RIT1428
7 .26,.35,.365,.38,.26,.365,.42,.45,.26,.405,2*.455,.26,3*.475,	RIT1429
8 .5,.535,.55,.565,.5,.585,.61,.63,.5,.605,2*.63,.5,3*.65,	RIT1430
9 .75,3*.735,.75,.78,.795,.81,.75,3*.8,.75,3*.82,	RIT1431
1 .98,.905,.895,.885,.98,3*.95,.98,.95,2*.945,.98,3*.96,	RIT1432
2 1.17,1.06,1.03,1.,1.17,1.1,1.08,1.065,1.17,1.09,2*1.065,1.17,	RIT1433
3 3*1.08 /	RIT1434

C

DATA XIN13/ 0.0,1., 2., 3., 4., 5., 6., 7., 8., 20. /,	RIT1435
1 YIN13/ 0.0, 20.0, 35.0, 50.0, 65.0, 72.5, 90.0 /,	RIT1436
2 ZOUT13/ .00973,.00973,.0093,.009,.0089,.0087,4*.0086,	RIT1437
3 .00973,.00973,.0093,.009,.0089,.0087,4*.0086,	RIT1438
4 .0097,.0097,.00927,.00902,.0089,.00872,4*.00862,	RIT1439
5 .0096,.0096,.0092,.009,.0089,.00872, 4*.00863,	RIT1440
6 .00943,.00943,.00897,.00882,.00875,.0085,4*.00854,	RIT1441
7 .0093,.0093,.0088,.0087,.00862,.0085,4*.00844,	RIT1442
8 .0093,.0093,.0088,.0087,.00862,.0085,4*.00844 /	RIT1443
DATA ZOUT14/1.28,1.28,1.27,1.252,1.244,1.233,1.227,1.217,2*1.21,	RIT1444
1 1.28,1.28,1.27,1.252,1.244,1.233,1.227,1.217,2*1.21,	RIT1445
2 1.324,1.324,1.311,1.295,1.281,1.266,1.258,1.24,2*1.23,	RIT1446
3 1.363,1.363,1.35,1.33,1.313,1.292,1.283,1.261,2*1.25,	RIT1447
4 1.392,1.392,1.38,1.36,1.339,1.317,1.303,1.279,2*1.27,	RIT1448
5 1.403,1.403,1.392,1.371,1.349,1.325,1.31,1.285,2*1.28,	RIT1449
6 1.403,1.403,1.392,1.371,1.349,1.325,1.31,1.285,2*1.28/RIT1450	RIT1451

C

DATA AMAP / 22.0, 21.7, 19.2, 18.35, 22.0, 21.2, 19.2, 27.0,11.75,	RIT1452
1 24.,24.,22.,20.,20.,19.,17.,15.,29.,27.,25., 2*0. /,	RIT1453
2 BMAP / 4 * 1.75, 3 * 2.0, 4 * 0.0 /	RIT1454

C

DATA NXIN15/ 6 /,	RIT1455
1 XIN15/ 0.0, 0.5, 0.6, 0.8, 0.9, 1.0 /,	RIT1456
2 NYIN15/ 5 /,	RIT1457
3 YIN15 / 0.0, 5.0, 15.0, 20.0, 23.0 /,	RIT1458
4 FOUT15/ 0.865, 0.756, 0.71, 0.6, 0.51, 0.45,	RIT1459

5		0.865, 0.756, 0.71, 0.6, 0.51, 0.45,	RIT1465
6		0.925, 0.965, 0.958, .905, .845, 0.78,	RIT1466
7		0.578, 1.02, 1.03, 1.02, 0.99, 0.975,	RIT1467
8		0.99, 1.04, 1.05, 1.05, 1.045, 1.04 /,	RIT1468
9	FOUT16/	6 * 1.05, 1.05, 1.04, 1.02, .94, .88, .84,	RIT1469
1		1.05, 1.05, 1.04, 1.01, 0.995, 0.985,	RIT1470
2		6 * 1.05, 6 * 1.05 /	RIT1471
	DATA X1	/ 0.7243, 0.8727, 1.0472, 1.2217, 1.3963, 1.5708 /,	RIT1472
1	Y1	/ 0.85, 0.69, 0.575, 0.51, 0.483, 0.48 /,	RIT1473
2	X9	/ 0.3665, 0.5236, 0.6981, 0.8727, 1.0472, 1.2217, 1.3963, 1.6 /,	RIT1474
3	Y9	/ 0.85, .625, .435, .35, 0.375, 0.447, 0.478, 0.48 /,	RIT1475
4	XAR	/ 1.0, 2.3, 3.0, 4.0, 5.0 /,	RIT1476
5	YDCL	/ -0.16, 0.0, 0.05, 0.085, 0.0855 /	RIT1477
			RIT1478
	DATA AAA	/ 8.48243E-4, 8.03417E-4, 3.31322E-6, -1.72323E-3,	RIT1479
4		2.02978E-3, 1.36146E-4, 3.79206E-5, 15 * 0.0 /,	RIT1480
5	BBB	/ -.0296429, -.0349676, -.0142049, .0310185,	RIT1481
6		-.07305, -4.05767E-3, -1.03545E-3, 15 * 0.0 /,	RIT1482
7	CCC	/ 1.23835, 1.15999, 1.03825, 0.585837, 1.57142, .712684,	RIT1483
8		.639394, 0.0, 0.0, 5 * 1.1019, 3 * .275475,	RIT1484
9		3 * 3.3057, 0.0, 0.0 /,	RIT1485
1	DDD	/ -2.47002, -1.71448, -1.9292, -.559968, -2.99739,	RIT1486
2		.0188291, .0607728, 15 * 0.0 /,	RIT1487
3	XTT	/ .35, .375, .41, .45, .37, .39, .42, .371, .5, .2,	RIT1488
4		.3, .4, .5, .6, .3, .4, .5, .3, .4, .5, 0.0, 0.0 /	RIT1489
	DATA XSWP	/ 0.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0 /,	RIT1490
1	YTR	/ 0.0, 0.25, 0.5, 1.0 /,	RIT1491
2	FFP35	/ 0.94, 0.951, 0.96, 0.968, 0.973, 0.974, 0.971,	RIT1492
3		0.992, 0.995, 0.997, 0.999, 0.998, 0.992, 0.984,	RIT1493
4		1.0, 1.0, 0.999, 0.997, 0.988, 0.971, 0.944,	RIT1494
5		0.996, 0.992, 0.986, 0.975, 0.959, 0.93, 0.863 /,	RIT1495
6	FEP7	/ 0.89, 0.907, 0.923, 0.936, 0.944, 0.948, 0.942,	RIT1496
7		0.99, 0.994, 0.996, 0.998, 0.991, 0.978, 0.956,	RIT1497
8		0.996, 0.992, 0.987, 0.98, 0.961, 0.924, 0.86,	RIT1498
9		0.986, 0.968, 0.942, 0.91, 0.874, 0.833, 0.79 /	RIT1499
	DATA XCLDB	/ 0.0, 0.3, 0.4, 0.5, 0.6, 0.7, 2.0 /,	RIT1500
1	YAKB	/ 0.33, 0.352, 0.44, 0.625, 1.0, 1.47, 2.0 /	RIT1501
			RIT1502
			RIT1503
	DATA XTR	/ 0.0, 0.1, 0.2, 0.3, 0.5, 1.0 /,	RIT1504
1	YC1	/ 0.0, 0.225, 0.47, 0.5, 0.32, 0.0 /,	RIT1505
2	YC2	/ 0.0, 0.22, 0.5, 0.91, 1.05, 0.85 /,	RIT1506
3	XSWP1	/ 0.0, 30.0, 50.0, 60.0 /,	RIT1507
4	YA	/ 0.9, 1.04, 1.2, 1.3 /,	RIT1508
5	YB	/ 0.0, 0.24, 0.5, 0.71 /,	RIT1509
6	XDY	/ 0.0, 2.0, 2.25, 2.5, 3.0, 4.0, 4.5, 10.0 /,	RIT1510
7	XM	/ 0.2, 0.4, 0.6, 1.0 /	RIT1511
	DATA CTAB	/ 8 * 0.0, -.02, -.02, -.26, -.32, -.355, -.41, -.445, -.45,	RIT1512
9		.12, .12, -.36, -.43, -.5, -.64, -.72, -.72,	RIT1513
1		.58, .58, -.15, -.34, -.57, -.76, -.92, -.92 /,	RIT1514
2	DTAB	/ 8 * 0.0, -.085, -.085, .02, .055, .045, .02, .02, .02,	RIT1515
3		-.2, -.2, -.01, -.02, -.03, -.07, -.09, -.09, 8*0.0 /,	RIT1516
4	XXCLM	/ 0.0, .2, .4, .6, .8, 1., 1.2, 1.6, 2.0, 2.4, 2.8,	RIT1517
5		3.2, 4.4 /,	RIT1518
6	YYDY	/ 0.0, .25, .5, .75, 1.0, 10.0 /	RIT1519
	DATA FCLMX	/ 0.9, 1.375, 1.57, 1.645, 1.645, 1.575, 1.4, 1.14,	RIT1520

8		1.02, .96, .92, .91, .9, .75, 1.28, 1.48, 1.55,	RIT1521
9		1.56, 1.5, 1.33, 1.11, 1.0, .94, .9, .88, .875,	RIT1522
1		.68, 1.19, 1.4, 1.47, 1.475, 1.4, 1.27, 1.08,	RIT1523
2		.975, .92, .88, .85, .84, .6, 1.1, 1.3, 1.375,	RIT1524
3		1.385, 1.33, 1.23, 1.06, .96, .9, .865, .84, .83,	RIT1525
4		.4, 1.02, 1.2, 1.29, 1.31, 1.27, 1.19, 1.04, .94,	RIT1526
5		.88, .84, .83, .82, .4, 1.02, 1.2, 1.29, 1.31,	RIT1527
6		1.27, 1.18, 1.02, .91, .85, .82, .8, .8 /,	RIT1528
7	XXC2 /	0.0, 2.0, 4., 6., 8., 10., 12., 14., 20./,	RIT1529
8	YYMACH /	0.0, 0.2, 0.4, 0.6, 1.0 /,	RIT1530
9	FDCLMX /	-.11, -.08, 0.0, .115, .23, .335, .36, .22, 0.0,	RIT1531
1		-.11, -.08, 0.0, .115, .23, .335, .36, .22, 0.0,	RIT1532
2		-.11, -.08, 0.0, .1, .2, .3, .32, .18, 0.0,	RIT1533
3		-.11, -.08, 0.0, .08, .15, .21, .205, .08, 10*0.0/	RIT1534
C			RIT1535
C	** FIGURE 4.1.1.4-5, DATCOM *****		RIT1536
C			RIT1537
	DATA XDY1 /	0.0, 1.1, 2.25, 2.5, 3.0, 3.5, 4., 4.5, 6. /,	RIT1538
1	YXMT /	0.3, 0.35, 0.4, 0.45 /,	RIT1539
2	ZC1MAX /	0.8, 0.8, 1.315, 1.43, 1.58, 1.58, 1.54, 1.47, 1.41,	RIT1540
3		0.8, 0.8, 1.315, 1.43, 1.51, 1.52, 1.49, 1.41, 1.36,	RIT1541
4		0.8, 0.8, 1.315, 1.39, 1.45, 1.45, 1.43, 1.35, 1.34,	RIT1542
5		0.8, 0.8, 1.3, 1.33, 1.35, 1.36, 1.33, 1.32, 1.32/,	RIT1543
6	XDY2 /	0.0, 1.0, 2.0, 2.2, 2.4, 3.0, 5.2, 10.0 /,	RIT1544
7	YFOC /	0.0, 2.0, 4.0, 6.0, 8.0, 100.0 /,	RIT1545
8	ZDC1M /	10*0.0, 0.1, 0.13, 0.15, 0.145, 0.05, 0.0,	RIT1546
9		0.0, 0.11, 0.275, 0.3, 0.3, 0.22, 0.10, 0.0,	RIT1547
1		0.1, 0.26, 0.44, 0.465, 0.45, 0.30, 0.17, 0.0,	RIT1548
2		0.2, 0.40, 0.60, 0.620, 0.60, 0.39, 0.25, 0.0,	RIT1549
3		0.2, 0.40, 0.60, 0.620, 0.60, 0.39, 0.25, 0.0 /	RIT1550
	DATA Z2DC1M/	9*0., .15, .28, .28, .27, .18, .07, .07,	RIT1551
1		.1, .28, .49, .47, .41, .26, .12, .12,	RIT1552
2		.2, .41, .61, .60, .55, .35, .17, .17,	RIT1553
3		.33, .52, .68, .65, .60, .39, .20, .20,	RIT1554
4		.33, .52, .68, .65, .60, .39, .20, .20 /	RIT1555
C			RIT1556
	DATA XSP /	0.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 90.0/,	RIT1557
1	YDYA /	0.0, 1.2, 2.0, 3.0, 4.0, 50.0 /,	RIT1558
2	FDA /	1.8, 2.2, 3.4, 5.0, 7.4, 10.2, 13.4, 13.4,	RIT1559
3		1.8, 2.2, 3.4, 5.0, 7.4, 10.2, 13.4, 13.4,	RIT1560
4		0.1, 1.1, 2.3, 3.9, 5.8, 6.75, 9.9, 9.9,	RIT1561
5		1.3, 1.7, 2.4, 3.2, 4.2, 5.3, 6.7, 6.7,	RIT1562
6		2.2, 2.0, 2.1, 2.3, 2.5, 2.85, 3.25, 3.25,	RIT1563
7		2.2, 2.0, 2.1, 2.3, 2.5, 2.85, 3.25, 3.25 /	RIT1564
C			RIT1565
	DATA XAB /	0.0, 1.0, 2.0, 3.0, 4.0, 6.0 /,	RIT1566
1	YCO /	-1., -.75, -.5, -.25, 0.0, .25, .5, .75 /,	RIT1567
2	FKVOFM /	1.57, 1.74, 2.02, 2.30, 2.58, 2.75,	RIT1568
3		1.78, 1.95, 2.18, 2.52, 2.85, 2.90,	RIT1569
4		2.03, 2.21, 2.42, 2.71, 3.00, 3.15,	RIT1570
5		2.48, 2.60, 2.73, 2.93, 3.19, 3.33,	RIT1571
6		3.13, 3.15, 3.20, 3.28, 3.45, 3.60,	RIT1572
7		3.79, 3.80, 3.81, 3.82, 3.88, 3.98,	RIT1573
8		6 * 4.45, 6 * 5.0 /	RIT1574
	DATA XANG /	0.0, .0698, .1396, .2094, .2792, .3490, .4188, .4886,	RIT1575
1		0.5584, 0.6980 /,	RIT1576

2	YRTOC / 0.0, 0.03, 0.05, 0.08, 0.12, 0.16, 0.5 /,	RIT1577
3	FRA / 0.45,0.245,0.075, 7*0.0,	RIT1578
4	1.0,.62,.40,.255,.19,.148,.11,.079,.05, 0.0,	RIT1579
5	1.0,.84,.72,.48,.34,.252,.195,.142,.1, 0.0,	RIT1580
6	1.0,.90,.858,.758,.618,.46,.345,.25, .17, 0.0,	RIT1581
7	1.0,.917,.885,.814,.72,.62,.51,.38, .23, 0.0,	RIT1582
8	1.0,.925,.895,.835,.76,.665,.565, .46, .35, 0.0,	RIT1583
9	1.0,.925,.895,.835,.76,.665,.565, .46, .35, 0.0 /	RIT1584
C		RIT1585
	DATA SPAN / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,	RIT1586
1	YTPR / 0.0, 0.25, 0.5, 1.0 /,	RIT1587
2	FKB / 0.0, 0.3, 0.555, 0.77, 0.925, 1.0,	RIT1588
3	0.0, 0.28,0.53, 0.75, 0.915, 1.0,	RIT1589
4	0.0, 0.265, 0.51, 0.73, 0.9, 1.0,	RIT1590
5	0.0, 0.25, 0.48, 0.70, 0.885, 1.0 /,	RIT1591
6	FKS / 0.0, 0.31, 0.63, 0.85, 0.945, 1.0 /,	RIT1592
7	FKD / 0.0, 0.38, 0.64, 0.84, 0.965, 1.0,	RIT1593
8	0.0,0.295, 0.54, 0.74, 0.9, 1.0,	RIT1594
9	0.0, 0.24, 0.475, 0.68, 0.855, 1.0,	RIT1595
1	0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /	RIT1596
C		RIT1597
	DATA XC2 / 0.0, 1.0, 2.0, 3.0, 4.0, 5.0 /,	RIT1598
1	YAST / 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0, 30.0 /,	RIT1599
2	FDAM / 10.,7.,4.,1.5,0.,.5, 8.6,5.5,2.5,.5,-.3,.5,	RIT1600
3	7.5,4.,1.,-.6,-1.,5, 5.5,1.5,-1.2,-2.5,-1.,.5,	RIT1601
4	3.,-1.5,-4.1,-3.,-1.,.5, .2,-4.5,-5.,-3.,-1.,.5,	RIT1602
5	-2.3,-6.5,-5.,-3.,-1.,.5, -4.5,-7.,-5.,-3.,-1.,.5,	RIT1603
6	-8.,-7.,-5.,-3.,-1.,0.5 /,	RIT1604
7	XCT / 4.0, 6.0, 8.0, 10.0, 12.0, 14.0 /,	RIT1605
8	YM / 0.2, 0.4, 0.6, 0.8, 1.0 /,	RIT1606
9	FDAM2 / 0.,1.5,4.,7.2,11.,15., 0.,.8,3.,6.,9.5,13.5,	RIT1607
1	0.,.3,1.5,3.2,5.5,8.6, 0.,0.,.5,2.,4.,7.,	RIT1608
2	0., .8, 3., 6., 9.5, 13.5 /	RIT1609
C		RIT1610
	DATA XXXCLM / 0.0, .2, .4, .6, .8, 1., 1.2, 1.6, 2.0 /,	RIT1611
1	FCLMXX / 0.7,1.2,1.37,1.43,1.43,1.365,1.25,1.08,.99,	RIT1612
1	.65,1.12,1.29,1.355,1.365,1.305,1.21,1.07,.98,	RIT1613
2	.6,1.09,1.23,1.29,1.295,1.24,1.17,1.05,.97,	RIT1614
3	.55,1.0,1.15,1.23,1.24,1.20,1.15,1.04,.96,	RIT1615
4	.5,.91,1.09,1.17,1.20,1.175,1.13,1.02,.935,	RIT1616
5	.5,.91,1.09,1.17,1.20,1.165,1.11,1.0,.91 /	RIT1617
C		RIT1618
	DATA XX / 0.0,.4,.8,1.0,1.2,1.4,1.6,2.0,2.4,2.8,3.2,3.37 /,	RIT1619
1	XY / 0.,.1309,.1745,.2618,.3490,.4363,1.0 /,	RIT1620
2	XF / 8*1.0,.915,.63,.29,0.0,	RIT1621
3	8*1.0,.915,.63,.29,0.0,	RIT1622
4	7*1.0,.915,.755,.52,.245,0.0,	RIT1623
5	1.0,.97,.95,.97,1.0,.95,.87,.69,.552,.38,.185,0.0,	RIT1624
6	.92,.90,.90,.95,1.0,.95,.845,.63,.48,.31,.150,0.0,	RIT1625
7	.79,.80,.84,.91,1.0,.945,.83,.585,.425,.27,.128,0.,	RIT1626
8	.79,.80,.84,.91,1.0,.945,.83,.585,.425,.27,.128,0. /	RIT1627
C		RIT1628
	DATA SPAN2 /0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0 /,	RIT1629
1	YTPR2 / 0.0, 0.2, 0.333, 0.5, 1.0 /,	RIT1630
2	FKSW / 0.0, 0.0285, 0.0435, 0.0490, 0.0480, 0.0417,	RIT1631
3	0.0320, 0.0212, 0.0112, 0.0035, 0.0,	RIT1632

4	0.0, 0.0310, 0.0460, 0.0545, 0.0565, 0.0530,	RIT1633
5	0.0460, 0.0350, 0.0240, 0.0120, 0.0,	RIT1634
6	0.0, 0.0290, 0.0460, 0.0560, 0.0590, 0.0570,	RIT1635
7	0.0510, 0.0410, 0.0290, 0.0150, 0.0,	RIT1636
8	0.0, 0.0270, 0.0450, 0.0560, 0.0610, 0.0600,	RIT1637
9	0.0545, 0.0450, 0.0330, 0.0180, 0.0,	RIT1638
1	0.0, 0.0225, 0.0400, 0.0525, 0.0605, 0.0620,	RIT1639
2	0.0600, 0.0520, 0.0400, 0.0230, 0.0 /	RIT1640
C DATA SPAN3 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,		RIT1641
1	YTPR3 / 0.0, 0.25, 0.5, 1.0 /,	RIT1642
2	FKM / 0.0, 0.645, 1.045, 1.240, 1.320, 1.330,	RIT1643
3	0.0, 0.440, 0.760, 0.955, 1.070, 1.120,	RIT1644
4	0.0, 0.310, 0.575, 0.775, 0.930, 1.030,	RIT1645
5	0.0, 0.200, 0.400, 0.600, 0.800, 1.000 /	RIT1646
DATA SPAN4 / 0.0, 0.4, 0.8, 1.6, 2.4 /,		RIT1647
1	YBF11 / 0.0, 0.05, 0.1, 0.15, 0.2 /,	RIT1648
2	FKA / 0.0, 0.38, 0.68, 1.00, 1.13,	RIT1649
3	0.0, 0.40, 0.74, 1.12, 1.32,	RIT1650
4	0.0, 0.49, 0.83, 1.38, 1.75,	RIT1651
5	0.0, 0.57, 1.05, 1.81, 2.43,	RIT1652
6	0.0, 0.70, 1.27, 2.34, 3.37 /	RIT1653
DATA SPAN5 / 0.2, 0.4, 0.6, 0.8, 1.0 / ,		RIT1654
1	YBF12 / 0.0, 0.1, 0.2 /,	RIT1655
2	FKF / 4.00, 1.60, 0.60, 0.17, 0.0,	RIT1656
3	3.40, 1.55, 0.65, 0.26, 0.12,	RIT1657
4	2.96, 1.46, 0.72, 0.40, 0.25 /	RIT1658
END		RIT1659
		RIT1660

CC = 00364

	OVERLAY(3,1)	RIT1662
	PROGRAM VGEOM	RIT1663
C		RIT1664
C	COMPUTES GEOMETRY FOR VARIABLE-SWEEP CONFIGURATIONS	RIT1665
C		RIT1666
	COMMON /BLKVGM/ KPASS, SWPLE, DOBR, CLDR, ARNR, SWLER, SWQCR,	RIT1667
1	SWMCR, SWTER, TOCR, TANSR, DZ, SXR, SWPMCR,	RIT1668
2	CBXR, CTXR, CBXPR, SOXPR, ARXRR, AROPR, FOCR,	RIT1669
2	DA1, DAC1, DSWP1, CPOCR,	RIT1670
3	DA2, DAC2, DSWP2, DTOC	RIT1671
C		RIT1672
	COMMON /BLKAO1/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	RIT1673
C		RIT1674
	COMMON /BLKAO2/ SREF, AR, TAPR, SWPR,	RIT1675
1	BLN(10), BWID(10), BHGT(10), BAWET(10), BQ(10),	RIT1676
2	BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),	RIT1677
3	BASE(10), ELEN(10), EWID(10), EHGT(10),	RIT1678
4	EAWE(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),	RIT1679
5	ELBT(10), EQF(10), ENO(10), CBAR(10), TW,	RIT1680
6	XLEW, YWW, YBD, CR, BOZ, BFUS, FMISC, AB, AFTAW,	RIT1681
6	CAN(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,	RIT1682
7	TWIST, ETWIST, WINC, XLE(11), CRW(11), YW(11),	RIT1683
8	XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTOC,	RIT1684
9	SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),	RIT1685
1	SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),	RIT1686
2	SCR(10), HTLE, HTY, HTZ, HTINC	RIT1687
C		RIT1688
	COMMON /BLKGO1/ FRBOD(10), ARS(10), SSEX(10), FRNAC(10), SWP, SWPQC,	RIT1689
1	SWPMC, SWPTE, DOB, TOCW, CLD, SEXW, ESWMC,	RIT1690
2	SXX(10), XLESWX(10), XMCSWX(10), ARW, TR, WPLAN,	RIT1691
3	CB, YB, XB, CRX, CBX, CTX, YIX, YOX, SIX, SOX,	RIT1692
4	ARI, ARXR, CBXP, AROP, SOXP, SWPLEI, SWPLEC,	RIT1693
5	SMPMCI, SWPMCO, ESWQC, ESWLE, ESWTE, CROB, DXOB,	RIT1694
6	XHT, XCRTF, XH, OMEGA, FOC, TWIST, DINC, SWPMT,	RIT1695
7	CBAR2, CLDS, TOCS, SWET, DXQC, GI(81)	RIT1696
C		RIT1697
	COMMON /BLKPRT/ KPRINT(50)	RIT1698
	DIMENSION X(6), Y(6), XT(6), YT(6), D(6), P(6)	RIT1699
C		RIT1700
C		RIT1701
C		RIT1702
	IF(KPASS.EQ.1) GO TO 5	RIT1703
	IF(SWPLE.EQ.SWPR) GO TO 10	RIT1704
	IF(KPASS.EQ.2) GO TO 20	RIT1705
	GO TO 25	RIT1706
5	DOBR = DOB	RIT1707
	CLDR = CLD	RIT1708
	SWLER = ESWLE	RIT1709
	SWQCR = ESWQC	RIT1710
	SWMCR = ESWMC	RIT1711
	SWTER = ESWTE	RIT1712
	ARWP = ARW	RIT1713
	TOCR = TOCW	RIT1714
	CBXR = CBX	RIT1715
	CYXR = CTX	RIT1716
	CBXPR = CBXP	RIT1717

C
C

5

C

R1T1718
R1T1719
R1T1720
R1T1721
R1T1722
R1T1723
R1T1724
R1T1725
R1T1726
R1T1727
R1T1728
R1T1729
R1T1730
R1T1731
R1T1732
R1T1733
R1T1734
R1 1735
R1 1736
R1. 1737
R1T1738
R1T1739
R1T1740
R1T1741
R1T1742
R1T1743
R1T1744
R1T1745
R1T1746
R1T1747
R1T1748
R1T1749
R1T1750
R1T1751
R1T1752
R1T1753
R1T1754
R1T1755
R1T1756
R1T1757
R1T1758
R1T1759
R1T1760
R1T1761
R1T1762
R1T1763
R1T1764
R1T1765
R1T1766
R1T1767
R1T1768
R1T1769
R1T1770
R1T1771
R1T1772
R1T1773

I	SWPMC = ATAN(0.5 * (TANSR + TAN(SWPR)))	R1T1774
	DSWP1 = SWMT - ATAN(0.5 * (TAN(SWPR) + TAN(SWPMC)))	R1T1775
	SWPTE = ATAN(TANSR)	R1T1776
	CPJCR = 0.5 * (COS(SWPR)/COS(SWPR-SWPMC) + COS(SWPTE)/	R1T1777
	COS(SWPMC-SWPTE))	R1T1778
C		R1T1779
	DA2 = 0.0	R1T1780
	DAC2 = 0.0	R1T1781
	DSWP2 = 0.0	R1T1782
	DTDC = 0.0	R1T1783
C		R1T1784
	GO TO 50	R1T1785
C		R1T1786
	20 CONTINUE	R1T1787
C		R1T1788
C		R1T1789
C		R1T1790
	25 Y(2) = YW(1+NPMLS)	R1T1791
	CR = SPLAN/(Y(2)* (1.+TAPR))	R1T1792
	X(4) = CR + XAPEX	R1T1793
	X(1) = XAPEX	R1T1794
	Y(1) = 0.0	R1T1795
	Y(4) = 0.0	R1T1796
	X(2) = X(1) + Y(2) * TAN(SWPR)	R1T1797
	X(3) = X(2) + TAPR * CR	R1T1798
	Y(3) = Y(2)	R1T1799
	DO 30 I = 1,4	R1T1800
	D(I) = SQRT((X(I) - XPIVOT)**2 + (Y(I) - YPIVOT)**2)	R1T1801
	P(I) = ATAN((X(I) - XPIVOT) / (Y(I) - YPIVOT))	R1T1802
	IF((Y(I) - YPIVOT).LE.0.0) P(I) = P(I) + 3.141592	R1T1803
C	30 CONTINUE	R1T1804
		R1T1805
C	DSWP = SWPLE - SWPR	R1T1806
C		R1T1807
	DO 40 I = 1,4	R1T1808
	XT(I) = XPIVOT + D(I) * SIN(DSWP + P(I))	R1T1809
	YT(I) = YPIVOT + D(I) * COS(DSWP + P(I))	R1T1810
C	40 CONTINUE	R1T1811
	DXQC = 0.75* XT(2) + 0.25* XT(3) - XLE(1) - CRW(1)*0.25	R1T1812
	SWPTE = ATAN((XT(3) - XT(4))/(YT(3) - YT(4)))	R1T1813
	CO = TAN(SWPTE) / TAN(SWPLE)	R1T1814
	SWPQC = ATAN((1.0 - (1.0 - CO)/4.0)* TAN(SWPLE))	R1T1815
	SWPMC = ATAN((1.0 - (1.0 - CO)/2.0)* TAN(SWPLE))	R1T1816
	XFWD = XT(2) - YT(2) * TAN(SWPLE)	R1T1817
	XAFT = XT(3) - YT(3) * TAN(SWPTE)	R1T1818
	CRCLP = XAFT - XFWD	R1T1819
	CPOC = 0.5 * (COS(SWPLE)/COS(SWPLE - SWPMC) + COS(SWPTE)/	R1T1820
	COS(SWPMC - SWPTE))	R1T1821
C	CLDS = CAM(NPMLS) * CPOC/CPOCR	R1T1822
	TDCS = TOC(NPMLS) * CPOC/CPOCR	R1T1823
	DOB = DOBR * Y(2)/YT(2)	R1T1824
C		R1T1825
	45 CONTINUE	R1T1826
	TTIP = 1.0 + YT(2)* TAN(SWPLE) * (CO - 1.0)/CRCLP	R1T1827
	TWIST = ATAN(DZ/(CRCLP*TTIP)) * 57.296	R1T1828
		R1T1829

```

C      DINC = WINC * (1.-TAN(DSWP) * TANSR) * COS(DSWP)                                RIT1830
C      C3   = CRCLP * (1.0 -(YT(3)/YT(2))*(1.0 - TTIP) )                             RIT1831
WPLAN = (CRCLP + C3) *YT(3) + C3 *(YT(2)-YT(3))                                     RIT1832
ARW   = 4. * YT(2)**2/WPLAN                                                         RIT1833
YMID  = YT(3) +(YT(2) - YT(3)) /(1.0/SQRT(C3/(TTIP*CRCLP)) + 1.0) RIT1834
TR    = 1. - (YMID/YT(2)) * (1.-TTIP)                                              RIT1835
XTE2  = XT(4) + YW(NPNLS)*TAN(SWPTE)                                              RIT1836
CRX2  = CRCLP * (1.-YW(NPNLS)/YT(2) * (1.-TTIP))                                RIT1837
SX2   = WPLAN - (CRCLP + CRX2) * YW(NPNLS)                                         RIT1838
STOTAL = SX2                                                                       RIT1839
TOCW  = TOCS                                                                       RIT1840
CLD   = CLDS                                                                       RIT1841
IF( NPNLS.EQ.1 ) GO TO 46                                                         RIT1842
C      S1   = (XT(4)-XT(3))/(YT(4)-YT(3))                                           RIT1843
S2    = (XLE(1)+CRW(1)-XLE(2)-CRW(2))/(YW(1)-YW(2))                             RIT1844
IF( S1.EQ.S2 ) GO TO 46                                                         RIT1845
YI    = (XT(4)-XLE(1)-CRW(1) +S1*YT(4) -S2*YW(1))/(S1-S2)                     RIT1846
DA    = (XTE2 - XLE(2) - CRW(2)) * (YW(2) - YI)                                  RIT1847
SX2   = SX2 + DA                                                                    RIT1848
STOTAL = SXX(1) + SX2                                                             RIT1849
TOCW  = SQRT((TOC(1)**2*SXX(1) + TOCS**2*SX2)/STOTAL)                         RIT1850
CLD   = SQRT((CAM(1)**2*SXX(1) + CLDS**2*SX2)/STOTAL)                         RIT1851
C      46 SWET = SX2 * (2.+1843*TOCS +1.5268*TOCS**2 -.8395*TOCS**3)             RIT1852
1      +DA1 +(DA2-DA1) * (SWPLE-SWPR)/(AFTSW-SWPR)                             RIT1853
TP    = TR * CRCLP/CRX2                                                            RIT1854
CBAR2 = 0.66667 * CRX2 * (1. + (TP**2/(1.+TP)) )                               RIT1855
CBAR2 = CBAR2 + DAC1 + (DAC2 - DAC1) * DSWP/(AFTSW-SWPR)                     RIT1856
C      SWPMT = SWPQC + DSWP1 + (DSWP2-DSWP1) * DSWP/(AFTSW-SWPR)               RIT1857
TOCW  = TOCW + DTOC * DSWP/(AFTSW - SWPR)                                       RIT1858
C      COSMC = COS(XMCSWX(1)) * SXX(1) + COS(SWPMC) * SX2                       RIT1859
COSQC  = COS(ATAN(0.5*(TAN(XLESWX(1))+TAN(XMCSWX(1)))) * SXX(1)                RIT1860
1      + COS(SWPQC) * SX2                                                         RIT1861
TANLE  = TAN(XLESWX(1)) * SXX(1) + TAN(SWPLE) * SX2                             RIT1862
TANTE  = (2.*TAN(XMCSWX(1))- TAN(XLESWX(1))) *SXX(1)                           RIT1863
1      + TAN(SWPTE) * SX2                                                         RIT1864
C      ESWMC = ACOS(COSMC/STOTAL)                                                  RIT1865
ESWQC  = ACOS(COSQC/STOTAL)                                                       RIT1866
ESWLE  = ATAN(TANLE/STOTAL)                                                       RIT1867
ESWTE  = ATAN(TANTE/STOTAL)                                                       RIT1868
C      SEXW  = STOTAL                                                             RIT1869
SXX(NPNLS) = SX2                                                                  RIT1870
XLESWX(NPNLS)= SWPLE                                                             RIT1871
XMCSWX(NPNLS)= SWPMC                                                             RIT1872
CBX    = CRX2                                                                     RIT1873
CTX    = CRCLP * TR                                                              RIT1874
SNX    = SX2                                                                     RIT1875
YOX    = YMID - YW(NPNLS)                                                         RIT1876
CBXP   = CRCLP * (1.-(YW(1)+YW(2))*0.5 *(1.-TR)/YMID )                       RIT1877
SNXP   = (CBXP + CTX) * (YMID -(YW(1)+YW(2))*0.5)                             RIT1878
RIT1879
RIT1880
RIT1881
RIT1882
RIT1883
RIT1884
RIT1885

```


ARXR	=	4. * (YMID - YW(1))**2 /SEXW	R1T1886
AROP	=	4. * (YMID - (YW(1)+YW(2)*.5))**2/SOXP	R1T1887
SWPLED	=	SWPLE	R1T1888
SWPMCO	=	SWPMC	R1T1889
FOC	=	CLD * FOCR /CLUR	R1T1890
C			R1T1891
	IF(KPASS.NE.2) GO TO 50		R1T1892
C			R1T1893
	IF(AFTAW.GT.0.0) DA2 = AFTAW - SWET		R1T1894
	IF(AFTCB.GT.0.0) DAC2 = AFTCB - CBAR2		R1T1895
	IF(AFTOC.GT.0.0) DTOC = AFTOC - TOCS		R1T1896
50	CONTINUE		R1T1897
	SWP = SWPLE		R1T1898
C			R1T1899
	IF(KPRINT(12).EQ.1) WRITE(6,1000) KPASS, SWPLE, DA1, DAC1,		R1T1900
1	DSWP1, CPOCR, DA2, DAC2, DSWP2, DTOC, DA		R1T1901
2	, TH1ST, DINC, SWPMT, CBAR2, CLDS, TOLS, SWET, DOB, TOCW,		R1T1902
3	CLD, SEXW, FSWLE, ESWQC, ESWMC, ESWTE, ARW, TR, WPLAN		R1T1903
1000	FORMAT(5X, *VGEOM DUMP, KPASS =*, I3, 10X, *SWPLE =* F12.4 /		R1T1904
1	(1X, 7F15.5)		R1T1905
C			R1T1906
	END		R1T1907

CC = 00246

OVERLAY(3,2)	RIT1909
PROGRAM MCRIT	RIT1910
C	RIT1911
C COMPUTES CRITICAL MACH NUMBER	RIT1912
C	RIT1913
COMMON /BLKOV3/ ID, XMACH, TOC, CLD, SWEEP, CV3(3), JPASS,	RIT1914
1 OV3A(3)	RIT1915
COMMON /BLKA05/ SCH(6), SECTG(5), C1MAX, CLMCR(10), XMCR(10),	RIT1916
1 XT(33), YT(33), YC(33), NMCR, NXSET	RIT1917
C	RIT1918
COMMON /BLKC03/ CLTAB(11), TABMCR(11)	RIT1919
C	RIT1920
IF(ID.LE.23) CALL CPZT(ID, 0.6, TOC, CLD, SWEEP)	RIT1921
C	RIT1922
100 CLTAB(1) = 0.0	RIT1923
DO 200 I = 1, 11	RIT1924
IF(I.GT.1) CLTAB(I) = CLTAB(I-1) + 0.1	RIT1925
CALL LNTP(CLTAB(I), TABMCR(I), CLMCR, XMCR, NMCR, 2)	RIT1926
CALL ADJUST(0.4, CLTAB(I), TABMCR(I))	RIT1927
IF(TABMCR(I).LT.0.0) TABMCR(I) = 0.0	RIT1928
C	RIT1929
200 CONTINUE	RIT1930
C	RIT1931
300 WRITE(6,1000) (CLTAB(I), TABMCR(I) , I =1,11)	RIT1932
C	RIT1933
1000 FORMAT(1H1, 5X, * MACH CRITICAL TABLE * /// 5X * CL *, 5X	RIT1934
1 * MACH CRITICAL * //(5X,2F15.4))	RIT1935
C	RIT1936
END	RIT1937

CC = 00029

	SUBROUTINE CPZT(ID, XMACH, TOC, CLD, SWEEP)	RIT1939
C		RIT1940
C	CALCULATES AIRFOIL PRESSURES - BRITISH METHOD	RIT1941
C	AND COMPUTES MACH CRITICAL USING CREST CRITERIA	RIT1942
C		RIT1943
	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	RIT1944
C		RIT1945
	COMMON /BLKA05/ SCW(6), SECTG(5), CIMAX, CLMCR(10), XMCR(10),	RIT1946
1	ORD(99), NMCR, NXSET	RIT1947
	COMMON /BLKRO7/ XU(32), ZT(32), ZS(32), RCC, ZTE	RIT1948
	COMMON /BLKCPI/ SA(32), SB(32), SC(32), SD(32), SE(32)	RIT1949
	COMMON /BLKPRT/ KPRINT(50)	RIT1950
	COMMON /BLKG01/ G1(200)	RIT1951
C		RIT1952
	DIMENSION C7SE(32), SINE(32), S1(32,32), S2(32,32), S3(33,32),	RIT1953
1	S4(32,32), S5(32,32), YCPI(32), DZOXUT(32)	RIT1954
C		RIT1955
	PI = 3.1415927	RIT1956
	AR = G1(80)	RIT1957
	SWP = SWEEP/57.29578	RIT1958
	COSPHI = (COS(SWP))** (AR/(1.4+AR))	RIT1959
	IF(SWEEP.GT.40.0) COSPHI = 0.5 * (COSPHI + .76604** (AR/(1.4+AR)))	RIT1960
C	COSPHI IS THE COSINE OF THE EFFECTIVE ISOBAR SWEEP	RIT1961
	SWP = ACOS(COSPHI)	RIT1962
C		RIT1963
	NCP = 31	RIT1964
	NMCR = 0	RIT1965
C		RIT1966
	DO 100 I= 1,NCP	RIT1967
C		RIT1968
	XI = I	RIT1969
	XN = NCP + 1	RIT1970
	NX = NCP + 1	RIT1971
	C7SE(I)= COS(XI*PI/XN)	RIT1972
	SINE(I)= SIN(XI*PI/XN)	RIT1973
	XU(I) = 0.5*(1.0 + C7SE(I))	RIT1974
100	CONTINUE	RIT1975
C		RIT1976
	CALL SECT(ID, TOC, CLD)	RIT1977
C		RIT1978
	DO 300 IV = 1, NCP	RIT1979
	DO 200 IU = 1, NCP	RIT1980
C		RIT1981
	IMV = IU - IV	RIT1982
	S1(IU,IV) = XN/SINE(IU)	RIT1983
	S2(IU,IV) = C7SE(IU)/(SINE(IU)**2)	RIT1984
	S3(IU,IV) = XN/SINE(IU)	RIT1985
	S4(IU,IV) = XN/SINE(IU) - 2.0 * ((-1.0)**IU -1.0) /	RIT1986
1	(XN * SINE(IU) * (1.0 - C7SE(IU)))	RIT1987
	S5(IU,IV) = - S2(IU,IV)	RIT1988
	IF(IU.EQ.IV) GO TO 150	RIT1989
C		RIT1990
	SIGN = ((-1.0)**IMV - 1.0)/XN	RIT1991
	DEM = (C7SE(IU) - C7SE(IV))	RIT1992
	S1(IU,IV) = SIGN * 2.0 * SINE(IU) /(DEM**2)	RIT1993
	S2(IU,IV) = (-2.0*(-1.0)**IMV*SINE(IU))/(SINE(IV)*DEM)	RIT1994

	S3(IU,IV) = S1(IU,IV) - SIGN * 2.0 * (1.0/(SINE(IU)*DEM))	R1T1995
	S4(IU,IV) = 2.0 * SIGN * (1.0 - COSE(IU)*COSE(IV)) /	R1T1996
1	(SINE(IV) * DEM**2)	R1T1997
2	- 2.0 * ((-1.0)**IU - 1.0) /	R1T1998
3	(XM * SINE(IV) * (1.0 - COSE(IU)))	R1T1999
	S5(IU,IV) = -2.0 * (-1.0)**IMV / DEM	R1T2000
C		R1T2001
150	IF(KPRINT(26).NE.3) GO TO 200	R1T2002
	WRITE(6,1000) S1(IU,IV), S2(IU,IV), S3(IU,IV)	R1T2003
1	,S4(IU,IV), S5(IU,IV)	R1T2004
200	CONTINUE	R1T2005
	S3(NX,IV) = ((-1.0)**IV - 1.0)/(XM*(1.0 + COSE(IV)))	R1T2006
300	CONTINUE	R1T2007
C		R1T2008
	IF(KPRINT(26).EQ.3) WRITE(6,1000) (S3(NX,J), J=1,NCP)	R1T2009
C		R1T2010
	ALPHA = 6.0	R1T2011
350	A = ALPHA/57.296	R1T2012
	NMCR = NMCR + 1	R1T2013
	XM = XMACH	R1T2014
	IF(XM.GT.0.9) XM = 0.9	R1T2015
	IF(XM.LE.0.0) XM = 0.01	R1T2016
	XM2 = XM**2	R1T2017
	IF(KPRINT(1).EQ.1) WRITE(6,1001) XM, ALPHA, SWEEP	R1T2018
	CLI = 0.0	R1T2019
	CL = 0.0	R1T2020
	DCPIM1 = 0.0	R1T2021
	DCPM1 = 0.0	R1T2022
	CMI = 0.0	R1T2023
	CM = 0.0	R1T2024
	DO 500 IV = 1, NCP	R1T2025
	SA(IV) = 0.0	R1T2026
	IF(ID.LT.10.OR.ID.GT.20) SB(IV) = 0.0	R1T2027
	SC(IV) = S3(NX,IV) * SQRT(ROC/2.0)	R1T2028
	SD(IV) = 0.0	R1T2029
	SE(IV) = 0.0	R1T2030
	DO 400 IU = 1, NCP	R1T2031
C		R1T2032
	SA(IV) = SA(IV) + S1(IU,IV)*ZT(IU)	R1T2033
	IF(ID.LT.10.OR.ID.GT.20) SB(IV) = SB(IV) + S2(IU,IV)*ZT(IU)	R1T2034
	SC(IV) = SC(IV) + S3(IU,IV)*ZT(IU)	R1T2035
	SD(IV) = SD(IV) + S4(IU,IV)*ZS(IU)	R1T2036
	SE(IV) = SE(IV) + S5(IU,IV)*ZS(IU)	R1T2037
400	CONTINUE	R1T2038
C		R1T2039
	CALL CPUOV(1.0, A, SWP, IV, CPIU, CPU, XM)	R1T2040
C		R1T2041
	CALL CPUOV(-1., A, SWP, IV, CPIL, CPL, XM)	R1T2042
C		R1T2043
	DCPI = CPIL - CPIU	R1T2044
	DCP = CPL - CPU	R1T2045
C		R1T2046
	DZDXU = SB(IV) + SE(IV)	R1T2047
	DZDXL = -SB(IV) + SE(IV)	R1T2048
C		R1T2049
	YCPI(IV) = CPIU	R1T2050

	DZDXUT(IV) = DZDXU	R1T2051
C	IF(KPRINT(1).EQ.1) WRITE(6,1000) XU(IV),CPIU,CPII,DCPI,DZDXU,	R1T2052
	1 DZDXL, CPU, CPL, DCP, XU(IV)	R1T2053
C		R1T2054
	DX = XU(NCP)	R1T2055
	XB = 0.5 * (1. + XU(1))	R1T2056
	IF(IV.NE.1) DX = XU(IV-1) - XU(IV)	R1T2057
	IF(IV.NE.1) XB = 0.5 * (XU(IV-1) + XU(IV))	R1T2058
	CLI = CLI + DX * (DCPI + DCPIM1) * 0.5	R1T2059
	CL = CL + DX * (DCP + DCPM1) * 0.5	R1T2060
	CMI = CMI + DX * (DCPI + DCPIM1)*0.5 * (XB - 0.25)	R1T2061
	CM = CM + DX * (DCP + DCPM1) *0.5 * (XB - 0.25)	R1T2062
C		R1T2063
	DCPIM1 = DCPI	R1T2064
	DCPM1 = DCP	R1T2065
C		R1T2066
	500 CONTINUE	R1T2067
C		R1T2068
	CLI = CLI + XU(NCP) * DCPI * 0.5	R1T2069
	CL = CL + XU(NCP) * DCP * 0.5	R1T2070
	CMI = CMI + XU(NCP) * DCPI * 0.5 * (0.5*XU(NCP) -0.25)	R1T2071
	CM = CM + XU(NCP) * DCP * 0.5 * (0.5*XU(NCP) -0.25)	R1T2072
	CPCRIT = (1.42857/XM2) * (0.52828 *(1. + 0.2 *XM2)**3.5 -1.)	R1T2073
C		R1T2074
	IF(KPRINT(1).GT.0) WRITE(6,1002) CLI,CMI,CPCRIT,ZTE,CL,CM	R1T2075
	600 CONTINUE	R1T2076
C		R1T2077
	IF(A.LT.DZDXUT(NCP)) GO TO 610	R1T2078
	NMCR = NMCR - 1	R1T2079
	ALPHA = ALPHA -1.	R1T2080
	GO TO 350	R1T2081
C		R1T2082
	610 CALL LNTP(A, XCREST, DZDXUT, XU, NCP, 4)	R1T2083
C		R1T2084
	620 CALL LNTP(XCREST, CPCR, XU, YCPI, NCP, 4)	R1T2085
C		R1T2086
	CPCRST = CPCR/COSPHI**2	R1T2087
	630 CPI2 = CPCRST * CPCRST	R1T2088
	XMDON = 1./((1.0323 -.8365*CPCRST -.361*CPI2 -.1336*CPCRST**3	R1T2089
	1 -.0173*CPCRST**4)	R1T2090
	IF(ID.EQ.8) XMDON = 1./((1.012 -.8551*CPCRST	R1T2091
	1 -.4493*CPI2 -.2219*CPCRST**3 -.0447*CPCRST**4)	R1T2092
C		R1T2093
	XMOD = XMDON/COSPHI	R1T2094
	IF(XMOD.LE.0.82.OR.ID.EQ.8) GO TO 640	R1T2095
	IF(XMOD.LE.1.08) XMOD = .2686 -.5729*XMOD +2.7123*XMOD**2	R1T2096
	1 -1.4582*XMOD**3	R1T2097
	IF(XMOD.GT.1.08) XMOD = 0.8545 +.1132*XMOD	R1T2098
C		R1T2099
	640 FMCRB = 1.0	R1T2100
	IF(NBODYS.GE.1) FMCRB = 1.0439 - 0.5828/G1(1)	R1T2101
	IF(G1(1).GT.10.0) FMCRB = 0.98	R1T2102
	IF(XMOD.GT.FMCRB) XMOD = FMCRB	R1T2103
C		R1T2104
	BMOD2 = 1. - (XMOD * COS(SWP))**2	R1T2105
		R1T2106

C	IF(BMDD2.GT.0.0) CLMDD = CLI/SQRT(BMDD2)	R1T2107
	IF(KPRINT(1).GT.0) WRITE(6,1003) XMDD,CLMDD,XCREST,CPCR,COSPHI	R1T2109
	CLMCR(NMCR) = CLMDD	R1T2110
	XMCR(NMCR) = XMDD	R1T2111
C		R1T2112
	700 IF(CLI.LE.0.0) RETURN	R1T2113
	IF(NMCR.GE.10) RETURN	R1T2114
	ALPHA = ALPHA - 1.	R1T2115
	GO TO 350	R1T2116
C		R1T2117
	1000 FORMAT(5X,10F12.4)	R1T2118
C		R1T2119
	1001 FORMAT(1H1, ///10X, 57HPRESSURE DISTRIBUTION AND SURFACE SLOPES	R1T2120
	10R 2-D AIRFOIL , 6X,*MACH =*F6.4, 5X,*ALPHA =*F6.3 ///	R1T2121
	1 10X,*SWEEP ANGLE =*, F7.3 //	R1T2122
	2 12X,*X/C*, 7X,*CPI(UPPER)*, 2X,*CPI(LOWER)*, 4X,*DCPI*,	R1T2123
	3 5X,*DZ/DX(UPPER)*, 2X,*DZ/DX(LOWER)*, 1X,*CP(UPPER)*,	R1T2124
	4 2X,*CP(LOWER)*, 4X,*DCP*, 9X,*X/C* //	R1T2125
C		R1T2126
	1002 FORMAT(//10X,*CLI =*F7.4, 10X,*CMI =*F7.4, 20X,*CP(M=1) = *,	R1T2127
	1 F8.4, 10X, *ZTE =*F7.4 /11X,*CL =*F7.4, 6X,*CM(C/4) =*	R1T2128
	2 F7.4 /)	R1T2129
C		R1T2130
	1003 FORMAT(/10X,*MACH CRIT =*F7.4, 10X, *CL AT MDD =*F7.4, 10X,	R1T2131
	1 *X-CREST =*F7.4, 10X,*CP-CREST =*F8.4 /	R1T2132
	2 50X, *COSINE OF EFFECTIVE ISOBAR =*F7.4 ///	R1T2133
C		R1T2134
	END	R1T2135

CC = 00197

	SUBROUTINE SECT(ID, TOC, CLD)	R1T2137
C		R1T2138
C	CALCULATES THICKNESS AND CAMBER	R1T2139
C		R1T2140
	COMMON /BLKCP1/ SA(32), SB(32), SC(32), SD(32), SE(32)	R1T2141
	COMMON /BLKRO7/ XU(32), ZT(32), ZS(32), ROC, ZTE	R1T2142
	COMMON /BLKA05/ XMU, ZMU, XML, ZML, ZPTE, ZTHIK, RLE, A4(25),	R1T2143
1	XT(33), YT(33), YC(33), NPCR, NXSET	R1T2144
	COMMON /BLKBD2/ AA(22), UB(22), CC(22), DD(22), XTT(22)	R1T2145
	COMMON /BLKPRT/ KPRINT(50)	R1T2146
C		R1T2147
	DIMENSION XSECT(26), Y164A(26), CAMB(26)	R1T2148
	DIMENSION Y163A(26), Y165A(26), Y163(26), Y164(26), Y165(26),	R1T2149
1	Y166(26), CAMB1(26)	R1T2150
	DIMENSION AO(11), A1(11), A2(11), A3(11), D1(11), D2(11), D3(11)	R1T2151
C		R1T2152
	DATA XSECT/ 0.0,0.005,0.0075,0.0125,0.025,0.05,0.075,0.1,	R1T2153
1	0.15,0.2,0.25,0.3,0.35,0.4,0.45,0.5,0.55,0.6,	R1T2154
2	0.65,0.7,0.75,0.8,0.85,0.9,0.95,1.0 /	R1T2155
	DATA Y163 / 0.0,.00829,.01004,.01275,.01756,.0244,.0295,.03362,	R1T2156
1	.03994,.04445,.04753,.04938,.05,.04938,.04766,.04496,	R1T2157
2	.0414,.03715,.03234,.02712,.02166,.01618,.01088,	R1T2158
3	.00604,.00214,0.0 /	R1T2159
	DATA Y164 / 0.0,.0082,.00989,.0125,.01701,.02343,.02826,.03221,	R1T2160
1	.03842,.04302,.04639,.04864,.0498,.04988,.04843,	R1T2161
2	.04586,.04238,.0382,.03345,.02827,.02281,.01722,	R1T2162
3	.01176,.00671,.00248,0.0 /	R1T2163
	DATA Y165 / 0.0,.00772,.00932,.01169,.01574,.02177,.02647,.0304,	R1T2164
1	.03666,.04143,.04503,.0476,.04924,.04996,.04963,	R1T2165
2	.04812,.0453,.04146,.03682,.03156,.02584,.01987,	R1T2166
3	.01385,.0081,.00306,0.0 /	R1T2167
	DATA Y166 / 0.0,.00759,.00913,.01141,.01516,.02087,.02536,.02917,	R1T2168
1	.0353,.04001,.04363,.04636,.04832,.04953,.05,.04971,	R1T2169
2	.04865,.04665,.04302,.03787,.03176,.02494,.01773,	R1T2170
3	.01054,.00408,0.0 /	R1T2171
	DATA Y163A/ 0.0,.00816,.00983,.0125,.01737,.02417,.02917,.03324,	R1T2172
1	.0395,.044,.04714,.04913,.04955,.04968,.04837,.04613,	R1T2173
2	.04311,.03943,.03517,.03044,.02545,.0204,.01535,.0103,	R1T2174
3	.00525,.00021 /	R1T2175
	DATA Y164A/ 0.0,0.00804,0.00969,0.01225,0.01688,0.02327,	R1T2176
1	0.02815,0.03199,0.03813,0.04272,0.04606,0.04837,	R1T2177
2	0.04968,0.04995,0.04894,0.04684,0.04388,0.04021,	R1T2178
3	0.03597,0.03127,0.02623,0.02103,0.01582,0.01062,	R1T2179
4	0.00541,0.00021 /	R1T2180
	DATA Y165A/ 0.0,.00765,.00928,.01183,.01623,.02182,.0265,.0304,	R1T2181
1	.03658,.04127,.04483,.04742,.04912,.04995,.04983,	R1T2182
2	.04863,.04632,.04304,.03899,.03432,.02912,.02352,	R1T2183
3	.01771,.01188,.00604,.00021 /	R1T2184
	DATA CAMB / 0.0,0.00281,0.00396,0.00603,0.01055,0.01803,	R1T2185
1	0.02432,0.02981,0.03903,0.04651,0.05257,0.05742,	R1T2186
2	0.06120,0.06394,0.06571,0.06651,0.06631,0.06508,	R1T2187
3	0.06274,0.05913,0.05401,0.04673,0.03607,0.02452,	R1T2188
4	0.01226,0.0 /	R1T2189
	DATA CAMB1/ 0.0,.0025,.0035,.00535,.0093,.0158,.0212,.02585,	R1T2190
1	.03365,.0398,.04475,.0486,.0515,.05355,.05475,.05515,	R1T2191
2	.05475,.05355,.0515,.0486,.04475,.0398,.03365,.02585,	R1T2192

3		.0158,0.0 /		R1T2193
	DATA A0	/ 5 * 0.2969, 3 * 0.14845, 3 * 0.514246 /,		R1T2194
1	A1	/ 0.213337, -.096082, -.246867, -.310275, -.27118,		R1T2195
2		0.412103, 0.193233, 0.083362, -.840115, 2*0.0 /,		R1T2196
3	A2	/ -2.931954, -.54331, 0.175334, 0.3417, 0.1402,		R1T2197
4		-1.67261, -.558168, -.18315, 1.1101, 2*0.0 /,		R1T2198
5	A3	/ 5.22917, 0.559395, -.266917, -.32182, -.082137,		R1T2199
6		1.68869, .283208, -.00691, -1.09401, 2*0.0 /,		R1T2200
7	D1	/ 0.2, 0.234, 0.315, 0.465, 0.7, 0.234, 0.315,		R1T2201
8		0.465, 0.234, 0.315, 0.465 /,		R1T2202
9	D2	/ -.040625, -.068571, -.233333, -.684, -1.6625,		R1T2203
1		-.068571, -.233333, -.684, -.068571, -.23333, -.684/,		R1T2204
2	D3	/ -.070312, -.093878, -.032407, 0.292, 1.3125,		R1T2205
3		-.093878, -.032407, 0.292, -.093878, -.032407, 0.292/		R1T2206
C				R1T2207
	NCP	= 31		R1T2208
	ROC	= AA(ID) + BB(ID)*TOC + CC(ID)*TOC*TOC + DD(ID)*TOC**3		R1T2209
C				R1T2210
C	ID	= 1 TO 4, 6 SERIES AIRFOIL	IJ = 1	R1T2211
C	ID	= 5 TO 7, 6A SERIES AIRFOIL	IJ = 1	R1T2212
C	ID	= 8, SUPERCritical AIRFOIL	IJ = 2	R1T2213
C	ID	= 9, BICONVEX	IJ = 3	R1T2214
C	ID	= 10 TO 20, 4 DIGIT AIRFOIL	IJ = 4	R1T2215
C	ID	= 21,22, INPUT AIRFOIL	IJ = 5	R1T2216
C				R1T2217
	IF(ID.LE.7)	IJ = 1		R1T2218
	IF(ID.EQ.8)	IJ = 2		R1T2219
	IF(ID.EQ.9)	IJ = 3		R1T2220
	IF(ID.GE.10.AND.ID.LE.20)	IJ = 4		R1T2221
	IF(ID.GE.21)	IJ = 5		R1T2222
C				R1T2223
	GO TO (200,300,100,900,700) , IJ			R1T2224
C				R1T2225
	100 WRITE(6,2000)			R1T2226
	DO 110 I = 1,NCP			R1T2227
	ZT(I)	= 2.0*TOC*XU(I)*(1.0 - XU(I))		R1T2228
	SB(I)	= 2.*TOC * (1.-XU(I))		R1T2229
	ZS(I)	= 0.0		R1T2230
	110 CONTINUE			R1T2231
	ROC	= 0.0		R1T2232
	GO TO 960			R1T2233
C				R1T2234
	200 IF(ID.LE.4) WRITE(6,2003)			R1T2235
	IF(ID.GT.4) WRITE(6,2001)			R1T2236
	DO 220 I = 1,NCP			R1T2237
	X	= XU(I)		R1T2238
	IF(ID.LE.4) CALL LNTP(X, ZS1, XSECT, CAMB1, 26, 4)			R1T2239
	IF(ID.GT.4) CALL LNTP(X, ZS1, XSECT, CAMB, 26, 4)			R1T2240
	ZS(I)	= ZS1 * CLD		R1T2241
	210 IF(ID.EQ.1) CALL LNTP(X, ZT(I), XSECT, Y163, 26, 4)			R1T2242
	IF(ID.EQ.2) CALL LNTP(X, ZT(I), XSECT, Y164, 26, 4)			R1T2243
	IF(ID.EQ.3) CALL LNTP(X, ZT(I), XSECT, Y165, 26, 4)			R1T2244
	IF(ID.EQ.4) CALL LNTP(X, ZT(I), XSECT, Y166, 26, 4)			R1T2245
	IF(ID.EQ.5) CALL LNTP(X, ZT(I), XSECT, Y163A, 26, 4)			R1T2246
	IF(ID.EQ.6) CALL LNTP(X, ZT(I), XSECT, Y164A, 26, 4)			R1T2247
	IF(ID.EQ.7) CALL LNTP(X, ZT(I), XSECT, Y165A, 26, 4)			R1T2248

C	220 ZT(I) = ZT(I) * TOC/0.1	R1T2249
	GO TO 960	R1T2250
C	300 WRITE(6,2002)	R1T2251
	WRITE(6,1000) XMU,ZMU, XML,ZML, ZPTE, ZTHIK	R1T2252
	400 CONTINUE	R1T2253
C	*****	R1T2254
C	*****	R1T2255
C	CARDS 2257 THRU 2335 ARE NOT SHOWN IN THIS LISTING	R1T2256
C	*****	R1T2257
C	*****	R1T2258
C	GO TO 960	R1T2259
C		R1T2334
C		R1T2335
C		R1T2336
C		R1T2337
C		R1T2338
C	700 ROC = RLE	R1T2339
C		R1T2340
	DO 800 I = 1, NCP	R1T2341
	X = XU(I)	R1T2342
	CALL LNTP(X, ZT(I), XT, YT, NXSET, 4)	R1T2343
	CALL LNTP(X, ZS1, XT, YC, NXSET, 4)	R1T2344
	ZS(I) = ZS1 * CLD	R1T2345
	800 CONTINUE	R1T2346
	GO TO 960	R1T2347
C		R1T2348
	900 WRITE(6,2004)	R1T2349
	I4D = ID - 9	R1T2350
	ROC = 0.5 * (AO(I4D)*TOC/.2)**2	R1T2351
C		R1T2352
	DO 950 I = 1, NCP	R1T2353
	X = XU(I)	R1T2354
	IF(X.GT.XTT(ID)) GO TO 910	R1T2355
C		R1T2356
	ZT(I) = AO(I4D) * SQRT(X) + A1(I4D) * X + A2(I4D) * X*X	R1T2357
	+ A3(I4D) * X**3	R1T2358
1	SB(I) = 0.5 * AO(I4D)/SQRT(X) + A1(I4D) + 2.*A2(I4D)*X	R1T2359
1	+ 3.* A3(I4D) * X**2	R1T2360
	GO TO 940	R1T2361
C		R1T2362
	910 X = (1. - X)	R1T2363
	ZT(I) = 0.002 + D1(I4D) * X + D2(I4D) * X*X + D3(I4D) * X**3	R1T2364
	SB(I) = -D1(I4D) - 2.* D2(I4D)*X + 3.* D3(I4D)*X*X	R1T2365
C		R1T2366
	940 ZT(I) = TOC/.2 * ZT(I)	R1T2367
	SB(I) = TOC/.2 * SB(I)	R1T2368
	CALL LNTP(X, ZS1, XSECT, CAMB1, 26, 4)	R1T2369
	ZS(I) = ZS1 * CLD	R1T2370
C		R1T2371
	950 CONTINUE	R1T2372
C		R1T2373
	960 IF(KPRINT(I).EQ.0) RETURN	R1T2374
	WRITE(6,1004) TOC, CLD, ROC	R1T2375
	DO 970 I = 1, NCP	R1T2376
	INV = NCP + 1 - I	R1T2377
		R1T2378

ZUP	= ZT(INV) + ZS(INV)	R1T2379
ZLO	= -ZT(INV) + ZS(INV)	R1T2380
WRITE(6,1005)	XU(INV), ZT(INV), ZS(INV), ZUP, ZLO	R1T2381
970	CONTINUE	R1T2382
C		R1T2383
1000	FORMAT(/5X,*XMU =*,F10.6,3X,*ZMU =*,F10.6,3X,*XML =*,F10.6,	R1T2384
1	3X,*ZML =*F10.6, /	R1T2385
1	3X,*ZPTE =*F10.6,3X,*ZTHIK =*,F10.6 //)	R1T2386
1001	FORMAT(5X, 7F10.6 /)	R1T2387
1002	FORMAT(15,5X,5F10.0)	R1T2388
1003	FORMAT(10F6.0)	R1T2389
1004	FORMAT(10X, *T/C =*F7.4, 10X,*CAMBER =*F7.4, 10X,*L.E.RADIUS =*,	R1T2390
1	F8.5 //T17,*X/C*,T26,*THICKNESS*,T44,*CAMBER*,T60,	R1T2391
2	*UPPER*,T75,*LOWER*)	R1T2392
1005	FORMAT(5X, 5F15.5)	R1T2393
2000	FORMAT(/ 10X, *BICONVEX AIRFOIL SECTION* /)	R1T2394
2001	FORMAT(/10X, *6A SERIES AIRFOIL SECTION* /)	R1T2395
2002	FORMAT(/ 10X, *SUPERCRITICAL AIRFOIL SECTION* /)	R1T2396
2003	FORMAT(/10X, *6 SERIES AIRFOIL SECTION* /)	R1T2397
2004	FORMAT(/10X, *4-DIGIT AIRFOIL SECTION* /)	R1T2398
	RETURN	R1T2399
	END	R1T2400

CC = 00190

C	SUBROUTINE CPUOV(S, A, SWP, IV, CPI, CP, XM)	R1T2402
C	COMPUTES CP FOR AN INFINITELY SHEARED WING	R1T2403
C	COMMON /BLKCPI/ SA(32), SB(32), SC(32), SD(32), SE(32)	R1T2404
C	COMMON /BLKRO7/ XU(32), ZT(32), ZS(32), RCC, ZTE	R1T2405
C	10 COSA = COS(A)	R1T2406
	SINA = SIN(A)	R1T2407
	COSL = COS(SWP)	R1T2408
	SINL = SIN(SWP)	R1T2409
C	20 F1 = SA(IV)	R1T2410
	F2 = SB(IV)	R1T2411
	F3 = SC(IV)	R1T2412
	F4 = SD(IV)	R1T2413
	F5 = SE(IV)	R1T2414
	FX = SQRT((1. - XU(IV))/XU(IV))	R1T2415
	30 DUL = 1.0/(1. + ((F2 + S* F5)/COSL)**2)	R1T2416
C	40 UIOV2 = DUL * (COSA * (1. + F1 * COSL + S* F4 * COSL)	R1T2417
	1 + S* SINA * COSL * (1. + F3/COSL) * FX)**2	R1T2418
	2 + DUL * (COSA * SINL * (F1 + S* F4)	R1T2419
	3 + S* SINA * SINL * (1. + F3/COSL) * FX)**2	R1T2420
	4 + (SINL * COSA)**2 * (1. - DUL)	R1T2421
C	CPI = 1.0 - UIOV2	R1T2422
C	CP = CPI	R1T2423
C	50 CPIO = (1. - SINL * SINL + F1 * F1 + 2.0 * F1 * COSL)	R1T2424
	1 / (1. + (F2/COSL)**2) - SINL * SINL	R1T2425
	CPIO = 1.0 - CPIO	R1T2426
	IF(CPIO.GT.0.0) CPIO = 0.0	R1T2427
C	XMN = XM * COSL	R1T2428
	IF(XM.LE.0.01.OR.XMN.GE.1.0) RETURN	R1T2429
C	XM2 = XM * XM	R1T2430
	BETA = SQRT(1. - XMN * XMN)	R1T2431
C	SINA = SINA/BETA	R1T2432
	60 TEST = (COSL * COSL - CPIO * XMN) * XM2	R1T2433
	IF(TEST.GE.1.0) WRITE(6,1000)	R1T2434
	IF(TEST.GE.1.0) GO TO 200	R1T2435
C	70 B = SQRT(1.0 - TEST)	R1T2436
C	F1 = F1/B	R1T2437
	F4 = F4/BETA	R1T2438
	F3 = F3/B	R1T2439
	80 DUL = 1.0/(1. + ((F2 + S* F5)/(B * COSL))**2)	R1T2440
C	90 UIOV2 = DUL * (COSA * (1. + F1 * COSL + S* F4 * COSL)	R1T2441
	1 + S* SINA * COSL * (1. + F3/COSL) * FX)**2	R1T2442

2	+ DUL * (COSA * SINL * (F1 +S* F4)	R1T2458
3	+S* SINA * SINL * (1. + F3/COSL) * FX)**2	R1T2459
4	+ (SINL * COSA)**2 * (1. - DUL)	R1T2460
C		R1T2461
100	CP = -1.42857/XM2	R1T2462
C		R1T2463
	TEST = 1.0 + 0.2 * XM2 * (1. - UOV2)	R1T2464
	IF(TEST.LE.0.0) GO TO 200	R1T2465
	CP = -CP * (TEST**3.5 - 1.0)	R1T2466
C		R1T2467
200	CONTINUE	R1T2468
	RETURN	R1T2469
1000	FORMAT(10X,*KUCHEMAN-WEBER CORRECTION FACTOR REACHES LIMIT*)	R1T2470
	END	R1T2471

CC = 00070

	OVERLAY (3,3)	R1T2473
	PROGRAM AERO	R1T2474
C		R1T2475
C	AERODYNAMICS	R1T2476
C		R1T2477
	COMMON /BLKOV3/ ITYP, OV3A(4), ALT, SPEED, SWEEP, JPASS, OV3B(2),	R1T2478
1	ITRIM	R1T2479
C		R1T2480
	COMMON /BLKGO1/ G1(200)	R1T2481
C		R1T2482
	COMMON /BLKAO2/ A1(433)	R1T2483
C		R1T2484
	COMMON /BLKCO1/ CL, CD, CM, ALPHA, CDM, CCL, CDR, CDRO, CLT, CDT,	R1T2485
1	DH, FK, DELCL, CMO, DCMCL, XACWB, CLA, ALO,	R1T2486
2	C1(45), CLDB, CLMAX, C2(7), FML1, FML2, CDC,	R1T2487
3	C3(12), XACS, C4(6), CDAFT, C5(5)	R1T2488
C		R1T2489
C		R1T2490
	CB = G1(83)	R1T2491
C		R1T2492
	RNOFT = ABS(ALT) * 10.0**6	R1T2493
	IF(ALT.LT.0.0) GO TO 80	R1T2494
	CALL ATMOS(ALT,T,SIGMA,RHO,THETA,DELTA,CA,AMU,1)	R1T2495
	RNOFT = SPEED*CA*RHO/AMU	R1T2496
80	CONTINUE	R1T2497
C		R1T2498
	70 FMACH=SPEED	R1T2499
C		R1T2500
C		R1T2501
	IF(JPASS.EQ.0) CALL CDR(0.0, FMACH, RNOFT, CDRO)	R1T2502
	IF(JPASS.EQ.1) CALL CDDR1(0.0, FMACH, RNOFT, CDRO)	R1T2503
	CALL CDDR1(CL, FMACH, RNOFT, CDRCL)	R1T2504
	CDR = CDRCL - CDRO	R1T2505
	IF(SPEED.GT.FML1) CDR = 0.0	R1T2506
C		R1T2507
	IF(JPASS.EQ.2) GO TO 30	R1T2508
C		R1T2509
	CALL DMIN(FMACH, RNOFT, CDMIN)	R1T2510
	CALL CLWRT(FMACH)	R1T2511
	RE = RNOFT * CB	R1T2512
	IF(RE.GT.4.0E6) RE = 4.0E6	R1T2513
	RE = RE/1.0E6	R1T2514
C		R1T2515
	CALL CLBRK(FMACH, RE, RNOFT)	R1T2516
	CALL CDLI(FMACH, RNOFT, FK, DELCL, PRIMEK, AKD, AKB)	R1T2517
	CALL CDRG(FMACH, FK, DELCL, CDC)	R1T2518
	CALL CMOW(FMACH, CMO)	R1T2519
	CALL ADJUST(1.0, FMACH, CMO)	R1T2520
C		R1T2521
	CALL WBAL(FMACH, XACWB)	R1T2522
C		R1T2523
	30 CONTINUE	R1T2524
	CALL CDL2(FMACH, CL, FK, DELCL, PRIMEK, AKD, AKB, CDL)	R1T2525
	IF(JPASS.LE.1) CALL AERA(FMACH, CL, ALPHA)	R1T2526
	CALL AERA1(FMACH, CL, ALPHA)	R1T2527
	IF(JPASS.GT.0) GO TO 50	R1T2528

	IF(A1(241).EQ.0.0) GO TO 50	R1T2529
C	DO 40 I = 1, 21	R1T2530
	ANG = I - 1	R1T2531
	CALL AFTCD(ANG, DCD)	R1T2532
	IF (I.EQ.1) WRITE(6,1000)	R1T2533
	40 WRITE(6,1001) ANG, DCD	R1T2534
	50 CALL AFTCD(ALPHA, CDAFT)	R1T2535
	CALL TDRG(ITRIM, FMACH, DCLT, DCDT)	R1T2536
C		R1T2537
	CDM = CDMIN + CDC + CDRO	R1T2538
	CD = CDM + CDL + CDR + CDAFT	R1T2539
C		R1T2540
	CDT = CD + DCDT	R1T2541
	CLT = CL + DCLT	R1T2542
C		R1T2543
C		R1T2544
	JPASS = 2	R1T2545
C		R1T2546
	1000 FORMAT(/// 10X, *FUSLAGE AFT-END UPSWEEP DRAG * //	R1T2547
	1 10X, *WING ANGLE*, 10X, *DELCD* /)	R1T2548
	1001 FORMAT(1X, 2F15.5)	R1T2549
	END	R1T2550
		R1T2551

CC = 00079

C	SUBROUTINE CDDR(CL, XMACH, RNOFT, CDR)	R1T2553
C	COMPUTES DRAG RISE	R1T2554
C	COMMON /BLKC03/ CLTAB(11), TABMCR(11)	R1T2555
C	COMMON /BLKC01/ C1(18), TCD(5), C2(48), FMCRO, FML1, FML2, CDC,	R1T2556
C	1 CLAMCR, CLAML2, C3(12), PO, A3, PL, C4(8)	R1T2557
C	COMMON /BLKG01/ G1(200)	R1T2558
C	COMMON /BLKPRT/ KPRINT(50)	R1T2559
C	ESWMC = G1(49)	R1T2560
C	TOC = G1(46)	R1T2561
C	FOC = G1(111)	R1T2562
C	RNOFT1 = RNOFT / XMACH	R1T2563
C	CALL LNTPI(0.0, FMCRO, CLTAB, TABMCR, 11, 2)	R1T2564
C	CALL FDRG(1.0, RNOFT1)	R1T2565
C	CDFF = TCD(2) + TCD(3)	R1T2566
C	CALL WDRG(1.00001)	R1T2567
C	CDW1 = TCD(4)	R1T2568
C	CALL WDRG(1.01)	R1T2569
C	CDW2 = TCD(4)	R1T2570
C	CDWP = (CDW2 - CDW1) * 100.	R1T2571
C	CDW1 = CDW1 - CDFF	R1T2572
C	XM = 1. - FMCRO	R1T2573
C	A3 = (XM * CDWP - 2. * CDW1) / XM**3	R1T2574
C	A2 = (3. * CDW1 - XM * CDWP) / XM**2	R1T2575
C	PL = 25. * (TOC + 2. * FOC) * COS(ESWMC)**3	R1T2576
C	P1 = A2 - PL	R1T2577
C	FML1 = 0.95	R1T2578
C	IF(FMCRO.GT.0.90) FML1 = FMCRO .05	R1T2579
C	IF(FML1.GT.1.0) FML1 = 1.0	R1T2580
C	FML2 = FML1 + 0.15	R1T2581
C	RNOFT2 = RNOFT*FMCRO/XMACH	R1T2582
C	CALL FORG(FMCRO,RNOFT2)	R1T2583
C	CALL WDRG(FMCRO)	R1T2584
C	CALL CLWBT(FMCRO)	R1T2585
C	CLAMCR = C1(17)	R1T2586
C	RNOFT2 = RNOFT*FML2/XMACH	R1T2587
C	CALL FORG(FML2,RNOFT2)	R1T2588
C	CALL WDRG(FML2)	R1T2589
C	CALL CLWBT(FML2)	R1T2590
C	CLAML2 = C1(17)	R1T2591
		R1T2592
		R1T2593
		R1T2594
		R1T2595
		R1T2596
		R1T2597
		R1T2598
		R1T2599
		R1T2600
		R1T2601
		R1T2602
		R1T2603
		R1T2604
		R1T2605
		R1T2606
		R1T2607
		R1T2608

C	IF(KPRINT(23).EQ.1) WRITE(6,1000) FMCRO, A2, A3, FML1, FML2,	R1T2609
1	CLAMCR, CLAML2, CDW1, CDOFF, CDWP	R1T2610
C		R1T2611
C		R1T2612
	ENTRY CDDR1	R1T2613
	CDR = 0.0	R1T2614
	CDRO = 0.0	R1T2615
	CALL LNTP(CL, XMCR, CLTAB, TABMCR, 11, 2)	R1T2616
	IF(XMACH.GT.1.0) RETURN	R1T2617
	XM = XMACH - XMCR	R1T2618
	XMO = XMACH - FMCRO	R1T2619
	IF(XMO.GT.0.0) CDRO = PD * XMO**2 + A3 * XMO**3	P1T2620
	IF(XM.GT.0.0) CDR = PL * XM**2	R1T2621
	IF(XM.GT.0.12) CDR = 0.0144*PL + 0.24*PL * (XM -.12)	R1T2622
	CDR = CDR + CDRO	R1T2623
C		R1T2624
	RETURN	R1T2625
1000	FORMAT(5X, *CDDR DUMP* /(1X, 7F15.5))	R1T2626
	END	R1T2627
		R1T2628

CC = 0076

	SUBROUTINE CLWBT(SPEED)	R1T2630
C		R1T2631
C	COMPUTES WING-BODY-TAIL LIFT CONTRIBUTION	R1T2632
C		R1T2633
	COMMON /BLKG01/ G1(44), DOB, TOC, CLD, SEXW, ESWMC, G2(30),	R1T2634
1	AR, TR, SPLAN, G3(118)	R1T2635
C		R1T2636
	COMMON /BLKA01 / NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPMLS	R1T2637
C		R1T2638
	COMMON /BLKA02 / SREF, A1(432)	R1T2639
C		R1T2640
	COMMON /BLKC01/ C1(16), CLA, ALO, C2(35), CLAW, CLAB, CLAT, A, B,	R1T2641
1	ADH, C, ABREAK, CLDH, CLPB, CLOB, CLMAX, ABRK,	R1T2642
2	AMAX, DAMAX, DEL, CLS, ARLO, C3(29)	R1T2643
C		R1T2644
	COMMON /BLKCLA/ CLAI(11)	R1T2645
C		R1T2646
C		R1T2647
	CLAW = 0.0	R1T2648
C		R1T2649
	CLAI(1) = SPLAN	R1T2650
	CLAI(2) = TOC	R1T2651
	CLAI(3) = TR	R1T2652
	CLAI(4) = AR	R1T2653
	CLAI(5) = 0.0	R1T2654
	IF(A1(234).EQ.8.) CLAI(5) = 0.0334	R1T2655
	CLAI(6) = CLD	R1T2656
	CLAI(7) = 0.0	R1T2657
	IF(A1(234).EQ.8.) CLAI(7) = 0.09	R1T2658
	CLAI(8) = DOB	R1T2659
	CLAI(9) = 0.0	R1T2660
	IF(A1(234).EQ.8.) CLAI(9) = 1.173763 * TOC	R1T2661
	CLAI(10)=ESWMC	R1T2662
	CLAI(11)= SREF	R1T2663
C		R1T2664
	CALL AER2(SPEED, CLA)	R1T2665
	CLAW = CLA	R1T2666
C		R1T2667
	CLAT = 0.0	R1T2668
	A = 0.0	R1T2669
	B = 0.0	R1T2670
	C = 0.0	R1T2671
	ABREAK = 90.0	R1T2672
C		R1T2673
	IF(NHT.EQ.0) GO TO 30	R1T2674
C		R1T2675
	20 CALL TAIL(SPEED)	R1T2676
	30 CONTINUE	R1T2677
C		R1T2678
	CLAB= 0.0	R1T2679
	IF (NBDYS.EQ.0) GO TO 50	R1T2680
C		R1T2681
	WR = A1(14)	R1T2682
	HB = A1(24)	R1T2683
	AB = A1(64)	R1T2684
	BL = A1(4)	R1T2685

	BLN = A1(84)	R1T2686
	PERIM = 3.14159*SQRT((WB**2 + HB**2)/2.)	R1T2687
	BHPF = WB/HB*PERIM/SQRT(AB)	R1T2688
	AK1 = 0.009942*BHPF - .000379*BHPF**2 + .00001*BHPF**3	R1T2689
	CLAB = AK1 * (BL/BLN)**0.3333 * AB/SREI	R1T2690
C		R1T2691
	50 CLA = CLAW + CLAT + CLAB	R1T2692
C		R1T2693
	CALL AALO(SPEED)	R1T2694
	CALL ADJUST(3.0, SPEFD, ALO)	R1T2695
C		R1T2696
	RETURN	R1T2697
	END	R1T2698

CC = 00069

C	SUBROUTINE AER2(SPEED, CLA)	R1T2700
C	LIFT CURVE SLOPE	R1T2701
C	COMMON /BLKPRT/ KPRINT(50)	R1T2702
	COMMON /BLKCLA/ SPLAN, TOC, TAPER, ARWS, DMSTR, CLD, DM1, DOB,	R1T2703
1	EP SL, SWPMC, SREF	R1T2704
C		R1T2705
C		R1T2706
C		R1T2707
C		R1T2708
	COSZ = COS(SWPMC)	R1T2709
	PI = 3.14159	R1T2710
	TWOD = 1.0	R1T2711
	IF(SPEED.LT.1.0) TWOD = 1. + EPSL/SQRT(1.-SPEED**2)	R1T2712
C		R1T2713
C		R1T2714
	CSUBO = (10.0 + 0.91 * ARWS**3)/(10.0 + ARWS**3)	R1T2715
	ZMSTR = CSUBO + (1.0 - CSUBO) * (1.0 - COSZ)**2	R1T2716
	ZMSTRO = ZMSTR	R1T2717
	ZMSTR = ZMSTR + DMSTR	R1T2718
	ZM1 = 1.0 - 2.0 * TOC * (ARWS**3/(4.0 + ARWS**3))	R1T2719
1	* COSZ**1.5 * (1.0 + 1.5*CLD**1.5)	R1T2720
	ZM10 = ZM1	R1T2721
	IF(ZM10.GT.ZMSTRO) ZM10 = ZMSTRO	R1T2722
	ZM1 = ZM10 + DM1	R1T2723
	IF(ZM1.GT.ZMSTR) ZMSTR = ZM1	R1T2724
	ZM2 = ZM1 + TOC + DM1*0.5	R1T2725
	ZM30 = 1.0 + TOC	R1T2726
	ZM3 = ZM30	R1T2727
	IF(ZM2.GT.1.0) ZM3 = ZM2 + TOC	R1T2728
C		R1T2729
	SIG1=0.0	R1T2730
	SIG2=0.0	R1T2731
	IF(ZM2.NE.ZM1) SIG1 = 0.5*((SPEED -ZM1)/(ZM2 -ZM1))	R1T2732
	IF(ZM3.NE.ZM2) SIG2 = 0.5*(1.0 + (SPEED -ZM2)/(ZM3 -ZM2))	R1T2733
C		R1T2734
	TOCL = 1.0/(4.4*ARWS*COSZ**1.5)	R1T2735
	DTOCL =(TOC - TOCL)/COSZ	R1T2736
	IF(DTOCL.LT.0.0) DTOCL = 0.0	R1T2737
	ARDT = ARWS * DTOCL	R1T2738
	IF(DTOCL.GT.0.07) DTOCL = 0.07	R1T2739
	IF(ARDT.GT.0.1) ARDT = 0.1	R1T2740
	GAMMA = 9.0 * (DTOCL/(1.0 + 0.5 * ARDT))	R1T2741
C		R1T2742
	GAMAO = GAMMA	R1T2743
	IF(ZM30.NE.ZM10) GAMMA = GAMAO *(ZM3-ZM1)/(ZM30-ZM10)	R1T2744
	IF (GAMMA.GT.GAMAO) GAMMA = GAMAO	R1T2745
C		R1T2746
	XF = (16.0 + 3.0*ARWS**2)/(18.0 + 5.0*ARWS**2)	R1T2747
	XKB =(1.0 +DOB)*(1.0 -DOB)**XF	R1T2748
	XKT = 1.0	R1T2749
	IF(SPEED.GE.ZM1.AND.SPEED.LE.ZM2) XKT = 1.0 -(4.0*SIG1*(1.0	R1T2750
1	- SIG1))**3 * GAMMA	R1T2751
	IF(SPEED.GT.ZM2.AND.SPEED.LE.ZM3) XKT = 1.0 -(4.0*SIG2*(1.0	R1T2752
1	- SIG2))**3 * GAMMA	R1T2753
1		R1T2754
	5 CONTINUE	R1T2755

C	Y1	= 2.0 + 0.66667 * SQRT(TAPER) - TAPER**2	R1T2756
	Y	= (1.0 + PI * ARWS)/(3.0 + PI * ARWS) * Y1	R1T2757
	TWOS	= 1.0	R1T2758
	IF(ZMSTR.LT.1.0)	TWOS = 1. + EPSL/SQRT(1.-ZMSTR**2)	R1T2759
C	CLAO	= (TWOS * PI * ARWS)/(TWOS + SQRT(TWOS + (1.0 -	R1T2760
	1	COSZ**1.3334) * (ARWS/(2.0*COSZ)**2))	R1T2761
	IF(SPEED.GT.0.0)	BETAP = (SPEED - ZMSTR) * (1.0 +	R1T2762
	1	(ZMSTR/SPEED)**Y)**2	R1T2763
	Z1	= PI * ARWS / CLAO	R1T2764
	Z1	= 3.0 * Z1 * (Z1 - 1.0) * COSZ**0.6667	R1T2765
	Z	= ZMSTR * CLAO + ARWS**2/Z1	R1T2766
C			R1T2767
C			R1T2768
	IF(SPEED.GT.ZMSTR)	GO TO 10	R1T2769
C			R1T2770
	CLAB	= (.0548311*TWOD*ARWS)/(TWOD + SQRT(TWOD + (1.0 -	R1T2771
	1	COSZ**1.3334 *(SPEED/ZMSTR)**2.667)*(ARWS/(2.*COSZ)**2	R1T2772
	2	))	R1T2773
	GO TO 20		R1T2774
	10	CLAB = 1.0/(57.3 *((ZMSTR/SPEED)**Z /CLAO + BETAP/4.0))	R1T2775
C			R1T2776
C			R1T2777
C			R1T2778
C			R1T2779
	20	CONTINUE	R1T2780
	CLA	= CLAB * XKT * XKB * SPLAN/SREF	R1T2781
C			R1T2782
C			R1T2783
C			R1T2784
	25	IF(KPRINT(11).EQ.0) GO TO 30	R1T2785
	WRITE(6,1000)	SPEED, CLA, CLAB, XKT, XKB, SPLAN, SREF,	R1T2786
	1	ZMSTR, ZM1, ZM2, ZM3, COSZ, CSUBO, ARDT,	R1T2787
	2	TDC, TAPER, ARWS, DMSTR, CLD, DM1, DOB, TWOD, SWPMC	R1T2788
	1000	FORMAT(10X*AER2 DUMP*, 6F15.5 /(19X,6F15.5))	R1T2789
C			R1T2790
	30	RETURN	R1T2791
	END		R1T2792
			R1T2793
			R1T2794

CC = 00095

C	SUBROUTINE TAIL(SPEED)	R1T2796
C	COMPUTES TAIL FLOW FIELD AND LIFT	R1T2797
C	COMMON /BLKCLA/ CLAI(11)	R1T2798
C	COMMON /BLKA02/ SREF, A1(428), HTLE, HTY, HTZ, TINC	R1T2799
C	COMMON /BLKBD1/ XSWPL(11), YRMIN(11)	R1T2800
C	COMMON /BLKG01/ G1(79), AR, TR, SPLAN, CB, YB, XMAC, GZ(16),	R1T2801
C	1 SWPQC, G3(4), XHT, XCRTE, G4(92)	R1T2802
C	COMMON /BLKC01/ C1(37), CDWING(4), C2(12), CLAW, CLAB, CLAT, A, B,	R1T2803
C	1 AOH, C, ASTR, CLOH, C3(31), DEDA, C4(6)	R1T2804
C	COMMON /BLKPRT/ KPRINT(50)	R1T2805
C	** DOWNWASH IS CALCULATED ***	R1T2806
C	ARHT = G1(11)	R1T2807
C	SEXHT = G1(21)	R1T2808
C	SWLHT = A1(389)	R1T2809
C	HTLAM = A1(409)	R1T2810
C	CAMHT = A1(339)	R1T2811
C	HTOC = A1(349)	R1T2812
C	SWMCHT = ATAN(TAN(SWLHT) - 2./ARHT * (1. - HTLAM)/(1. + HTLAM))	R1T2813
C	10 ZKA = 1./AR - 1./(1. + AR**1.7)	R1T2814
C	IF(AR.LT.2.3) ZKA = 0.37 - 0.0567 * AR	R1T2815
C	ZKTR = (10. - 3.0 * TR)/7.0	R1T2816
C	B = SQRT(AR * SPLAN)	R1T2817
C	DEDA = 0.0	R1T2818
C	XLHT = XHT - XMAC	R1T2819
C	IF(XLHT.LE.0.0) GO TO 30	R1T2820
C	HHT = HTZ	R1T2821
C	ZKH = (1. - HHT/B)/(2.0 * XLHT/B)**0.33333	R1T2822
C	COSQ = SQRT(COS(SWPQC))	R1T2823
C	20 DEDAO = 4.44 * (ZKA * ZKTR * ZKH * COSQ)**1.19	R1T2824
C	IF(KPRINT(21).EQ.0) GO TO 21	R1T2825
C	WRITE(6,1000)CLAW,DEDAO,CDWING(1),CDWING(2),CDWING(3),CDWING(4),	R1T2826
C	1 HTLE,A1(281),CB,ZKA,ZKTR,ZKH,COSQ	R1T2827
C	21 CONTINUE	R1T2828
C	CALL AER2(0.1, CLAMO)	R1T2829
C	DEDA = DEDAO * CLAW/CLAMO	R1T2830
C	** DYNAMIC PRESSURE AT THE TAIL ***	R1T2831
C	30 CDO = CDWING(1)+CDWING(2)+CDWING(3)+CDWING(4)	R1T2832
C	XOC = (XHT - XCRTE)/CB	R1T2833

	DQQQ = 0.0	RIT2852
	IF(XOC.LE.0.0) GO TO 50	RIT2853
C		RIT2854
	ZWOC = 0.68 * SQRT(CDO * (XOC + 0.15))	RIT2855
	DQQQO = 2.42 * SQRT(CDO)/(XOC + 0.3)	RIT2856
C		RIT2857
	GAMMA = ATAN(HTZ/(XOC*CB))	RIT2858
	AW = 2.0	RIT2859
	CL = CLAW* AW	RIT2860
	EW = 0.51566 * CL/AR	RIT2861
	Z7C = XOC * TAN(GAMMA + EW - AW/57.3)	RIT2862
	ZOZW = ZOC/ZWOC	RIT2863
C		RIT2864
	IF(ZOZW.GE.1.0) GO TO 50	RIT2865
C		RIT2866
	DQQQ = DQQQO * (COS(1.570796* ZOZW))**2	RIT2867
C		RIT2868
	50 QQQ = 1.0 - DQQQ	RIT2869
C		RIT2870
	** CARRY-OVER FACTORS ZKWB AND ZKBW ARE COMPUTED ***	RIT2871
C		RIT2872
	60 DOBHT = 2.0 * HTY/(2.0 * HTY + SQRT(ARHT * SEXHT))	RIT2873
C		RIT2874
	ZKWB = 1.0028 +.7116*DOBHT +.42*DOBHT**2 -.1366*DOBHT**3	RIT2875
	ZKBW = .0004 +1.2662*DOBHT +.6018*DOBHT**2 +.1263*DOBHT**3	RIT2876
C		RIT2877
	** LIFT-CURVE-SLOPE OF THE EXPOSED SURFACE IS COMPUTED ***	RIT2878
C		RIT2879
	CLAI(1) = SEXHT	RIT2880
	CLAI(2) = HTOC	RIT2881
	CLAI(3) = HTLAM	RIT2882
	CLAI(4) = ARHT	RIT2883
	CLAI(5) = 0.0	RIT2884
	CLAI(6) = CAMHT	RIT2885
C		RIT2886
	CLAI(7) = 0.0	RIT2887
	CLAI(8) = 0.0	RIT2888
	CLAI(9) = 0.0	RIT2889
	CLAI(10) = SWMCHT	RIT2890
C		RIT2891
	70 CALL AER2(SPEED, CLA)	RIT2892
	CLAT = CLA * (ZKWB + ZKBW) * (1. - DEDA) * QQQ	RIT2893
	CLDH = CLA * ZKWB * QQQ	RIT2894
C		RIT2895
	** INDUCED DRAG FACTOR FOR TAIL IS COMPUTED ***	RIT2896
C		RIT2897
	CALL LNTPI(SWLHT, RTSUB, XSWPL, YRMIN, 11, 2)	RIT2898
	RTAIL = RTSUB	RIT2899
	IF(SPEED.LE.0.9) GO TO 80	RIT2900
	ZNDM = 12.0 * (COS(SWLHT)**1.6) * (SPEED - 0.9)	RIT2901
	FNDM = 1.0/(1. + ZNDM + ZNDM*ZNDM)	RIT2902
	RTAIL = RTSUB * FNDM	RIT2903
C		RIT2904
	80 HTK1 = (1.-RTAIL) * 0.01745 /(CLA * ZKWB)	RIT2905
	HTK2 = RTAIL * 0.31831/ARHT * SREF/SEXHT	RIT2906
		RIT2907

C	HTK	= HTK1 + HTK2	R1T2908
			R1T2909
	A	= HTK * CLDH * CLDH	R1T2910
	B	= 2.0 * HTK * CLAT * CLDH	R1T2911
	AOH	= (1.-DEDA) * (A1(279) - TINC)	R1T2912
C			R1T2913
C			R1T2914
	IF(KPRINT(21).EQ.1)	WRITE(6,1000) CLAT, SPEED, CLA, ZKWB, DOBHT,	R1T2915
1		ZKBW, DEDA, QOQ, CLDH, A, B, AOH, DEDA0,	R1T2916
2		RTSUB, RTAIL, HTK1, HTK2, HTK	R1T2917
C			R1T2918
	RETURN		R1T2919
	1000	FORMAT(10X, *TAIL LIFT DUMP* / (1X, 7F15.5))	R1T2920
	END		R1T2921

CC = 00126

C	SUBROUTINE AALO(SPEED)	R1T2923
C	CALCULATE . ZERO LIFT ANGLE OF ATTACK	R1T2924
C	COMMON /BLKA02/ A1(433)	R1T2925
C	COMMON /BLKG01/ G1(44), DOB, TOC, CLD, G2(33), TAP, G3(20),	R1T2926
C	1 ESWQC, G4(98)	R1T2927
C	COMMON /BLKC01/ C4(16), CLA, ALO, C5(35), C1, C2, C3, C6(44)	R1T2928
C	COMMON /BLKPRT/ KPRINT(50)	R1T2929
C		R1T2930
C		R1T2931
C		R1T2932
C		R1T2933
C		R1T2934
C		R1T2935
C		R1T2936
C		R1T2937
C	ALOC = 0.0	R1T2938
C	ALOT = 0.0	R1T2939
C	ALOI = 0.0	R1T2940
C	TWIST = G4(10)	R1T2941
C	RINC = A1(280)	R1T2942
C	DINC = G4(11)	R1T2943
C	TINC = A1(433)	R1T2944
C		R1T2945
C	XMCR = 0.75	R1T2946
C	XMNO = SPEED * COS(ESWQC)	R1T2947
C		R1T2948
C	TOCO = TOC/COS(ESWQC)	R1T2949
C	IF(TOCO.LT.0.1) XMCR = 0.75 + 1.25 * (0.1 - TOCO)	R1T2950
C	DALODC = 5.6	R1T2951
C	IF(XMNO.GT.XMCR) DALODC = 5.6 - 249.0 * (XMNO - XMCR)**2	R1T2952
C	IF(DALODC.LT.0.0) DALODC = 0.0	R1T2953
C		R1T2954
C	ALOC = - DALODC * CLD	R1T2955
C		R1T2956
C		R1T2957
C	200 IF(TWIST.EQ.0.0) GO TO 300	R1T2958
C	IF(SPEED.GT.1.0) GO TO 300	R1T2959
C	TAU = TWIST	R1T2960
C	BETA = SQRT(1.0 - SPEED**2)	R1T2961
C	SWPQCB = 90.0	R1T2962
C	IF(BETA.GT.0.0) SWPQCB = ATAN(TAN(ESWQC)/BETA) * 57.29578	R1T2963
C	DALO = 0.093 - 0.000571*SWPQCB + 0.5761*TAP - 0.2645*TAP**2	R1T2964
C	ALOT = - DALO * TAU	R1T2965
C		R1T2966
C	300 IF(DINC.EQ.0.0.AND.TINC.EQ.0.0) GO TO 400	R1T2967
C	ZKRW = .0004 + 1.2662*DOB + .6018*DOB**2 + .1263*DOB**3	R1T2968
C	ZKWB = 1.0028 + .7116*DOB + .42 *DOB**2 - .1366*DOB**3	R1T2969
C	CLAB = C3 + ZKRW/(ZKRW + ZKWB) * C1	R1T2970
C	ALOI = (CLAB * DINC + C2 *(DINC - TINC))/CLA	R1T2971
C	1 + (RINC - DINC)	R1T2972
C		R1T2973
C	400 ALO = + ALOC + ALOT + ALOI	R1T2974
C	IF(KPRINT(13).GT.0) WRITE(6,1000) ALO, ALOC, ALOT,	R1T2975
C	1 ALOI, SPEED, ESWQC, TOC, CLD, TAU, SWPQCB, DOB,	R1T2976
C	2 TAP, C1, C2, C3, CLAB, ZKRW, ZKWB, DINC,TINC	R1T2977
C	3 ,RINC	R1T2978

C
1000 FORMAT(5X,*AALD DUMP*,5X,*ALO =*,F6.3,5X,*ALOC =*,F6.3,
1 5X,*ALOT =*,F6.3,5X,*ALOI =*,F6.3 /(5X,7F15.5))
RETURN
END

RIT2979
RIT2980
RIT2981
RIT2982
RIT2983

CC = 00061

C	SUBROUTINE CLBRK(SPEED, RE, RNOFT)	R1T2985
C	CALCULATES LIFT BREAK CL AND CLMAX	R1T2986
C		R1T2987
C	COMMON /BLKG01/ G1(45), TOC, CLD, SEXW, G2(31), AR, TR, SPLAN,	R1T2988
1	CB, G3(18), SWPQC, SWPLE, G4(97)	R1T2989
C		R1T2990
C	COMMON /BLKA01/ NBODY5, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	R1T2991
C		R1T2992
C	COMMON /BLKA02/ SREF, A1(432)	R1T2993
C		R1T2994
C	COMMON /BLKPRT/ KPRINT(50)	R1T2995
C	COMMON /BLKC01/ C3(16), CLA, ALO, C4(35), CLAW, CLAB, CLAT, C6(6),	R1T2996
1	CLPB, CLDB, CLMAX, ABRK, AMAX, DAMAX, DEL,	R1T2997
2	CLS, ARLO, C5(29)	R1T2998
C		R1T2999
C	COMMON /BLKMAP/ MAP, TRANS, DY, AMAP(22), BMAP(11)	R1T3000
C	COMMON /BLKCLB/ X1(6), Y1(6), X9(8), Y9(8), XAR(5), YDCL(5)	R1T3001
C		R1T3002
C	COMMON /BLKBD2/ AA(22), BB(22), CC(22), DD(22), XT(22)	R1T3003
C		R1T3004
C	COMMON /BLKBD5/ XDYO(6), YXMN(6), Z1BD5(6,6)	R1T3005
C		R1T3006
C	COMMON /BLKMAX/ XTR(6), YC1(6), YC2(6), XSWP(4), YA(4),	R1T3007
1	YB(4), XDY(8), XM(4), CTAB(8,4), DTAB(8,4),	R1T3008
2	XXCLM(13), YYDY(6), FCLMX(13,6), XXC2(9),	R1T3009
3	YYMACH(5), FDCLMX(9,5), XDY1(9), YXMT(4),	R1T3010
4	ZCLMAX(9,4), XDY2(8), YFOC(6), ZDC1M(8,6),	R1T3011
5	Z2DC1M(8,6)	R1T3012
C	COMMON /BLKMX2/ XSP(8), YDY(6), FDA(8,6), XAB(6), YCO(8),	R1T3013
1	FKVOFM(6,8), XANG(10), YRTQC(7), FRA(10,7)	R1T3014
C	COMMON /BLKMX4/ XC2(6), YAST(9), FDAM(6,9), XCT(6), YM(5), FDAM2(6,5)	R1T3015
C	COMMON /BLKMX5/ XXXCLM(9), FCLMXX(9,6)	R1T3016
C		R1T3017
C		R1T3018
C	DIMENSION XF1(7), YF1(7), YF2(7)	R1T3019
C		R1T3020
C	DATA XF1 / 5.0, 6.0, 6.47712, 6.77815, 6.95424, 7.39794, 8. /,	R1T3021
1	YF1 / -.125, -.11, -.02, -.01, 0., 0.03, 0.035 /,	R1T3022
2	YF2 / -.1063, -.0713, -.055, -.0175, 0., 0.0375, 0.0463/	R1T3023
C		R1T3024
C	IF(SPEED. GE.1.0) GO TO 20	R1T3025
C		R1T3026
C	NI = A1(234)	R1T3027
C	CONCL = A1(276)	R1T3028
C	DYO = AMAP(NI) * TOC	R1T3029
C	IF(DYO.LT.0.8) DYO = 0.8	R1T3030
C	IF(DYO.GT.2.4) DYO = 2.4	R1T3031
C		R1T3032
C	XMN = SPEED * COS(SWPQC)	R1T3033
C	IF(XMN.LT.0.2) XMN = 0.2	R1T3034
C	XMNO = XMN	R1T3035
C	CLIN = CLD/COS(SWPQC)**2	R1T3036
C	IF(XMN.GT.0.7) XMNO = 0.7	R1T3037
C		R1T3038
C	ABRKO = DLNT(DYO, XMNO, XDYO, YXMN, Z1BD5, 6,6, 6, 4,2)	R1T3039
C	ABRK = (ABRKO + (12.05-4.1*XMN)*CLIN) * COS(SWPQC)	R1T3040

	ABRKLO = 2./COS(SWPLE)	RIT3041
	IF(ABRK.LT.ABRKLO) ABRK = ABRKLO	RIT3042
	CLPBO = CLA * ABRK	RIT3043
C		RIT3044
	DCLPB = 0.0	RIT3045
	FPB = 1.54	RIT3046
	IF(XMN.GT.0.5) FPB = 1.54 - 2.9 * (XMN - 0.5)	RIT3047
	IF(CONCL.GT.0.0) DCLPB = FPB * ((CONCL + .0643)*COS(SWPQC))**2	RIT3048
C		RIT3049
	CLPB = CLPBO + DCLPB	RIT3050
C		RIT3051
	DCLDRF = 0.1226 -.00714*SPEED -.12857*SPEED**2	RIT3052
C		RIT3053
	IF(NI.LE.9) DY = AMAP(NI)*TOC +BMAP(NI)*CLD	RIT3054
	IF(NI.GT.9) DY = AMAP(NI)*TOC +1.75*CLD	RIT3055
C		RIT3056
	TRANS = 0.0	RIT3057
	IF(DY.GT.1.65) TRANS = (DY - 1.65)/.4	RIT3058
	IF(DY.GE.2.05) TRANS = 1.0	RIT3059
	IF(SWPLE.GE.0.87) TRANS = 0.0	RIT3060
	CLDB = CLPB + TRANS *(-.0376 -.24414*SPEED -.0685*SPEED**2	RIT3061
I	+ .4149*SPEED**3 + RE * DCLDRF)	RIT3062
C		RIT3063
	GO TO 30	RIT3064
C		RIT3065
20	BETAC = SQRT(SPEED * SPEED - 1.)/TAN(SWPLE)	RIT3066
	CLSB1 = 0.85	RIT3067
	CLSR9 = 0.85	RIT3068
	IF(SWPLE.GT.0.7243) CALL LNTP(SWPLE, CLSB1, X1, Y1, 6, 4)	RIT3069
	IF(SWPLE.GT.0.3665) CALL LNTP(SWPLE, CLSR9, X9, Y9, 8, 4)	RIT3070
	CALL LNTP(AR, DCLAR, XAR, YDCL, 5, 4)	RIT3071
	CLSB9 = CLSB9 + DCLAR	RIT3072
	CLSB = CLSB1 + (CLSR9 - CLSB1) *(BETAC - .1) * 1.25	RIT3073
	CLSB = CLSB + 0.5 * CLD	RIT3074
	CLPB = CLSB	RIT3075
	CLDB = CLSB	RIT3076
C		RIT3077
C	SUPERSONIC MAXIMUM LIFT COEFFICIENT ****	RIT3078
	CNA048 = CLAW * 14.325 * SQRT(SPEED*SPEED -1.)	RIT3079
C		RIT3080
	OM = 1./SPEED	RIT3081
	CLMAX = 0.7722 + 0.3384 * OM +1.1648 * OM*OM -0.8215 * OM**3	RIT3082
C		RIT3083
	CM1 = 1. - CNA048	RIT3084
	IF(CM1.LT.0.0) CM1 = 0.0	RIT3085
C		RIT3086
	CLMAX = CLMAX - 0.048 * CM1	RIT3087
C		RIT3088
	AMAX = 68.5429 -177.2327 * OM +461.9204 *OM*OM	RIT3089
I	-629.4522 * OM**3 +321.4001 * CM**4	RIT3090
C		RIT3091
	AMAX = AMAX +15.8074 -3.0001*CNA048 -12.8073*CNAC48**2	RIT3092
C		RIT3093
	IF(CLMAX.GT.1.25) CLMAX = 1.25	RIT3094
	IF(AMAX.GT.54.5) AMAX = 54.5	RIT3095
C		RIT3096

GO TO 300	RIT3097
30 CONTINUE	RIT3098
C CLMAX = 0.0	RIT3099
NI = A1(234)	RIT3100
DY = AMAP(NI) * TOC	RIT3101
CALL LNTP(TR, C1, XTR, YC1, 6, 4)	RIT3102
CALL LNTP(TR, C2, XTR, YC2, 6, 4)	RIT3103
C C1COS = (C1 + 1.) * COS(SWPLF)	RIT3104
ARUP = 4./C1COS	RIT3105
ARLO = 3./C1COS	RIT3106
C	RIT3107
C	RIT3108
XMT = XT(NI)	RIT3109
IF(XMT.LT.0.3) XMT = 0.3	RIT3110
IF(XMT.GT.0.45) XMT = 0.45	RIT3111
IF(AR.LE.ARLO) GO TO 200	RIT3112
C	RIT3113
CIMAXB = DLNT(DY, XMT, XDY1, YXMT, ZC1MAX, 9, 4, 9, 2,2)	RIT3114
FOC = 5.500 * CLD	RIT3115
DC1MAX = DLNT(DY, FOC, XDY2, YFOC, ZDC1M, 8, 6, 8, 2,2)	RIT3116
IF(XMT.GE.0.35) DC1MAX = DLNT(DY,FOC,XDY2,YFOC,Z2DC1M,8,6,8,2,2)	RIT3117
C	RIT3118
RNCB = ALOG10(RNOFT * CB)	RIT3119
CALL LNTP(RNCB, F1, XF1, YF1, 7, 2)	RIT3120
CALL LNTP(RNCB, F2, XF1, YF1, 7, 2)	RIT3121
DC1RN = F1 + F2 * DY	RIT3122
C	RIT3123
CIMAX = CIMAXB + DC1MAX + DC1RN	RIT3124
C	RIT3125
SWEEP = SWPLE * 57.2956	RIT3126
100 CALL LNTP(SWEEP, A, XSWP, YA, 4, 4)	RIT3127
CALL LNTP(SWEEP, B, XSWP, YB, 4, 4)	RIT3128
C	RIT3129
DYM14 = DY - 1.4	RIT3130
IF(DYM14.LT.0.0) DYM14 = 0.0	RIT3131
IF(DY.GT.2.5) DYM14 = 1.1	RIT3132
C	RIT3133
CLMOC1 = A - B * DYM14	RIT3134
CLMAX = CLMOC1 * CIMAX	RIT3135
C	RIT3136
C = DLNT(DY, SPEED, XDY, XM, CTAB, 8, 4, 8, 2,2)	RIT3137
D = DLNT(DY, SPEED, XDY, XM, DTAB, 8, 4, 8, 2,2)	RIT3138
C	RIT3139
DCLMAX = C + (D - C) * SWEEP/60.	RIT3140
CLMAX = CLMAX + DCLMAX	RIT3141
C	RIT3142
120 DAMAX = DLNT(SWEEP, DY, XSP, YDY, FDA, 8, 6, 8, 2,2)	RIT3143
AMAX = CLMAX/CLAW +ALO +DAMAX	RIT3144
CLS = CLA * (AMAX - 2.*DAMAX - ALO)	RIT3145
GO TO 300	RIT3146
C	RIT3147
C LOW ASPECT RATIO METHOD	RIT3148
C	RIT3149
200 CONTINUE	RIT3150
	RIT3151
	RIT3152

C	ABETA = AR/SQRT(1. - SPEED * SPEED)	R1T3153
	XCLM = ABETA * C1COS	R1T3154
	IF(XMT.LE.0.35.OR.XCLM.GE.2.0) CLMXB = DLNT(XCLM, DY, XXCLM,	R1T3155
1	YYDY, FCLMX, 13, 6, 13, 2,2)	R1T3156
	IF(XMT.GT.0.35.AND.XCLM.LT.2.0) CLMXB = DLNT(XCLM, DY, XXXCLM,	R1T3157
1	YYDY, FCLMXX, 9, 6, 9, 2,2)	R1T3158
C		R1T3159
	C2TAN = (C2 + 1.) * AR * TAN(SWPLE)	R1T3160
	DCLMX = DLNT(C2TAN, SPEED, XXC2, YMACH, FDCLMX, 9, 5, 9, 2,2)	R1T3161
C		R1T3162
	CLMAX = CLMXB + DCLMX	R1T3163
	AST = AR * COS(SWPLE) * (1. + 4.0 * TR*TR)	R1T3164
	AMAXB = 35.0	R1T3165
	IF(XCLM.GT.0.9) AMAXB = 49.8473 - 20.6922*XCLM + 5.0674*XCLM**2	R1T3166
1	- 0.4279*XCLM**3	R1T3167
	IF(XCLM.GT.3.6) AMAXB = 21.0	R1T3168
	IF(C2TAN.LE.4.5) DAM = DLNT(C2TAN, AST, XC2, YAST, FDAM, 6, 9, 6, 4, 2)	R1T3169
	IF(C2TAN.GT.4.5) DAM = DLNT(C2TAN, SPEED, XCT, YM, FDAM2, 6, 5, 6, 4, 2)	R1T3170
C		R1T3171
	AMAX = AMAXB + JAM	R1T3172
C		R1T3173
300	CONTINUE	R1T3174
C	TAIL CONTRIBUTION TO CLMAX *****	R1T3175
C		R1T3176
	DEL = CLAT * 57.3 * SIN(AMAX/57.3) * COS(AMAX/57.3)**2	R1T3177
	CLMAX = CLMAX + DEL	R1T3178
C		R1T3179
	IF(KPRINT(16).EQ.1) WRITE(6,1000) CLPB, CLDB, ABRKO, ABRK,	R1T3180
1	XMN, DYO, DCLPB, TRANS, BETAC, DY, C1, C2, AMAX, ARUP,	R1T3181
2	ARLO, XMT, C1MAXB, DC1MAX, CLMOC1, DCLMAX, DEL,	R1T3182
3	XCLM, CLMXB, C2TAN, DCLMX, CLMAX, DC1RN, RNCB	R1T3183
1000	FORMAT(10X, *CLBRK DUMP* /(1X, 7F15.5))	R1T3184
C		R1T3185
	RETURN	R1T3186
	END	R1T3187
		R1T3188

CC = 00204

C	SUBROUTINE CDLI(SPEED, RNOFT, FK, DELCL, PRIMEK, AKD, AKB)	R1T3190
C	COMPUTES DRAG DUE TO LIFT CONSTANTS	R1T3191
C		R1T3192
C	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	R1T3193
C		R1T3194
C	COMMON /BLKA02/ SREF, A1(223), CBAR(10), YW(10), CAM(10), TOC(10),	R1T3195
1	A2(12), CONCL, A3(25), YW(11), A4(120)	R1T3196
C		R1T3197
C	COMMON /BLKG01/ G1(44), DOB, TOCW, CLD,	R1T3198
1	G3(12), XLESW(10), G4(10), AR, TR, SPLAN, CB,	R1T3199
2	G5(18), SWPQC, SWPLE, G6(11), CBAR2, CLDS, TOCS,	R1T3200
3	G7(83)	R1T3201
C		R1T3202
C	COMMON /BLKBD1/ XSWPL(11), YRMIN(11)	R1T3203
C		R1T3204
C	COMMON /BLKBD2 / AA(22), BB(22), CC(22), DD(22), XT(22)	R1T3205
	COMMON /BLKA16/ XSWP(7), YTR(4), FEP35(7,4), FEP7(7,4),	R1T3206
1	XCLDB(7), YAKB(7)	R1T3207
C		R1T3208
C	COMMON /BLKCO1/ C1(100)	R1T3209
C		R1T3210
C	COMMON /BLKPRT/ KPRINT(50)	R1T3211
C		R1T3212
C	DIMENSION DCURV(13), RCURV(13), XRN(9), YDRT1(9), YDRT2(9)	R1T3213
C		R1T3214
C	DATA DCURV / 2., 4., 6., 10., 20., 30., 40., 50., 60.,	R1T3215
1	80., 100., 130., 200. /,	R1T3216
2	RCURV / 0.17, 0.37, 0.48, 0.6, 0.73, 0.782, 0.81, 0.83, 0.84,	R1T3217
3	0.855, 0.865, 0.874, 0.874 /	R1T3218
	DATA XRN / 0., 5., 7., 10., 20., 40., 100., 300., 600. /,	R1T3219
1	YDRT1 / .07, .07, .05, .025, 5*0.0 /,	R1T3220
2	YDRT2 / .175, .175, .165, .12, .05, .03, .015, .005, 0.0 /	R1T3221
C		R1T3222
C		R1T3223
C	CLM = C1(17) * 57.29578 * SREF/SPLAN	R1T3224
	B02X = YW(1+NPNLS) - YW(1)	R1T3225
	FKL1 = 0.0	R1T3226
	DCL1 = 0.0	R1T3227
	FKL2 = 0.0	R1T3228
	DCL2 = 0.0	R1T3229
	CLM0 = CLM	R1T3230
	CLM2 = C1(77) * 57.29578 * SREF/SPLAN	R1T3231
	RBAR = 0.0	R1T3232
C		R1T3233
C	CLDB = C1(64)	R1T3234
	CALL LNTP(CLDB, AKB, XCLDB, YAKB, 7, 2)	R1T3235
	FAKB = 1.24 - .04 * RNOFT * CB * 1.0E-6	R1T3236
	IF (AKB.LT.1.0) FAKB = 1.0	R1T3237
	AKB = FAKB * AKB * SREF/SPLAN	R1T3238
C		R1T3239
C	B = -1.41 + 1.442*TR - 1.26*TR**2 + .528*TR**3	R1T3240
	C = 0.7125 - 1.497*TR + 1.476*TR**2 - .6909*TR**3	R1T3241
	EO = 1.0 + B * DOB + C * DOB**2	R1T3242
C		R1T3243
C	FMACH = SPEED	R1T3244
		R1T3245

	FMCRO = C1(72)	R1T3246
	FML1 = C1(73)	R1T3247
	FML2 = C1(74)	R1T3248
	IF(FMACH.GT.FMCRO) FMACH = FMCRO	R1T3249
	IF(SPEED.GE.FMCRO.AND.SPEED.LT.FML2) CLMO= C1(76) *	R1T3250
1	57.29578 * SREF/SPLAN	R1T3251
	IF(SPEED.GT.1.0) CALL ADCL(SPEED, CLOPT)	R1T3252
	IF(SPEED.LT.FML2) CALL ADCL(FMACH, CLOPT)	R1T3253
C		R1T3254
	NI = TW(1)	R1T3255
	DO 100 I = 1, NPNLS	R1T3256
C		R1T3257
	TC = TOC(I)	R1T3258
	RLEOC = AA(NI) + BB(NI)*TOC(I) + CC(NI)*TOC(I)**2	R1T3259
1	+DD(NI)*TOC(I)**3	R1T3260
	IF(NI.EQ.8) RLEOC = 0.88216 * TOC(I)**1.606	R1T3261
	RLE = RLEOC * CBAR(I)	R1T3262
	IF(ISWP.EQ.0.OR.I.NE.NPNLS) GO TO 90	R1T3263
	RLEOC = AA(NI) + BB(NI)*TOCS + CC(NI)*TOCS**2 + DD(NI)*TOCS**3	R1T3264
	IF(NI.EQ.8) RLEOC = 0.88216 * TOCS**1.606	R1T3265
	RLE = RLEOC * CBAR2	R1T3266
	TC = TOCS	R1T3267
C		R1T3268
	90 CONTINUE	R1T3269
C		R1T3270
	RNLER = RLE * RNOFT/10.0**3	R1T3271
	COTANS = 5.0 - 6.511 * XLESW(I)	R1T3272
	IF(SWPLE.GT.0.35) COTANS = 1./TAN(XLESW(I))	R1T3273
C		R1T3274
	OMEGA = RNLER * COTANS * SQRT(1. - (FMACH * COS(XLESW(I)))**2)	R1T3275
	CALL LNTP(OMEGA, RT, OCURV, RCURV, 13, 2)	R1T3276
	CALL LNTP(XLESW(I), RMIN, XSWPL, YRMIN, 11, 2)	R1T3277
	IF(RT.LT.RMIN) RT = RMIN	R1T3278
	DRT = 0.0	R1T3279
C		R1T3280
	CALL LNTP(RNLER, DRT1, XRN, YDRT1, 9, 2)	R1T3281
	CALL LNTP(RNLER, DRT2, XRN, YDRT2, 9, 2)	R1T3282
	IF(TC.GT.0.03) DRT = DRT1 * (TC - 0.03)/0.03	R1T3283
	IF(TC.GT.0.06) DRT = DRT1 + DRT2 * (TC - 0.06)/0.06	R1T3284
	RT = RT + DRT	R1T3285
C		R1T3286
	RI = RT + (0.824 - RT) * (CLD + CONCL)/0.6	R1T3287
C		R1T3288
	IF(RI.GT.0.874) RI = 0.874	R1T3289
C		R1T3290
	RBAR = RBAR + RI * (YW(I+1) - YW(I))/BQZX	R1T3291
100	CONTINUE	R1T3292
C		R1T3293
	AT = AR * TR/COS(SWPLE)	R1T3294
	DELR = 0.0482*AT - 0.01102*AT**2 + 0.001197*AT**3	R1T3295
1	-0.00004833*AT**4	R1T3296
C		R1T3297
	R = RBAR + DELR	R1T3298
C		R1T3299
	FK = ((1.-R)/CLMO + R/(3.14159*AR*EO)) * SREF/SPLAN	R1T3300
C		R1T3301

E	= 1./ (3.14159 * AR * FK) * SREF/SPLAN	R1T3302
DELCL	= CLOPT * (1.-E)	R1T3303
PRIMEK	= 0.518/SQRT(AR)	R1T3304
C		R1T3305
SWPC4	= SWPC * 57.29578	R1T3306
IF(SWPC4.GT.60.0)	SWPC4 = 60.0	R1T3307
EP35	= DLNT(SWPC4, TR, XSWP, YTR, FEP35, 7, 4, 7, 2,2)	R1T3308
EP7	= DLNT(SWPC4, TR, XSWP, YTR, FEP7, 7, 4, 7, 2,2)	R1T3309
IF(AR.LE.3.5)	EP = 1. - (1.-EP35) * AR/3.5	R1T3310
IF(AR.GT.3.5)	EP = EP35 + (EP7 - EP35)*(AR-3.5)/3.5	R1T3311
C		R1T3312
EPP	= EP * (1.-DOB**2)	R1T3313
AKD	= 1./ (3.14159 * AR * EPP) * SREF/SPLAN	R1T3314
C		R1T3315
C		R1T3316
RD	= R	R1T3317
H	= 1.1	R1T3318
ARTANS	= AR * TAN(SWPLE)	R1T3319
IF(ARTANS.GT.3.5)	H = 1.1 + 0.1*(ARTANS - 3.5)	R1T3320
IF(SPEED.GE.1.0)	PRIMEK = H/CLM * SREF/SPLAN	R1T3321
C		R1T3322
IF(SPEED.LE.FML2)	GO TO 200	R1T3323
ZNDM	= 12. * (COS(SWPLE)**1.6) * (SPEED-FMCRO)	R1T3324
FNDM	= 1./ (1. + ZNDM + ZNDM**2)	R1T3325
R	= RD * FNDM	R1T3326
FK	= ((1.-R)/CLM + R/(3.14159*AR*EO)) * SREF/SPLAN	R1T3327
DELCL	= CLOPT	R1T3328
C		R1T3329
GO TO 500		R1T3330
C		R1T3331
200	CALL KGIN(CLDB, FK, DELCL, SPEED, FKL1, DCL1)	R1T3332
IF(SPEED.LE.FML1)	GO TO 500	R1T3333
C	DRAG POLAR IS CALCULATED BY LINEAR INTERPOLATION BETWEEN THE	R1T3334
C	LIMITS FML1 AND FML2.	R1T3335
C		R1T3336
ZNDM	= 12. * (COS(SWPLE)**1.6) * (FML2 - FMCRO)	R1T3337
FNDM	= 1./ (1. + ZNDM + ZNDM**2)	R1T3338
RL2	= RD * FNDM	R1T3339
FKL2	= ((1.-RL2)/CLM2 + RL2/(3.14159*AR*EO)) * SREF/SPLAN	R1T3340
CALL	ADCL(FML2, DCL2)	R1T3341
C		R1T3342
CALL	KGIN(0.50, FK, DELCL, FML1, FKL1, DCL1)	R1T3343
C		R1T3344
FK	= FKL1 + (FKL2-FKL1)*(SPEED-FML1)/(FML2-FML1)	R1T3345
DELCL	= DCL1 + (DCL2-DCL1)*(SPEED-FML1)/(FML2-FML1)	R1T3346
C		R1T3347
500	IF(KPRINT(22).EQ.0) RETURN	R1T3348
WRITE(6,1000)	SPEED, FMACH, FK, DELCL, PRIMEK, AKD, AKB,	R1T3349
1	CLDB, CLM, SREF, SPLAN, FMCRO, FML1, FML2,	R1T3350
2	RLE, RNLER, OMEGA, R, RBAR, RT, CLD, CONCL, E, AR, SWPC4,	R1T3351
3	EP, DOB, SWPLE, FKL1, DCL1, FKL2, DCL2, RHIN	R1T3352
4	, (TOC(I), CBAR(I), XLESW(I), YW(I), I = 1, NPNLS),	R1T3353
5	YW(1+NPNS)	R1T3354
C		R1T3355
1000	FORMAT(5X, *DCL1 DUMP* / (1X, 7F15.5))	R1T3356
RETURN		R1T3357

END

RIT3358

CC = 00169

	SUBROUTINE ADCL(SPEED, CLOPT)	RIT3360
C		RIT3361
C	POLAR AXIS DISPLACEMENT	RIT3362
C		RIT3363
	COMMON /BLKA02/ SREF, A1(432)	RIT3364
C		RIT3365
	COMMON /BLKG01/ G1(46), CLD, SEXW, G2(33), SPLAN, G3(20),	RIT3366
1	ESWLE, G4(97)	RIT3367
C		RIT3368
	CONCL = A1(276)	RIT3369
C		RIT3370
	CLOPT = 0.0	RIT3371
	IF (SPEED .GE. 1.0) GO TO 10	RIT3372
	FDC = 0.06 * CLD	RIT3373
C		RIT3374
	IF(CLD.GT.0.0) CLOPT = -.001 +16.934*FDC -216.2697*FDC**2	RIT3375
1	+1781.3562*FDC**3	RIT3376
	IF(CONCL.GT.0.0)CLOPT = CLOPT -.0017 +1.1334*CONCL	RIT3377
1	-1.8498*CONCL**2 +1.0605*CONCL**3	RIT3378
	IF(A1(234).EQ.8.) CLOPT = 0.51951 * CLD**0.75	RIT3379
	GO TO 20	RIT3380
C		RIT3381
10	BETAT = 10.	RIT3382
	IF(ESWLE.GT.0.0) BETAT = SQRT(SPEED**2 -1.)/TAN(ESWLE)	RIT3383
	DELCL = CLD * (0.25 - 0.225 * BETAT)	RIT3384
	IF(BETAT.GE.1.11) DELCL = 0.0	RIT3385
	CLOPT = DELCL	RIT3386
C		RIT3387
20	CLOPT = CLOPT * SPLAN/SREF	RIT3388
	RETURN	RIT3389
	END	RIT3390

CC = 00031

C	SUBROUTINE KGIN(CLDB, AKIN, DECLIN, SPEED, AKOUT, DCLOUT)	R1T3392
C	COMPUTES POLAR USING LEAST-SQUARES CURVE FIT	R1T3393
C	COMMON /BLKPRT/ KPRINT(50)	R1T3394
	DIMENSION CL(11), CDL(11), SA(12), RE(11)	R1T3395
C	DCL = CLDB/10.0	R1T3396
	CL(1) = 0.0	R1T3397
C	DO 20 I = 1, 11	R1T3398
	IF(I.NE.1) CL(I) = CL(I-1) + DCL	R1T3399
	CLX = CL(I)	R1T3400
C	CALL CDDR1(CLX, SPEED, CDR)	R1T3401
C	IF(I.EQ.1) CDRO = CDR	R1T3402
	DCDR = CDR - CDRO	R1T3403
C	40 CDL(I) = AKIN * (CL(I) - DECLIN)**2 + DCDR	R1T3404
	20 CONTINUE	R1T3405
C	CALL PLSQ(CL, CDL, 7, 2, SA, 0, EMAX, ERMS, EMEQ)	R1T3406
C	AKOUT = SA(1)	R1T3407
	DCLOUT = -SA(2)/(2.0 * AKOUT)	R1T3408
	ERR = SA(3) - AKOUT * DCLOUT**2	R1T3409
C	IF(ABS(ERR).LT.0.001.OR.KPRINT(14).EQ.0) GO TO 30	R1T3410
C	WRITE (6, 1000) SPEED, ERR, AKOUT, DCLOUT,	R1T3411
	1 AKIN, DECLIN, (CL(I), CDL(I), I = 1, 7)	R1T3412
	1000 FORMAT(10X *KGIN* 5X, *MACH **F8.5, 2X*ERR **F8.5, 2X, 4F8.5, /	R1T3413
	1 1X, 4F8.5)	R1T3414
C	30 RETURN	R1T3415
	END	R1T3416
		R1T3417
		R1T3418
		R1T3419
		R1T3420
		R1T3421
		R1T3422
		R1T3423
		R1T3424
		R1T3425
		R1T3426
		R1T3427
		R1T3428

CC = 00037

	SUBROUTINE CDL2(SPEED, CL, AEROK, DELCL, PRIMEK, AKD, AKB, CDL)	RIT3430
C		RIT3431
C	COMPUTES DRAG DUE TO LIFT	RIT3432
C		RIT3433
	COMMON /BLKPRT/ KPRINT(50)	RIT3434
C		RIT3435
	COMMON /BLKCO1/ C1(100)	RIT3436
C		RIT3437
	CLPB = C1(63)	RIT3438
	CLDB = C1(64)	RIT3439
C		RIT3440
C		RIT3441
	20 CONTINUE	RIT3442
	CDL = AEROK * (CL - DELCL)**2	RIT3443
	IF(CL.LE.CLPB) GO TO 500	RIT3444
C		RIT3445
C	DRAG DUE TO LIFT ABOVE POLAR BREAK	RIT3446
	CDL = CDL + PRIMEK * (CL - CLPB)**2	RIT3447
	IF(CL.LE.CLDB) GO TO 500	RIT3448
C		RIT3449
C	DRAG DUE TO LIFT ABOVE DRAG BREAK (CLDB)	RIT3450
C		RIT3451
	IF(SPEED.GE.1.0) GO TO 410	RIT3452
	CDPCB = AEROK * (CLDB - DELCL)**2 - AKD * CLDB**2	RIT3453
	1 + PRIMEK * (CLDB - CLPB)**2	RIT3454
C		RIT3455
	DCDB = AKB * (CL - CLDB)**2	RIT3456
	DCDB = DCDB + 0.08 * SQRT(DCDB)	RIT3457
C		RIT3458
	CDL = CDPCB + DCDB + AKD * CL**2	RIT3459
C		RIT3460
	400 IF(SPEED.LT.1.0) GO TO 500	RIT3461
	410 CONTINUE	RIT3462
	CDL = (AEROK - PRIMEK) * (CLPB - DELCL)**2	RIT3463
	1 + PRIMEK * (CL - DELCL)**2	RIT3464
C		RIT3465
C		RIT3466
C		RIT3467
	500 CONTINUE	RIT3468
	IF(KPRINT(14).LE.0) GO TO 50	RIT3469
	WRITE(6,1000) SPEED, CL, CDL, AFROK, DELCL, PRIMEK, CLPB,	RIT3470
	1 CLDB, AKD, AKB	RIT3471
	KPRINT(14) = KPRINT(14) - 1	RIT3472
	50 CONTINUE	RIT3473
	RETURN	RIT3474
	1000 FORMAT (10X, *CDL2 DUMP*/(1X, 7F15.5))	RIT3475
	END	RIT3476

CC = 00047

C	SUBROUTINE AERA(SPFED, CL, ALPHA)	R1T3478
C	CALCULATES ANGLE OF ATTACK	R1T3479
C		R1T3480
C	COMMON /BLKCO1/ C1(100)	R1T3481
C		R1T3482
C	COMMON /BLKGO1/ G1(44), DOB, TOC, CLD, SEXW, SWPMC,	R1T3483
1	G2(30), AR, TR, SPLAN, G3(6), YIX, YOX,	R1T3484
2	SIX, SOX, ARI, C4(4), SWLEI, G5(3), SWPQC,	R1T3485
3	SWPLE, SWPTE, G6(96)	R1T3486
C		R1T3487
C	COMMON /BLKAO2/ SREF, A1(432)	R1T3488
C		R1T3489
C	COMMON /BLKPRT/ KPRINT(50)	R1T3490
C		R1T3491
C	COMMON /BLKAO1/ NBOOYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	R1T3492
C		R1T3493
C	COMMON /BLKT22/ XIN13(10), YIN13(7), ZOUT13(10,7), ZOUT14(10,7)	R1T3494
C	COMMON /BLKMX2/ XSWP(8), YDY(6), FDA(8,6), XAB(6), YCO(8),	R1T3495
1	FKVOFM(6,8), XANG(10), YRTOC(7), FRA(10,7)	R1T3496
C	COMMON /BLKMX6/ XX(12), XY(7), XF(12,7)	R1T3497
C		R1T3498
C	DIMENSION CLTAB(13), ATAB(13)	R1T3499
C	EQUIVALENCE (CLA,C1(17)), (ALO,C1(18)), (CLAW,C1(54)),	R1T3500
1	(CLD,C1(64)), (CLMAX,C1(65)), (AMAX,C1(67)),	R1T3501
2	(DAMAX,C1(68)), (DEL,C1(69)), (CLS,C1(70)),	R1T3502
3	(ARLO,C1(71)), (CDL,C1(6))	R1T3503
C		R1T3504
C	SWEEP=SWPLE*57.29578	R1T3505
C	KPRT = KPRINT(17)	R1T3506
C	CLVM1 = 0.0	R1T3507
C		R1T3508
C	10 ALPHA = CL/CLA + ALO	R1T3509
C	IF(SPEED.GT.1.0) RETURN	R1T3510
C	IF(NPNLS.GT.1) GO TO 250	R1T3511
C	IF(AR.LE.ARLO) GO TO 100	R1T3512
C		R1T3513
C	20 CONTINUE	R1T3514
C		R1T3515
C	IF(KPRT.GT.0) WRITE(6,1000) CL, ALPHA, AR, ARLO, CLMAX, CLS,	R1T3516
1	DAMAX, CLA, ALO	R1T3517
C		R1T3518
C	IF(CL.LE.CLS) GO TO 200	R1T3519
C		R1T3520
C	HIGH ASPECT RATIO LIFT METHOD *****	R1T3521
C		R1T3522
C	DA = 0.0	R1T3523
C	DCL = CLMAX - CLS	R1T3524
C	DA = ((CL - CLS)/DCL)**2 * DAMAX	R1T3525
C	30 ALPHA = ALPHA + DA	R1T3526
C	IF(CL.LE.CLMAX) GO TO 200	R1T3527
C	ALPHA = ALPHA + 5.0	R1T3528
C	IF(ALPHA.GT.90.0) ALPHA = 90.0	R1T3529
C		R1T3530
C	GO TO 200	R1T3531
C	100 CONTINUE	R1T3532
C		R1T3533

```

C      LOW ASPECT RATIO LIFT METHOD      *****      R1T3534
C
AS      = 0.0      R1T3535
Z      = 2. * COS(SWPMC) / AR      R1T3536
HOB      = (AMAX - AS)/114.6 /AR      * 1.5 * (TR+TR**2)/(1.+TR+TR**2) R1T3537
XEP      = 1.0014 -1.969*HOB +3.0021*HOB**2 -2.0072*HOB**3      R1T3538
DCLFP      = (Z + SQRT(1.+Z**2))/(XEP*Z + SQRT(1. +(XEP*Z)**2))      R1T3539
DCLPA      = (DCLFP -1.) * CLAW * 57.3      R1T3540
CLPA      = CLA * 57.3      R1T3541
DCLD      = - CLA * ALD      R1T3542
CLVM1      = CLMAX - CLPA * SIN(AMAX/57.3) * COS(AMAX/57.3)**2 -DCLD      R1T3543
CLVM2      = CLVM1 - DCLPA * SIN(AMAX/57.3) * COS(AMAX/57.3)**2      R1T3544
IF( CLVM1.LE.0.0 ) DAMAX = AMAX - CLMAX/CLA - ALD      R1T3545
IF( CLVM1.LE.0.0 ) CLS = CLA * (AMAX - 2.*DAMAX - ALD)      R1T3546
IF( CLVM1.LE.0.0 ) GO TO 20      R1T3547
C
CO      = TAN(SWPTE)/TAN(SWPLE)      R1T3548
BETA      = SQRT(1. - SPEED**2)      R1T3549
ABETA      = 4.0/TAN(SWPLE) /(1.-CO) * BETA      R1T3550
TOCR      = TOC      R1T3551
IF( A1(234).EQ.9 ) TOCR = 0.0      R1T3552
C
110 CLVOFM = DLNT(ABETA, CO, XAB, YCO, FKVOFM, 6,8,6,2,2)      R1T3553
TS2      = TAN(SWPLE)**2      R1T3554
FM      = SQRT((1. + TS2)/(BETA**2 + TS2))      R1T3555
CLVA      = CLVOFM * FM * SEXW/SREF      R1T3556
C
ATAB(1)= 0.0      R1T3557
DO 120 I = 1, 13      R1T3558
IF( I.GT.1 ) ATAB(I) = ATAB(I-1) + 3.0      R1T3559
ANG      = ATAB(I)* 0.01745      R1T3560
DCLP      = 0.0      R1T3561
IF( ATAB(I).LE.AS ) GO TO 115      R1T3562
C
TIP VORTEX EFFECT      R1T3563
C
HOB      = (ATAB(I) -AS)/114.6 /AR      *1.5 *(TR+TR*TR)/(1.+TR+TR*TR) R1T3564
XEP      = 1.0014 -1.969*HOB +3.0021*HOB**2 -2.0072*HOB**3      R1T3565
IF( HOB.LE.0.0 ) XEP = 1.0      R1T3566
DCLFP      = (Z +SQRT(1.+Z*Z))/(XEP*Z + SQRT(1. + (XEP*Z)**2))      R1T3567
DCLPA      = (DCLFP -1.) * CLAW * 57.3      R1T3568
DCLP      = DCLPA * SIN(ANG)      R1T3569
115 CONTINUE      R1T3570
C
LEADING EDGE EFFECT (RA)      R1T3571
RA      = DLNT(ANG, TOCR, XANG, YRTOC, FRA, 10, 7, 10, 2,2)      R1T3572
C
VORTEX BREAKDOWN EFFECT (FVL)      R1T3573
X      = AR      R1T3574
FVL      = 0.0      R1T3575
IF( X.LT.3.37 ) FVL = DLNT(X,ANG, XX, XY, XF, 12, 7, 12, 4,2)      R1T3576
C
CLV      = (1.-RA) * FVL * CLVA * SIN(ANG)**2 * COS(ANG)      R1T3577
IF( CLV.GT.CLVM2 ) CLV = CLVM2      R1T3578
CLP      = CLPA * SIN(ANG) * COS(ANG)**2      R1T3579
CLTAB(I) = DCLD + CLV + CLP + DCLP      R1T3580
IF( KPRINT(17).EQ.2.AND.I.EQ.1 ) WRITE(6,1002)      R1T3581
R1T3582
R1T3583
R1T3584
R1T3585
R1T3586
R1T3587
R1T3588
R1T3589

```

IF(KPRINT(17).EQ.2) WRITE(6,1001) CLTAB(1), ATAB(1), RA,	R1T3590
1 CLP, CLV, DCLP	R1T3591
C	R1T3592
120 CONTINUE	R1T3593
C	R1T3594
IF(KPRT.GT.0) WRITE(6,1003) CL, ALPHA, AR, ARLO, Z, HOB, DCLPA,	R1T3595
1 CLPA, DCLO, CLVM1, CLVM2, TOCR, CLVA, RA, FVL,	R1T3596
2 ANG	R1T3597
C	R1T3598
125 IF(CLVM1.LE.0.0) GO TO 20	R1T3599
130 CALL LNTP(CL, ALPHA, CLTAB, ATAB, 13, 4)	R1T3600
C	R1T3601
GO TO 200	R1T3602
C	R1T3603
C	R1T3604
C	R1T3605
C	R1T3606
C	R1T3607
290 CONTINUE	R1T3608
300 BETA = SQRT(1. - SPEED**2)	R1T3609
BETAN = BETA * TAN(SWLE1)	R1T3610
A = DLNT(BETAN,SWEEP, XIN13,YIN13,ZOUT13, 10,7,10, 2,2)	R1T3611
EN = DLNT(BETAN,SWEEP, XIN13,YIN13,ZOUT14, 10,7,10, 2,2)	R1T3612
C	R1T3613
C	R1T3614
ACLARU = CLDB/CLA + ALO	R1T3615
EBRK = YIX/(YIX + YOX)	R1T3616
CLB = A * ACLARU**EN * EBRK/ARI * CLAW * 57.29578	R1T3617
310 IF(CL.LE.CLDB) GO TO 200	R1T3618
CLT = CLB + CL - CLDB	R1T3619
ALPHA = (CLT * ARI/(A * CLAW * 57.29578 * EBRK))**(1./EN)	R1T3620
C	R1T3621
IF(KPRT.GT.0) WRITE(6,1004) CL, ALPHA, AR, A, EN, ACLARU,	R1T3622
1 EBRK, CLB, CLDB, CLAW	R1T3623
C	R1T3624
200 CONTINUE	R1T3625
IF(ALPHA.LT.6.0) GO TO 205	R1T3626
CDLA = CL * TAN(ALPHA/57.296)	R1T3627
IF(CDLA.LT.CDL) CDL = CDLA	R1T3628
205 RETURN	R1T3629
C	R1T3630
1000 FORMAT(5X, *AERA DUMP HIGH ASPECT RATIO* /(1X,7F15.5))	R1T3631
1003 FORIAT(5X, *AERA DUMP LOW ASPECT RATIO* /(1X,7F15.5))	R1T3632
1004 FORMAT(5X, *AERA DUMP CRANKED WING* /(1X,7F15.5))	R1T3633
1001 FORMAT(5X,*VORTEX LIFT*, 6F15.5)	R1T3634
1002 FORMAT(///5X,*VORTEX LIFT*, 8X,*CL*,12X,*ALPHA*,13X,*R*, 13X,	R1T3635
1 *CLP*, 11X,*CLV*, 11X, *DCLP*, /)	R1T3636
ENTRY AFRA1	R1T3637
400 KPRT = KPRINT(17) - 1	R1T3638
ALPHA = CL/CLA + ALO	R1T3639
IF(SPEED.GT.1.0) RETURN	R1T3640
IF(NPNLS.GT.1) GO TO 310	R1T3641
IF(AR.LE.ARLO.AND.(CLVM1.GT.0.0) GO TO 125	R1T3642
GO TO 20	R1T3643
C	R1T3644
END	R1T3645

CC = 00168

C	SUBROUTINE AFTCD(ALPHA, CDAFT)	R1T3647
C	CCMPUTE DRAG INCREMENT DUE TO FUSELAGE UPSWEEP	R1T3648
C	COMMON /BLKA02/ A1(433)	R1T3649
C	DIMENSION X(6), Y(4), Z(6,4)	R1T3650
C	DATA X / -5., 0., 5., 10., 15., 20. /,	R1T3651
1	Y / 0., 5., 10., 15. /,	R1T3652
2	Z / 2*0.0, .01, .045, .125, .28, 0., .0005, .013, .054, .142,	R1T3653
3	.33, 0., .004, .0175, .065, .172, .41, 0., .014, .023,	R1T3654
4	.085, .215, .53 /	R1T3655
C	WINC = A1(280)	R1T3656
	BFUS = A1(241)	R1T3657
	AB = A1(243)	R1T3658
	IF(AB.EQ.0.0) AB = 1.0	R1T3659
	BAMX = A1(65)	R1T3660
	SREF = A1(1)	R1T3661
C	CDAFT = 0.0	R1T3662
	IF(BFUS.EQ.0.0) RETURN	R1T3663
	AFUS = ALPHA - WINC	R1T3664
	BMA = BFUS - AFUS	R1T3665
C	DCD = DLNT(BMA, AFUS, X, Y, Z, 6, 4, 6, 4, 2)	R1T3666
	CDAFT = DCD * BAMX/SREF * AB/1.5	R1T3667
C	RETURN	R1T3668
	END	R1T3669
		R1T3670
		R1T3671
		R1T3672
		R1T3673
		R1T3674
		R1T3675
		R1T3676
		R1T3677

CC = 00031

C	SUBROUTINE TDRG(ITRIM, SPEED, DCLT, DCDT)	R1T3679
C	COMPUTES EFFECT OF TRIM	R1T3680
C	COMMON /BLKA02/ A3(280), XLE(11), A4(142)	R1T3681
C	COMMON /BLKA03/ A1(10), XCG, ZCG, CMAC, A2(7)	R1T3682
C	COMMON /BLKG01/ G1(82), CB, YB, XB, CRX, G2(22), XH, OMEGA,	R1T3683
1	G3(90)	R1T3684
C	COMMON /BLKC01/ CL,CD,CM,ALPHA,C1(6),DH,FK,DELCL,CMO,DCMCL,XACWB,	R1T3685
1	C2(39), CLAT, AH, BH, HSTAR, CH, ABREAK, CLDH,	R1T3686
2	CLPB, CLDB, CLMAX, C3(22), XACS, C4(12)	R1T3687
C	DCLT = 0.0	R1T3688
C	DCDT = 0.0	R1T3689
C	CGOCR = (XCG - XLE(1))/CRX	R1T3690
C	XAC = XACWB	R1T3691
C	IF(CL.GT.CLDB) XAC = XACWB + (XACS - XACWB) * ((CL-CLDB)/	R1T3692
1	(CLMAX-CLDB))**2	R1T3693
C	IF(CL.GT.CLMAX) XAC = XACS	R1T3694
C	IF(SPEED.GE.1.0) XAC = XACWB	R1T3695
C	DCMCL = (CGOCR - XACWB) * CRX/CMAC	R1T3696
C	IF(ITRIM.EQ.1) DH = 0.0	R1T3697
C	CLTAIL = CLAT * (ALPHA - HSTAR) + CLDH * CH	R1T3698
C	CLWB = CL - CLTAIL	R1T3699
C	XLT = XH * COS(OMEGA - ALPHA/57.3) /CMAC	R1T3700
C	CM = CMO + (CGOCR - XAC) *CRX/CMAC * CLWB - CLTAIL * XLT	R1T3701
C	IF(CLDH.EQ.0.0.OR.XLT.EQ.0.0) GO TO 60	R1T3702
C	IF(ITRIM.EQ.1) DH = CM / (CLDH * XLT)	R1T3703
C	IF(ITRIM.EQ.1) CM = 0.0	R1T3704
C	60 CONTINUE	R1T3705
C	IF(DH.EQ.0.0) GO TO 100	R1T3706
C	DCLT = DH*CLDH	R1T3707
C	DCDT = AH*DH**2 + BH*DH*(ALPHA - HSTAR)	R1T3708
C	IF(ALPHA.GT.ABREAK) DCDT = DCDT + CH*DH*(ALPHA-ABREAK)	R1T3709
C	100 RETURN	R1T3710
C	END	R1T3711
		R1T3712
		R1T3713
		R1T3714
		R1T3715
		R1T3716
		R1T3717
		R1T3718
		R1T3719
		R1T3720
		R1T3721
		R1T3722
		R1T3723
		R1T3724

CC = 00046

C	SUBROUTINE DMIN(SPEED, RNOFT, CDMIN)	R1T3726
C	MINIMUM DRAG	R1T3727
C	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	R1T3728
C	COMMON /BLKA02/ SREF, A1(103), BASE(10), A2(319)	R1T3729
C	COMMON /BLK01/ C1(18), TCD(5), CDFUS(5), CDBOD(5), C2(59),	R1T3730
C	1 CDMISC, C3(7)	R1T3731
C		R1T3732
C	CDFUS(5) = 0.0	R1T3733
C	CDBOD(5) = 0.0	R1T3734
C	FMISC = A2(128)	R1T3735
C	CALL FDRG(SPEED, RNOFT)	R1T3736
C	CDMISC = (TCD(1) + TCD(2) + TCD(3)) * FMISC * 0.01	R1T3737
C	CALL ADJUST(2,0, SPEED, CDMISC)	R1T3738
C	DO 20 I = 1, NBDYS	R1T3739
C	CALL BDRG(SPEED, BASE(I), SREF, CDB)	R1T3740
C	IF(I.EQ.1) CDFUS(5) = CDB	R1T3741
C	IF(I.GT.1) CDBOD(5) = CDBOD(5) + CDB	R1T3742
C	20 CONTINUE	R1T3743
C	TCD(5) = CDFUS(5) + CDBOD(5)	R1T3744
C	CALL WDRG(SPEED)	R1T3745
C	CDMIN = TCD(1) + TCD(2) + TCD(3) + TCD(4) + TCD(5)	R1T3746
C	CDMIN = CDMIN + CDMISC	R1T3747
C		R1T3748
C	30 RETURN	R1T3749
C	END	R1T3750
		R1T3751
		R1T3752
		R1T3753
		R1T3754
		R1T3755
		R1T3756
		R1T3757
		R1T3758
		R1T3759
		R1T3760
		R1T3761
		R1T3762
		R1T3763

CC = 00038

	SUBROUTINE FDMG(SPEED, RNOFT)	R1T3765
C		R1T3766
C	THIS SUBROUTINE CALCULATES FRICTION, FORM, AND INTERFERENCE	R1T3767
C		R1T3768
	COMMON /BLKA01/ NBOODYS, NNACS, NSURFS, NLT, NVT, ISWP, NPNLS	R1T3769
	COMMON /BLKA02/ SREF, AR, TAPR, SWPLE,	R1T3770
1	BLN(10), BWID(10), BHGT(10), BAWET(10), BQ(10),	R1T3771
2	BNO(10), BAMX(10), BABS(10), BLNS(10), BLBT(10),	R1T3772
3	BASE(10), ELEN(10), EWID(10), EHGT(10),	R1T3773
4	EAWE(10), EAMX(10), EIN(10), EXIT(10), ELNS(10),	R1T3774
5	ELBT(10), EQF(10), ENO(10), CBAR(10), TW,	R1T3775
5	XLEW, YWW, YB, CR, BD2, BFJS, FMISC, AB, AFTAW,	R1T3776
6	CAN(10), TOC(10), AWET(10), SWMT, SPLAN, CONCL,	R1T3777
7	TWIST, FTWIST, WINC, XLE(11), CRW(11), YW(11),	R1T3778
8	XPIVOT, YPIVOT, XAPEX, AFTSW, AFTCB, AFTUC,	R1T3779
9	SBAR(10), TS(10), SCAM(10), STOC(10), SAWET(10),	R1T3780
1	SMTSW(10), SHF(10), SWL(10), SWT(10), STAPR(10),	R1T3781
2	SCR(10), HTLF, HTY, HTZ, HTINC	R1T3782
C		R1T3783
	COMMON /BLKCO1/ C1(18), TCD(5), CDFUS(5), CDBOD(5), CONAC(4),	R1T3784
1	CDWING(4), CDH(4), CDVT(4), CDSURF(4), C2(47)	R1T3785
C		R1T3786
	COMMON /BLKOV3/ OV3A(7), J, OV3B(4)	R1T3787
C		R1T3788
	COMMON /BLKSUR/ SUR(162), TRB(5,10), TRN(5,10), TRU(5,10),	R1T3789
1	TRL(5,10), TRS(5,10)	R1T3790
	COMMON /BLKG01/ FRBOD(10), G1(20), FRNAC(10), G2(73),	R1T3791
1	SWPMT, CBAR2, CLDS, TOCS, SWET, G3(82)	R1T3792
C		R1T3793
	CRITM = C2(19)	R1T3794
C		R1T3795
	XTR = 0.0	R1T3796
	DO 10 I=1,3	R1T3797
	TCD(I) = 0.0	R1T3798
	CDFUS(I) = 0.0	R1T3799
	CDBOD(I) = 0.0	R1T3800
	CONAC(I) = 0.0	R1T3801
	CDWING(I) = 0.0	R1T3802
	CDHT(I) = 0.0	R1T3803
	CDVT(I) = 0.0	R1T3804
	CDSURF(I) = 0.0	R1T3805
10	CONTINUE	R1T3806
C		R1T3807
	BODY CONTRIBUTIONS	R1T3808
	IF (NBOODYS.EQ.0) GO TO 30	R1T3809
	DO 20 I =1,NBOODYS	R1T3810
	IF(J.GT.0) XTR = YRB(J,I) * BLEN(I)	R1T3811
	CALL CFEQ(RNOFT,SPEED,BLEN(I),XTR,CDI)	R1T3812
	CALL FFACT(1, FRBOD(I), 0, 0, SPEED, 0.0, FF)	R1T3813
	DRAG = CDF*BAWET(I)/SREF	R1T3814
	IF (I.EQ.1) GO TO 25	R1T3815
	FI = BQ(I)	R1T3816
	IF(SPEED.GT.1.0) FI = 1.	R1T3817
	CDBOD(1) = CDBOD(1) + DRAG	R1T3818
	CDBOD(2) = CDBOD(2) + DRAG*(FF - 1.)	R1T3819
	CDBOD(3) = CDBOD(3) + DRAG*FF*(FI - 1.)	R1T3820

GO TO 20	RIT3821
25 REFUS = RNOFT*BLEN(1)	RIT3822
CALL IFACT(1, REFUS, CRITH, SPEED, FI)	RIT3823
CDFUS(1) = DRAG	RIT3824
CDFUS(2) = DRAG*(FF-1.)	RIT3825
CDFUS(3) = DRAG*(FF)*(FI-1.)	RIT3826
20 CONTINUE	RIT3827
C	RIT3828
C NACELLE CONTRIBUTIONS	RIT3829
30 IF (NNACS.EQ.0) GO TO 40	RIT3830
DO 35 I = 1, NNACS	RIT3831
IF(J.GT.0) XTR = TRN(J,I) * ELEN(I)	RIT3832
CALL CFEQ(RNOFT, SPEED, ELEN(I), XTR, CDF)	RIT3833
CALL FFACT(2, FRNAC(I), 0, 0, SPEED, 0.0, FF)	RIT3834
DRAG = CDF*EAWET(I)/SREF	RIT3835
FI = EQF(I)	RIT3836
IF(SPEED.GT.1.) FI = 1.	RIT3837
CDNAC(1) = CDNAC(1) + DRAG	RIT3838
CDNAC(2) = CDNAC(2) + DRAG*(FF-1.)	RIT3839
CDNAC(3) = CDNAC(3) + DRAG*FF*(FI-1.)	RIT3840
35 CONTINUE	RIT3841
C	RIT3842
C WING CONTRIBUTION	RIT3843
40 IF (NPNLS.EQ.0) GO TO 50	RIT3844
DO 45 I = 1, NPNLS	RIT3845
IF(J.GT.0) XTR = TRU(J,I) * CBAR(I)	RIT3846
CALL CFEQ(RNOFT, SPEED, CBAK(I), XTR, CDFU)	RIT3847
IF(J.GT.0) XTR = TRL(J,I) * CBAR(I)	RIT3848
CALL CFEQ(RNOFT, SPEED, CBAR(I), XTR, CDFL)	RIT3849
CDF = 0.5 * (CDFU + CDFL)	RIT3850
C	RIT3851
CALL FFACT(3, TOC(I), TW, CAM(I), SPEED, CRITH, FF)	RIT3852
IF(I.EQ.1) CALL IFACT(2, SWMT, CRITH, SPEED, FI)	RIT3853
DRAG = CDF*AWET(I)/SREF	RIT3854
C	RIT3855
IF(ISWP.LT.0.AND.I.EQ.NPNLS) CALL CFEQ(RNOFT, SPEED, CBAK2,	RIT3856
1 0.0, CDF)	RIT3857
IF(ISWP.GT.0.AND.I.EQ.NPNLS) CALL FFACT(3, TOCS, TW, CLDS,	RIT3858
1 SPEED, CRITH, FF)	RIT3859
IF(ISWP.GT.0) CALL IFACT(2, SWPMT, CRITH, SPEED, FI)	RIT3860
IF(ISWP.GT.0.AND.I.EQ.NPNLS) DRAG = CDF * SWET / SREF	RIT3861
C	RIT3862
CDWING(1) = CDWING(1) + DRAG	RIT3863
CDWING(2) = CDWING(2) + DRAG*(FF-1.)	RIT3864
CDWING(3) = CDWING(3) + DRAG*FF*(FI-1.)	RIT3865
45 CONTINUE	RIT3866
C	RIT3867
C SURFACE CONTRIBUTION	RIT3868
50 IF (NSURFS.EQ.1) GO TO 60	RIT3869
NSURF1 = NSURFS - 1	RIT3870
DO 55 I = 1, NSURF1	RIT3871
IF(J.GT.0) XTR = TRS(J,I) * SBAR(I)	RIT3872
CALL CFEQ(RNOFT, SPEED, SBAR(I), XTR, CDF)	RIT3873
CALL FFACT(3, STOC(I), TS(I), SCAM(I), SPEED, 0.0, FF)	RIT3874
CALL IFACT(3, SMTSW(I), CRITH, SPEED, FI)	RIT3875
IF(SPEED.LE.1.) FI = FI * SHF(I)	RIT3876

```

      DRAG = CDF*SAWET(1)/SREF
      IF ( 1.EQ.1.AND.NHT.GT.0) GO TO 100
200  IF ( NHT.EQ.0 ) GO TO 260
      IF ( 1.EQ.2.AND.NVT.GT.0) GO TO 101
201  IF (NVT.EQ.0) GO TO 270
      IF ( 1.GT.2 ) GO TO 102
      GO TO 55
260  IF ( 1.EQ.1.AND.NVT.GT.0) GO TO 101
      IF ( NVT.EQ.0 ) GO TO 102
270  IF ( 1.GT.1 ) GO TO 102
      GO TO 55
100  CDHT(1) = DRAG
      CDHT(2) = DRAG*(FF-1.)
      CDHT(3) = DRAG*FF*(FI - 1.)
      GO TO 200
101  CDVT(1) = DRAG
      CDVT(2) = DRAG*(FF-1.)
      CDVT(3) = DRAG*FF*(FI - 1.)
      GO TO 201
102  CDSURF(1) = CDSURF(1) + DRAG
      CDSURF(2) = CDSURF(2) + DRAG*(FF-1.)
      CDSURF(3) = CDSURF(3) + DRAG*FF*(FI - 1.)
      GO TO 55
55  CONTINUE
60  CONTINUE
      DO 70 I = 1,3
      TCD(I) = CDFUS(I) + CDBOD(I) + CDNAC(I) + CDWING(I)
      +CDHT(I) + CDVT(I) + CDSURF(I)
1    CONTINUE
70  CONTINUE
      RETURN
      ENG

```

```

R1T3877
R1T3878
R1T3879
R1T3880
R1T3881
R1T3882
R1T3883
R1T3884
R1T3885
R1T3886
R1T3887
R1T3888
R1T3889
R1T3890
R1T3891
R1T3892
R1T3893
R1T3894
R1T3895
R1T3896
R1T3897
R1T3898
R1T3899
R1T3900
R1T3901
R1T3902
R1T3903
R1T3904
R1T3905
R1T3906
R1T3907

```

CC = 00143

C	SUBROUTINE CFEG(RNOFT,ZMACH,CBAR,XTR,CF)	R1T3909
C	THIS SUBROUTINE CALCULATES THE SKIN FRICTION COEFFICIENT	R1T3910
C	USING THE WHITE-CHRISTOPH TECHNIQUE.	R1T3911
C	COMMON /BLKPR1/ KPRINT(50)	R1T3912
C	COMMON /BLKA03/ ROUGHK, A1(19)	R1T3913
C	FTURB(X) = T**2*0.430/(ALOG10(RNL*X*T**1.67*F))**2.56	R1T3914
C	FLAM(Y) = 1.328*CFCFIL/SQRT(Y*RNL)	R1T3915
C	FPRIM(X) = 0.43*T*F*F*((ALOG10(RNL *X*T**1.67*F))**(-2.56)	R1T3916
C	1 -1.11178*(ALOG10(RNL *X*T**1.67*F))**(-3.56))	R1T3917
C	IF(KPRINT(19).GT.0) WRITE(6,1001) RNOFT,ZMACH,CLAR,XTR,ROUGHK	R1T3918
C	IF(CBAR.LE.0.0.OR.CBAR.GT.10000.) GO TO 500	R1T3919
C	ZMACH2 = ZMACH*ZMACH	R1T3920
C	T = 1.0/(1.0+ 0.178*ZMACH2)	R1T3921
C	F = 1.0 + 0.03916*ZMACH2*T	R1T3922
C	DXNP1 = 0.0	R1T3923
C	RNCO = 0.0	R1T3924
C	RNL = RNOFT	R1T3925
C	IF(ROUGHK.LE.0.0) GO TO 5	R1T3926
C	RNL = RNOFT * CBAR	R1T3927
C	AK1= 37.587 + 4.615*ZMACH+2.949*ZMACH2+4.132*ZMACH*ZMACH2	R1T3928
C	RNCO = AK1*(CBAR*12./ROUGHK)**1.0489	R1T3929
C	IF (RNL.GT.RNCO) RNL = RNCO	R1T3930
C	RNL = RNL/CBAR	R1T3931
C	5 CONTINUE	R1T3932
C	CFCFIL = (1.0+ 0.1256*ZMACH2)**(-.12)	R1T3933
C	IF (XTR.LE.0.0) GO TO 100	R1T3934
C	DXN = 0.1 * XTR	R1T3935
C	10 DP = FPRIM(DXN)	R1T3936
C	IF (DP.EQ.0.0) GO TO 200	R1T3937
C	DXNP1 = DXN - (DXN * FTURB(DXN) - XTR * FLAM(XTR))/DP	R1T3938
C	IF (DXNP1.LE.0.0) DXNP1 = 0.5 * DXN	R1T3939
C	DX = ABS(DXNP1 - DXN)	R1T3940
C	DY = 0.0001*CBAR	R1T3941
C	IF (DX.LE.DY) GO TO 20	R1T3942
C	DXN = DXNP1	R1T3943
C	GO TO 10	R1T3944
C	20 XP = DXNP1 + CBAR - XTR	R1T3945
C	CF = (XP/CBAR)*FTURB(XP)	R1T3946
C	GO TO 300	R1T3947
C	100 CF = FTURB(CBAR)	R1T3948
C	GO TO 300	R1T3949
C	200 WRITE (6,1000) DXN, RNL	R1T3950
C	CF = 0.0	R1T3951
		R1T3952
		R1T3953
		R1T3954
		R1T3955
		R1T3956
		R1T3957
		R1T3958
		R1T3959
		R1T3960
		R1T3961
		R1T3962
		R1T3963
		R1T3964

300	CFBAR = FTURB(CBAR)	RIT3965
	IF (KPRINT(19).EQ.0) GO TO 400	RIT3966
	WRITE(6,1003) CF, CFBAR, DXNP1, CFCFIL, RNL, RNCO	RIT3967
C		RIT3968
400	RETURN	RIT3969
500	CF = 0.0	RIT3970
	WRITE(6,1002) CBAR	RIT3971
	RETURN	RIT3972
C		RIT3973
1000	FORMAT (10X*SUBROUTINE CFEQ WILL NOT CONVERGE*/ 1X,3F15.7)	RIT3974
1001	FORMAT(10X,*CFEQ INPUT*, 1PE15.4,0P4F15.7)	RIT3975
1002	FORMAT(10X,*CBAR =*,1PE14.5,2X,*OUT OF RANGE IN CFEQ, CF SET EQUAL	RIT3976
	1 TO ZERO* //)	RIT3977
1003	FORMAT(10X,*CFEQ OUTPUT*, 4F15.7, 1P2E15.4)	RIT3978
	END	RIT3979

CC = 00071

C	SUBROUTINE FFACT(ID,GEOM, TYP,CLO,SPEED,CRITP,FF)	RIT3981
C	THIS SUBROUTINE CALCULATES FORM FACTORS FOR EACH COMPONENT.	RIT3982
C	ITYP = TYP	RIT3983
	FF = 1.0	RIT3984
	IF (SPEED.GT.1.0) GO TO 40	RIT3985
C		RIT3986
	GO TO (10,20,30) ID	RIT3987
C		RIT3988
	FUSELAGE OR BODIES (ID=1)	RIT3989
C	10 FR = GEOM	RIT3990
	FF = 1.0 + 60./FR**3.0 + 0.0025*FR	RIT3991
	GO TO 40	RIT3992
C		RIT3993
	NACELLE (ID=2)	RIT3994
C	20 FR = GEOM	RIT3995
	FF = 1.0 + 0.35/FR	RIT3996
	GO TO 40	RIT3997
C		RIT3998
	WING AND SURFACES (ID=3)	RIT3999
C	30 TOC = GEOM	RIT4000
	TOC2= TOC*TOC	RIT4001
	IF (ITYP.LE. 7) FF = 1.0 + 1.44*TOC + 2.0*TOC2	RIT4002
	IF (ITYP.EQ. 9) FF = 1.0 + 1.2 *TOC + 100.*TOC2*TOC2	RIT4003
	IF (ITYP.GE. 10.AND. ITYP.LE.20) FF = 1.0 + 1.68*TOC +3.*TOC2	RIT4004
	IF (ITYP.NE. 8) GO TO 40	RIT4005
	DELTAM = SPEED - CRITM	RIT4006
	IF (DELTAM.GE.0.0) ZK = 0.202	RIT4007
	IF (DELTAM.LE.-.2) ZK = 0.12	RIT4008
	IF (DELTAM.GT.-.2.AND.DELTAM.LY.0.0) ZK = 0.202 + .8972*DELTAM	RIT4009
	1 +2.1944*DELTAM**2 -1.2340*DELTAM**3	RIT4010
	FF =ZK*CLO/.4+1.44*TOC + 2.0*TOC2 +1.	RIT4011
C		RIT4012
	40 RETURN	RIT4013
	END	RIT4014
		RIT4015
		RIT4016

CC = 00036

C	SUBROUTINE IFACT(ID,PARAM,CRITM,SPEED,FI	RIT4018
C	THIS SUBROUTINE CALCULATES INTERFERENCE FACTORS FOR THE	RIT4019
C	FUSELAGE AND LIFTING SURFACES.	RIT4020
C		RIT4021
C	DIMENSION XMACH(7),REFUS(9), WBR(63)	RIT4022
C		RIT4023
	DATA XMACH / 0.25,0.4,0.6,0.7,0.8,0.85,0.9 /,	RIT4024
1	REFUS / 3.0,10.,20.,30.,40.,60.,80.,200.,100000./,	RIT4025
1	WBR / 1.063,1.02,.979,.955,.926,.902,.867,	RIT4026
1	1.076,1.036,.996,.972,.942,.92,.884,	RIT4027
1	1.066,1.059,1.02,.996,.966,.943,.91,	RIT4028
1	1.044,1.049,1.037,1.013,.988,.968,.939,	RIT4029
1	0.993,1.018,1.032,1.015,1.003,.993,.973,	RIT4030
1	0.940,0.992,1.018,1.014,1.014,1.01,1.007,	RIT4031
1	0.932,0.984,5*1.015,	RIT4032
1	0.924,0.976,5*1.015,0.924,0.976,5*1.015 /	RIT4033
C		RIT4034
	FI = 1.0	RIT4035
	IF (SPEED.GT.1.0) GO TO 30	RIT4036
C		RIT4037
	GO TO (10,20,20) ID	RIT4038
C		RIT4039
C	FUSELAGE CORRELATION FACTOR (RWB)	RIT4040
C	10 PAR10 = PARAM/10.0**6	RIT4041
	XM1 = CRITM - .1	RIT4042
	XM = SPEED	RIT4043
	IF(XM.GT.XM1) XM = XM1	RIT4044
C		RIT4045
	FI = DLNT(XM, PAR10, XMACH, REFUS, WBR, 7,9,7,2,2)	RIT4046
	RWB = FI	RIT4047
	GO TO 30	RIT4048
C		RIT4049
C	LIFTING SURFACES	RIT4050
C	20 A = COS(PARAM)	RIT4051
	IF (A.GE.0.75) RLS = -.9946 + 5.3568*A -4.5389*A*A +1.2487*A**3	RIT4052
	IF (A.GT.0.65.AND.A.LT.0.75) RLS=.9966 -.752*(.75-A)	RIT4053
	IF (A.LE.0.65) RLS = 0.9214	RIT4054
	B = SPEED - 0.25	RIT4055
	IF (B.LT.0.0) B=0.0	RIT4056
	RLS = RLS - .0015 + .1818*B - .2756*B*B + 1.0677*B**3	RIT4057
	FI = RLS*RWB	RIT4058
	IF (ID.EQ.3) FI = RLS	RIT4059
C		RIT4060
	30 RETURN	RIT4061
	END	RIT4062
		RIT4063

CC = 00046

C	SUBROUTINE WDRG(FMACH)	RIT4065
C	COMPUTES TOTAL AIRPLANE WAVE DRAG	RIT4066
C	COMMON /BLKA01/ NBODY5, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	RIT4067
C	COMMON /BLKA02/ SREF, A1(53), BND(10),	RIT4068
1	BAMX(10), BABS(10), BLNS(10),	RIT4069
2	BLBT(10), BASE(10), E1(40), EAMX(10), EIN(10),	RIT4070
3	EXIT(10), ELNS(10), ELBT(10), EQF(10), ENQ(10),	RIT4071
4	W1(95), S1(30), STOC(10), S2(30),	RIT4072
5	SWL(10), SWT(10), STAPR(10), SCR(10), S3(4)	RIT4073
C	COMMON /BLKG01/ FRBCD(10), ARS(10), SSEX(10), FRNAC(10), SWPLE,	RIT4074
1	SWPQC, SWPMC, SWPTE, DOB, TOCW, CLD, SEXW, B1(31),	RIT4075
2	AR, TAPR, SPLAN, G1(20), ESWLE, ESWTE, G2(96)	RIT4076
C	COMMON /BLKWPD/ WPD(9)	RIT4077
C	COMMON /BLKBTA/ BETA	RIT4078
C	COMMON /BLKCO1/ C1(18), TCD(5), CDFUS(5), CDBOD(5), CDNAC(4),	RIT4079
1	CDWING(4), CDHT(4), CDVT(4), CDSURF(4), C2(47)	RIT4080
C	TCU(4) = 0.0	RIT4081
C	CDFUS(4) = 0.0	RIT4082
C	CDBOD(4) = 0.0	RIT4083
C	CDNAC(4) = 0.0	RIT4084
C	CDWING(4) = 0.0	RIT4085
C	CDHT(4) = 0.0	RIT4086
C	CDVT(4) = 0.0	RIT4087
C	CDSURF(4) = 0.0	RIT4088
C	IF(FMACH.LE.1.0) RETURN	RIT4089
C	BETA = SQRT(FMACH**2 -1.)	RIT4090
C	BODY CONTRIBUTIONS	RIT4091
C	IF(NBODY5.EQ.0) GO TO 100	RIT4092
C	DO 50 I = 1, NBODY5	RIT4093
C	CALL CDWN(BAMX(I), BLNS(I), 0.0, CDW1)	RIT4094
C	REX = SQRT(BABS(I)/3.14159)	RIT4095
C	CALL COWT(BAMX(I), BLBT(I), REX, CDW2)	RIT4096
C	IF(I.EQ.1) CDFUS(4) = (CDW1 + CDW2) * BAMX(I)/SREF	RIT4097
C	IF(BND(I).EQ.0.0) BND(I) = 1.0	RIT4098
C	IF(I.GT.1) CDBOD(4) = CDBOD(4) + (CDW1+CDW2)*BAMX(I)/SREF*BND(I)	RIT4099
50	CONTINUE	RIT4100
C	NACELLE CONTRIBUTION	RIT4101
C	100 IF(NNACS.EQ.0) GO TO 200	RIT4102
C	DO 100 I = 1, NNACS	RIT4103
C	RIN = SQRT(EIN(I)/3.141593)	RIT4104
C	RFX = SQRT(EXIT(I)/3.141593)	RIT4105
		RIT4106
		RIT4107
		RIT4108
		RIT4109
		RIT4110
		RIT4111
		RIT4112
		RIT4113
		RIT4114
		RIT4115
		RIT4116
		RIT4117
		RIT4118
		RIT4119
		RIT4120

C	CALL CDWN(EAMX(I), ELNS(I), RIN, CDW1)	R1T4121
	CALL CDWN(EAMX(I), ELBT(I), REX, CDW2)	R1T4122
C	IF(ENO(I).EQ.0.0) ENO(I) = 1.0	R1T4123
	CDNAC(4) = (CDW1 + CDW2) * EAMX(I)/SREF * ENO(I) + CDNAC(4)	R1T4124
150	CONTINUE	R1T4125
C	WING CONTRIBUTION	R1T4126
C	200 WPD(1) = AR * (1.-DOB)**2 * SPLAN/SEXW	R1T4127
	WPD(2) = TAPR/(1.-DOB*(1.-TAPR))	R1T4128
	WPD(3) = ESWLE	R1T4129
	WPD(4) = ESWTE	R1T4130
	WPD(5) = FMACH	R1T4131
	WPD(6) = SEXW/SREF	R1T4132
	WPD(7) = W1(11)	R1T4133
	WPD(8) = CLD	R1T4134
	WPD(9) = TOCW	R1T4135
C	CALL CDWW(CDW1)	R1T4136
	CDWING(4) = CDW1	R1T4137
C	250 IF(NSURFS.LE.1) GO TO 300	R1T4138
	DO 290 J = 2, NSURFS	R1T4139
	I = J-1	R1T4140
C	WPD(1) = ARS(I)	R1T4141
	WPD(2) = STAPR(I)	R1T4142
	WPD(3) = SWL(I)	R1T4143
	WPD(4) = SWT(I)	R1T4144
	WPD(6) = SSEX(I)/SREF	R1T4145
	WPD(7) = S1(10+I)	R1T4146
	WPD(8) = S1(20+I)	R1T4147
	WPD(9) = STOC(I)	R1T4148
C	CALL CDWW(CDW1)	R1T4149
	IF(1.EQ.1.AND.NHT.GT.0) CDHT(4) = CDW1	R1T4150
	IF(NHT.EQ.0) GO TO 260	R1T4151
	IF(1.EQ.2.AND.NVT.GT.0) CDVT(4) = CDW1	R1T4152
	IF(NVT.EQ.0) GO TO 270	R1T4153
	IF(1.GT.2) CDSURF(4) = CDSURF(4) + CDW1	R1T4154
	GO TO 290	R1T4155
C	260 IF(1.EQ.1.AND.NVT.GT.0) CDVT(4) = CDW1	R1T4156
	IF(NVT.EQ.0) GO TO 280	R1T4157
	270 IF(1.GT.1) CDSURF(4) = CDSURF(4) + CDW1	R1T4158
	GO TO 290	R1T4159
C	280 CDSURF(4) = CDSURF(4) + CDW1	R1T4160
	290 CONTINUE	R1T4161
C	300 TCD(4) = CDFUS(4) + CDBOD(4) + CDNAC(4) + CDWING(4) + CDHT(4)	R1T4162
	1 + CDVT(4) + CDSURF(4)	R1T4163
C	RETURN	R1T4164
		R1T4165
		R1T4166
		R1T4167
		R1T4168
		R1T4169
		R1T4170
		R1T4171
		R1T4172
		R1T4173
		R1T4174
		R1T4175
		R1T4176

END

RIT4177

CC = 00113

C	SUBROUTINE CDWN(AMAX, XLNOS, RIN, CDW)	R1T4179
C	NOSE WAVE DRAG (BODY COMPONENT)	R1T4180
C	COMMON /BLKBTA/ BETA	R1T4181
C	DIMENSION XLD(7), YBETA(4), FCDW(7,4)	R1T4182
C	DATA XLD / 1., 2., 3., 4., 6., 8., 12. /,	R1T4183
1	YBETA/ 0.0, 0.32, .458, 0.6633 /,	R1T4184
2	FCDW / .1656,.0617,.0373,.0287,.0198,.0149,.0083,	R1T4185
3	.2376, .1256, .088, .061, .0308, .0177, .0082,	R1T4186
4	.2952, .1667, .1067, .066, .0311, .0175,.0081,	R1T4187
5	.3816, .2010, .1031, .0624,.0297, .0172,.0079 /	R1T4188
C	CDW = 0.0	R1T4189
	IF(AMAX.LE.0.0) GO TO 50	R1T4190
	DMAX=SQRT(AMAX/0.785398)	R1T4191
	ELOD=XLNOS/DMAX	R1T4192
	IF(RIN.GT.0.0) GO TO 40	R1T4193
	IF(ELOD.LE.0.0) GO TO 50	R1T4194
C	IF(BETA.GE.0.6633) GO TO 20	R1T4195
C	10 CDW = DLNT(ELOD,BETA, XLD, YBETA, FCDW, 7,4,7, 2,2)	R1T4196
	GO TO 50	R1T4197
C	20 X = BETA/SQRT(1.+ ELOD**2)	R1T4198
	CDW = (1.2 + 1.15 * X)/(1. + 1.9 * X) /(1. + ELOD**2)	R1T4199
	GO TO 50	R1T4200
C		R1T4201
C		R1T4202
C	FOR A BODY WITH AN OPEN NOSE, SUCH AS A NACELLE, THE CONTRIBUTION	R1T4203
C	OF THE NOSE TO THE TOTAL BODY WAVE DRAG COEFFICIENT IS COMPUTED AS	R1T4204
C	INDICATED BELOW. (THE EXPRESSIONS USED FOR OPEN AND CLOSED BODY	R1T4205
C	BOATTAIL CONTRIBUTIONS DO NOT REQUIRE THIS DISTINCTION--SEE CDWT.)	R1T4206
C		R1T4207
C	40 SQRTB = SQRT(BETA)	R1T4208
	IF(SQRTB.LT.0.8144) SQRTB = 0.8144	R1T4209
	CDW = ((1.-2.*RIN/DMAX)/ELOD)**1.5/SQRTB	R1T4210
C		R1T4211
C	50 RETURN	R1T4212
	END	R1T4213
		R1T4214
		R1T4215
		R1T4216
		R1T4217
		R1T4218
		R1T4219
		R1T4220
		R1T4221

CC = 00043

C	SUBROUTINE CDWT(AMAX, XLAFT, REX, CDW)	R1T4223
C		R1T4224
C	BOATTAIL WAVE DRAG (BODY COMPONENT)	R1T4225
C		R1T4226
C	COMMON /BLK8TA/ BETA	R1T4227
C		R1T4228
C	DIMENSION DBOD(5), CDWBT(5)	R1T4229
C		R1T4230
	CDW = 0.0	R1T4231
	IF(AMAX.LE.0.0) RETURN	R1T4232
	IF (XLAFT .LE. 0.0001) RETURN	R1T4233
	DMAX=SQRT(AMAX/0.785398)	R1T4234
	ELOD=XLAFT/DMAX	R1T4235
	X=BETA/ELOD	R1T4236
	Y=X*X	R1T4237
	Z=ELOD*ELOD	R1T4238
C		R1T4239
C		R1T4240
C	BOATTAIL WAVE DRAG IS A FUNCTION OF BOATTAIL FINENESS RATIO, BASE	R1T4241
C	DIAMETER/MAXIMUM DIAMETER, AND MACH NUMBER. COMPUTE THE BOATTAIL	R1T4242
C	WAVE DRAG COEFFICIENT AT FOUR DBOD(I) VALUES AND INTERPOLATE TO	R1T4243
C		R1T4244
	DBOD(1)=0.0	R1T4245
	CDWBT(1) = (1.165 -.5112*X -.5372*Y +.3964*X*Y)/Z	R1T4246
	IF(X.GT.1.0) CDWBT(1) = 0.513/X/Z	R1T4247
C		R1T4248
	DBOD(2)=0.4	R1T4249
	CDWBT(2) = (1.067 -1.709*X +1.6632*Y -.686*X*Y)/Z	R1T4250
	IF(X.GT.1.0) CDWBT(2) = 0.3352/X/Z	R1T4251
C		R1T4252
	DBOD(3)=0.6	R1T4253
	CDWBT(3) = (0.7346 -1.4618*X +1.5795*Y -.6542*X*Y)/Z	R1T4254
	IF(X.GT.1.0) CDWBT(3) = 0.1980/X/Z	R1T4255
C		R1T4256
	DBOD(4)=0.8	R1T4257
	CDWBT(4) = (0.2555 -.5008*X +.5024*Y -.2077*X*Y)/Z	R1T4258
	IF(X.GT.1.0) CDWBT(4) = 0.0494/X/Z	R1T4259
C		R1T4260
	CDWBT(5)=0.0	R1T4261
	DBOD(5)=1.0	R1T4262
	BDOD=2.0*REX/DMAX	R1T4263
	CALL LNTP(BDOD, CDW, DBOD, CDWBT, 5, 2)	R1T4264
C		R1T4265
	RETURN	R1T4266
	END	R1T4267

CC = 00045

	SUBROUTINE CDWW(CDOSR)	R1T4269
C	PROCEDURE FOR WING PRESSURE DRAG	R1T4270
C		R1T4271
	COMMON /BLKMPD/ AR, ZLAM, ZLE, ZTE, ZM, SOSR, TYPE, CAM, TOC	R1T4272
C		R1T4273
	COMMON /BLKPRT/ KPRINT(50)	R1T4274
	COMMON /BLKBD2/ AA(22), BB(22), CC(22), DD(22), XT(22)	R1T4275
C		R1T4276
	10 N = TYPE	R1T4277
C		R1T4278
	HOT = CAM * 0.055 / TOC	R1T4279
	XTOC=XT(N)	R1T4280
	TOC2= TOC**2	R1T4281
	ROOT=(AA(N)+BB(N)*TOC+CC(N)*TOC2+DD(N)*TOC2*TOC)/TOC	R1T4282
	IF (N.EQ.8) ROOT= 0.88216*(TOC)**1.606/TOC	R1T4283
C		R1T4284
C		R1T4285
C	CALCULATION OF QUANTITIES THAT ARE FUNCTIONS OF GEOMETRY.	R1T4286
C		R1T4287
	17 ZKW =1.0	R1T4288
	TOC2 = TOC**2	R1T4289
	TOC13 = TOC**0.33333333	R1T4290
	TOC53 = TOC**1.66666667	R1T4291
	ARW = AR*TOC13	R1T4292
	ARW3 = TOC*AR**3	R1T4293
	DEN1 = 1.0/ARW+ARW3	R1T4294
	DEN2 = 2.0/ARW3+1.0	R1T4295
	ZKT = 1.+4.*(1.5-XTOC*(1.+5*SQRT(ROOT)))**2	R1T4296
	1 -0.25 * SQRT(ROOT) * (1.-XTOC)**2	R1T4297
	ZKC = 1.0 + 2.5* HOT **2	R1T4298
	ZKB = 1.069	R1T4298.1
	IF(ZKW.EQ.1.2) ZKB = 1.0	R1T4298.2
	TR1 = 0.5/(1.+ZLAM)**2	R1T4299
	TR2 = 1./(1.+ZLAM**2)**2	R1T4300
	ZLAM1 = 1. + TR2 * (TAN(ZLE) + TAN(ZTE))**2	R1T4301
	IF(ZLE.GE.ZTE) ZKP = (COS(ZLE) + TR1 * (TAN(ZLE)**2	R1T4302
	1 - TAN(ZTE)**2))/ZLAM1	R1T4303
	IF(ZTE.GT.ZLE) ZKP = (COS(ZTE) + TR1 * (TAN(ZTE)**2	R1T4304
	1 - TAN(ZLE)**2))/ZLAM1	R1T4305
C		R1T4306
	ZK = (2.0/DEN1 + 3.33/DEN2)*ZKP	R1T4307
	X = ARW*ZKP	R1T4308
	IF(ARW.GE.1.0) X= SQRT(ZKP/(1.7321*(1.0-ZKP)+1.0/(ZKP*ARW**2)))	R1T4309
C		R1T4310
C	THIS DO LOOP SOLVES FOR *ARE* BY NEWTON'S METHOD.	R1T4311
C		R1T4312
	DO 100 I=1,10	R1T4313
	X3 = X**3	R1T4314
	X4 = X**4	R1T4315
	F = (2.0*X)/(X4+1.0) + (3.33*X3)/(X3+2.0) - ZK	R1T4316
	FP = (2.0-6.0*X4)/((X4+1.0)**2) + (20.0*X**2)/((X3+2.0)**2)	R1T4317
	FFP = F/FP	R1T4318
	IF(KPRINT(20).EQ.2 .AND. I.EQ.1) WRITE(6,90)	R1T4319
C		R1T4320
	IF(KPRINT(20).EQ.2) WRITE(6,95) I, X, F, FP, FFP	R1T4321
C		R1T4322

	X = X-FFP	R1T4323
	IF (ABS(FFP).LE.0.0001) GO TO 110	R1T4324
100	CONTINUE	R1T4325
110	IF(KPRINT(20).EQ.2) WRITE(6,36)	R1T4326
C		R1T4327
	IF(KPRINT(20).EQ.2) WRITE(6,45) ZLE, ZTE, TOC2, TOC13, TOC53,	R1T4328
1	ARW, ARW3, DEN1, DEN2, ZKT, ZKC, ZLAM1, ZKP	R1T4329
C		R1T4330
C		R1T4331
	BETA2 = ZM**2-1.0	R1T4332
	BETA = SQRT(BETA2)	R1T4333
	DIV = BETA/TOC2	R1T4334
	BETAW = BETA/TOC13	R1T4335
	BETAW2= BETAW**2	R1T4336
	BETAW3= BETAW*BETAW2	R1T4337
	CD1 = 0.0	R1T4338
	CD2 = 0.0	R1T4339
	CDA = TOC2*(2.0+3.333/(ZKW**2.8))	R1T4340
	ZKLIN = CDA/TOC2	R1T4341
	IF (ZM.GT. 1.0) CD1 = CDA/BETA	R1T4342
	TEST1 = 1.0/COS(ZLE) + 0.01	R1T4343
	IF (ZM.GT. TEST1) CD2 = CDA/SQRT(BETA2-TAN(ZLE)**2)	R1T4344
	ARE=X	R1T4345
	TEST = ABS(TAN(ZLE))	R1T4346
	UEXP = (1.+2.* ZLAM**0.33333)	R1T4347
	FB = 0.3 + 0.7*ZKP**UEXP	R1T4348
	XM = 0.5* (1.+ZLAM**2 *(2.-ZLAM)**3)	R1T4349
	Z = COS(ZLE) + COS(ZTE)	R1T4350
	IF(BETA.GT.TEST) XM = (TEST/BETA)**Z	R1T4351
130	ZML = SQRT (TEST**2+1.0)	R1T4352
	ARE3 = ARE**3	R1T4353
	ZKW1 = ZKW**3.8	R1T4354
	T1 = 1./ARE /(1.+(1.+ZLAM) * FB * BETAW**(1.+ZKW))	R1T4355
	T2 = ARE3 /(1.+0.33333 *ARE3 *BETAW**4)	R1T4356
	T3 = 2./ARE3 /(1.+(.66667+ZLAM) *BETAW**(1.+ZKW1) *FB)	R1T4357
	T4 = 1./(1.+3.* ARE * BETAW**4)	R1T4358
	ZFEXP = FB**XM	R1T4359
	FBXM=FB**XM	R1T4360
	D1 = ZKB * ZKW * BETAW * FBXM +(T1+T2)	R1T4361
	D2 = ZKB * ZKW1 * BETAW * FBXM +(T3+T4)	R1T4362
	CDW = ZKT * ZKW * ZKC * ZKB * TOC53 * (2./D1 +3.33/D2)	R1T4363
	CDWA = CDW/TOC53	R1T4364
	CDOSR=CDW*OSR	R1T4365
	IF (ZM.GT.1.0) CDD= CDW*DIV	R1T4366
	AREW = ARE	R1T4367
	ARE = AREW/TOC13	R1T4368
C		R1T4369
	IF(KPRINT(20).EQ.2) WRITE(6,47)	R1T4370
C		R1T4371
	IF (KPRINT(20).EQ.2) WRITE(6,48) DIV, BETA2, BETAW, BETAW2, CDA, CD1, CD2,	R1T4372
1	TEST, XM, FB, ZML, T1, T2, T3, T4, D1, D2, CDW, CDD	R1T4373
C		R1T4374
	VLE = ZLE * 57.2956	R1T4375
	VTE = ZTE * 57.2956	R1T4376
	IF (KPRINT(20).GT.0) WRITE(6,25) AR, ZKP, ZLAM, ZKT, VLE, ZKC, VTE,	R1T4377
1	ZKLIN, TOC-ARW, TYPE, CAM, TOC, TOC13, XTOC, TOC53, ROOT,	R1T4378

2	FB, HOT, ZML, ZKW, ARE, SOSR, AREW	R1T4379
C	550 IF (KPRINT(20).GT.0) WRITE(6,600)	R1T4380
C	1 IF (KPRINT(20).EQ.1) WRITE(6,610) ZM,BETAW,XM,ZFEXP,CDWA,CDW,	R1T4381
C	1 CD1,CD2,CDDSR	R1T4382
C		R1T4383
C		R1T4384
C		R1T4385
	90 FORMAT(4H0 I,5X,1HX,14X,1HF,14X,2HFP,13X,3HFFP/)	R1T4386
	95 FORMAT(1H ,13,3X,1P4E15.7)	R1T4387
	36 FORMAT(1H0,2X,3HZLE,12X,3HZTE,12X,4HTOC2,11X,5HTOC13,10X,5HTOC53/	R1T4388
	1 1H ,2X,3HARW,12X,4HARW3,11X,4HDEN1,11X,4HDEN2,11X,3HZKT/	R1T4389
	1 3X,3HZKC,12X,5HZLAM1,10X,3HZKP)	R1T4390
	45 FORMAT(1H0,1P5E15.7/(1X,1P5E15.7))	R1T4391
	47 FORMAT(1H0,2X,3HDIV,12X,5HBETA2,10X,5HBETAW,10X,6HBETA2,9X,3HCDA/	R1T4392
	1 1H ,2X,3HCD1,12X,3HCD2,12X,4HTEST,11X,2HXM,13X,2HFB/	R1T4393
	1 1H ,2X,3HZML,12X,2HT1,13X,2HT2,13X,2HT3,13X,2HT4/	R1T4394
	1 1H ,2X,2HD1,13X,2HD2,13X,3HCDW,12X,6HKLINEQ)	R1T4395
	48 FORMAT(1H0,1P5E15.7/(1X,1P5E15.7))	R1T4396
	25 FORMAT(///,16X,36H W I N C P R E S S U R E D R A G//	R1T4397
	111X,17H INPUT PARAMETERS,16X,18H OUTPUT PARAMETERS//	R1T4398
	17X,19H ASPECT RATIO = ,F7.4,10X,12H KPLANE = ,F7.4/	R1T4399
	17X,19H TAPER RATIO = ,F7.4,10X,12H KTHICK = ,F7.4/	R1T4400
	17X,19H LE SWEEP DEG = ,F7.4,10X,12H KCAMB = ,F7.4/	R1T4401
	17X,19H TE SWEEP DEG = ,F8.4,10X,12H KLINTH = ,F7.4/	R1T4402
	17X,19H THICK RATIO = ,F7.4,10X,12H ARWIG = ,F7.4/	R1T4403
	17X,19H AIRFOIL TYPE = ,F3.0,F4.3,F5.3,5X,12H T/C 1/3 = ,F7.4/	R1T4404
	17X,19H XT/C = ,F7.4,10X,12H T/C 5/3 = ,F7.4/	R1T4405
	17X,19H RO/T = ,F7.4,10X,12H F = ,F7.4/	R1T4406
	17X,19H H/T = ,F7.4,10X,12H M* = ,F7.4/	R1T4407
	17X,19H KW = ,F7.4,10X,12H ARE = ,F7.4/	R1T4408
	17X,19H SW/SREF = ,F7.4,10X,12H AREW = ,F7.4/)	R1T4409
	600 FORMAT (//31X,8H RESULTS //,72H MACH BWIG EXP FEXP CDWIGP	R1T4410
	1 CD CDTH2D CDTH3D CD/SREF)	R1T4411
	610 FORMAT(1H , 4(F5.3,2X),5(F7.5,2X))	R1T4412
	RETURN	R1T4413
	END	R1T4414
		R1T4415

CC = 00149

C	SUBROUTINE CDRG(SPEED, AEROK, DELCL, CDC)	R1T4417
C	CAMBER DRAG	R1T4418
C	COMMON /BLKG01/ G1(47), SEXW, G2(31), AR, TR, SPLAN, G3(118)	R1T4419
C	COMMON /BLKA02/ SREF, A1(432)	R1T4420
	CDC = 0.0	R1T4421
	ARSREF = AR * SPLAN/SREF	R1T4422
	E = 1.0/(3.14159*ARSREF*AEROK)	R1T4423
C		R1T4424
C		R1T4425
	COEF1 = 1.0/(3.14159*ARSREF*(1.0-E))	R1T4426
	COEF2 = 0.7 * SEXW/SREF	R1T4427
	IF(E.GE.1.0) GO TO 15	R1T4428
	IF(SPEED.GE.1.0) GO TO 5	R1T4429
	IF(COEF1.GT.COEF2) GO TO 15	R1T4430
	5 CDC = COEF1 * DELCL**2	R1T4431
	GO TO 10	R1T4432
	15 CONTINUE	R1T4433
	CDC = COEF2 * DELCL**2	R1T4434
C		R1T4435
	10 RETURN	R1T4436
	END	R1T4437
		R1T4438
		R1T4439
		R1T4440

CC = 00024

C	SUBROUTINE BDRG(SPEED, AB, SREF, CDB)	R1T4442
C	BASE DRAG	R1T4443
C	COMMON /BLKPRT/ KPRINT(50)	R1T4444
C	CDB = 0.0	R1T4445
	IF(AB.LE.0.0) RETURN	R1T4446
	IF(SPEED.GE.1.0), GO TO 10	R1T4447
	CDB = (0.1 + 0.1222 * SPEED**8) * AB/SREF	R1T4448
	GO TO 20	R1T4449
C	10 CONTINUE	R1T4450
	CDB = (1.42 / (3.15 + SPEED*SPEED)) * AB /SREF	R1T4451
	IF(SPEED.LE.1.8) CDB = 0.2222 * AB/SREF	R1T4452
C	20 CONTINUE	R1T4453
	IF(KPRINT(15).EQ.0) GO TO 30	R1T4454
	WRITE(6,1000) SPEED, AB, SREF, CDB	R1T4455
	1000 FORMAT(10X*MACH =*F10.7,5X*BASE AREA =*F10.7,5X*SREF =*	R1T4456
	1 F12.5,5X*BASE DRAG =*F10.7 /)	R1T4457
C	30 RETURN	R1T4458
	END	R1T4459
		R1T4460
		R1T4461
		R1T4462
		R1T4463
		R1T4464
		R1T4465

CC = 00024

C	SUBROUTINE CMOW(SPEED, CMO)	R1T4467
C	COMPUTES ZERO LIFT PITCHING MOMENT OF WING	R1T4468
C	COMMON /BLKBD4/ XBD4(5), YBD4(5), ZBD4(3), FBD4(5,5,3)	R1T4469
C	COMMON /BLKAO1/ NBODYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	R1T4470
C	COMMON /BLKAO2/ A1(433)	R1T4471
C	COMMON /BLKCO1/ C1(100)	R1T4472
C	COMMON /BLKGO1/ G1(49), SXX(10), G2(42), ESWQC, G3(98)	R1T4473
C	COMMON /BLKPRT/ KPRINT(50)	R1T4474
C	DIMENSION CMC4(20)	R1T4475
C	DATA CMC4 / 4*-0.25, 3*-0.219, -0.3, 0.0, 11*-0.2066 /	R1T4476
C	FMCR = C1(72)	R1T4477
C	CMOS = 0.0	R1T4478
C	NI = A1(235)	R1T4479
C	DO 100 I = 1, NPNLS	R1T4480
C	CMOS = CMOS + A1(244 + I) * SXX(I) * CMC4(NI)	R1T4481
C	100 CONTINUE	R1T4482
C	CMO = 0.0	R1T4483
C	IF(SPEED.GE.1.) RETURN	R1T4484
C	ARXR = G2(35)	R1T4485
C	TOC = G1(46)	R1T4486
C	TWIST = A1(278)	R1T4487
C	SEXW = G1(48)	R1T4488
C	CMOS = CMOS/SEXW	R1T4489
C	CMOB = ARXR * COS(ESWQC)**2/(ARXR + 2.*COS(ESWQC)) * CMOS	R1T4490
C	TRX = G2(29)/G2(27)	R1T4491
C	CALL TLNT(ESWQC,ARXR,TRX, CMOOT, XBD4,YBD4,ZBD4,FBD4, 5,5,3, 5,5)	R1T4492
C	CMOT = CMOOT * TWIST	R1T4493
C	FMACH = SPEED	R1T4494
C	IF(FMACH.GT.FMCR) FMACH = FMCR	R1T4495
C	CMACH = (1. + 5.9*TOC*FMACH**5)/SQRT(1.-(FMACH*COS(ESWQC))**2)	R1T4496
C	CMO = (CMOB + CMOT) * CMACH	R1T4497
C	IF(KPRINT(25).GT.0) WRITE(6,1000) CMO, CMOB, CMOT, CMACH,	R1T4498
C	1 CMOS, ARXR, TOC, TWIST, ESWQC, TRX, CMOOT, FMCR,	R1T4499
C	2 (A1(234+I), A1(244+I), SXX(I), I = 1,NPNLS)	R1T4500
C	1000 FORMAT(5X, *CMOW DUMP*/(1X,7F15.5))	R1T4501
C	RETURN	R1T4502

END

RIT4523

CC = 00057

C	SUBROUTINE WBAC(SPEED, XACR)	R1T4525
C	COMPUTES PITCHING MOMENT SLOPE OF WING	R1T4526
C		R1T4527
C		R1T4528
C	COMMON /BLKA01/ NBDYS, NNACS, NSURFS, NHT, NVT, ISWP, NPNLS	R1T4529
C		R1T4530
C	COMMON /BLKA02/ A02(433)	R1T4531
C		R1T4532
C	COMMON /BLKA03/ ROUGHK, CLE(3), CCR(3), YC(3), A03(10)	R1T4533
C		R1T4534
C	COMMON /BLKG01/ G1(44), DOB, TOC, CLD, G2(32), AR, TR, SPLAN,	R1T4535
1	G3(3), CRX, CBX, CTX, YIX, YOX, SIX, SOX, ARI,	R1T4536
2	ARXR, CBXP, AROP, SOXP, SWPLEI, SWPLEO,	R1T4537
3	SWPMCI, SWPMCO, G4(99)	R1T4538
C		R1T4539
C	COMMON /BLKC01/ C1(100)	R1T4540
C		R1T4541
C	COMMON /BLKPRT/ KPRINT(50)	R1T4542
C		R1T4543
C	DIMENSION XDB(8), YDOB(8), XBDOL(5), YAON(6), FCP(5,6)	R1T4544
C		R1T4545
C	DATA XDB / 0., .05, .1, .15, .2, .3, .4, .5 /,	R1T4546
1	YDOB / 0., .1, .154, .19, .219, .266, .3, .33 /	R1T4547
C		R1T4548
C	DATA XBDOL / 0.4, .7, 1.0, 1.25, 1.67 /,	R1T4549
1	YAON / 0., .4, .8, 1.2, 1.6, 2.0 /,	R1T4550
2	FCP / .54, .535, .525, .516, .5, .42, .435, .45, .46, .46,	R1T4551
3	.35, .377, .4, .414, .425, .295, .33, .355, .375, .394,	R1T4552
4	.246, .285, .32, .345, .365, .21, .25, .288, .315, .34 /	R1T4553
C		R1T4554
C	ARI = ARXR	R1T4555
	TR1 = CTX/CRX	R1T4556
	TW = A02(235)	R1T4557
	FMCRO = C1(72)	R1T4558
C		R1T4559
C	IF(NPNLS.GT.1) ARI = ARI	R1T4560
	IF(NPNLS.GT.1) TR1 = CBX/CRX	R1T4561
C		R1T4562
20	CALL ACCR(SPEED, ARI, SWPLEI, SWPMCI, TR1, SIX, TOC, TW, FMCRO, XACR, CLAI,	R1T4563
1	XACSW)	R1T4564
	XACRX = XACR	R1T4565
	IF(NPNLS.EQ.1) GO TO 100	R1T4566
C		R1T4567
C	CRANKED WING PLANFORMS	R1T4568
C		R1T4569
C	TR2 = CTX/CBXP	R1T4570
C		R1T4571
C	CALL ACCR(SPEED, AROP, SWPLEO, SWPMCO, TR2, SOXP, TOC, TW, FMCRO,	R1T4572
1	XACRP, CLAOP, XACSO)	R1T4573
C		R1T4574
C	XACRO = XACRP * CBXP/CRX - YIX*0.5*TAN(SWPLEO)/CRX	R1T4575
1	+ YIX * TAN(SWPLEI)/CRX	R1T4576
	XACSO = XACSO * CBXP/CRX + XACRO - XACRP*CBXP/CRX	R1T4577
C		R1T4578
C	XACR = (CLAI * SIX * XACRX + CLAOP * SOXP * XACRO) /	R1T4579
		R1T4580

1	(CLAI * SIX + CLAOP * SOXP)	R1T4581
C	XACSW = (CLAI * SIX * XACRX + CLAOP * SOXP * XACSO) /	R1T4582
1	(CLAI * SIX + CLAOP * SOXP)	R1T4583
100	XACMB = XACR	R1T4584
C		R1T4585
C		R1T4586
C	WING-BODY COMBINATION	R1T4587
C		R1T4588
	BLN = A02(85)	R1T4589
	XLE = A02(281)	R1T4590
	TANSQC = 0.5 * (TAN(SWPLEI) + TAN(SWPMCI))	R1T4591
	BO2 = 0.5 * SQRT(AR * SPLAN)	R1T4592
	YWX = A02(303)	R1T4593
	DIA = DOB * BO2 * 2.	R1T4594
	IF(SPEED.GE.1.2) GO TO 200	R1T4595
C	SUBSONIC CALCULATION OF XACN (NOSE) AND XACBW (WING CARRY-OVER)	R1T4596
	XLEQ = BLN + 1.6 * (XLE - BLN)	R1T4597
	XACN = -0.54 * XLEQ / CRX	R1T4598
C		R1T4599
	BARE = ARXR * SQRT(1.-FMCRO**2)	R1T4600
	IF(SPEED.LT.FMCRO) BARE = ARXR * SQRT(1.-SPEED**2)	R1T4601
	CALL LNTP(DOB, FDOB, XDB, YDOB, 8, 4)	R1T4601
	DXQC = G4(18)	R1T4602
	XACBW = 0.25 + DXQC * FDOB / CRX	R1T4603
C		R1T4604
	IF(BARE.GE.4.) GO TO 190	R1T4605
	XACBWO = 0.125 * ARXR * TAN(SWPLEI) * (1.+CTX/CRX)	R1T4606
	IF(XACBWO.GT.0.5) XACBWO = 0.5	R1T4607
C		R1T4607
	XACBW = (XACBWO - XACBW) * (BARE - 4.)**2/16. + XACBW	R1T4608
190	XACBW1 = XACBW	R1T4609
	XACN1 = XACN	R1T4610
C		R1T4610
	IF(SPEED.LE.FMCRO) GO TO 290	R1T4611
C		R1T4612
C	SUPERSONIC CALCULATION OF XACN & XACBW	R1T4613
200	BETA = 0.663325	R1T4613
	IF(SPEED.GT.1.2) BETA = SQRT(SPEED**2 - 1.)	R1T4614
C		R1T4615
	AON = (XLE - BLN) / BLN	R1T4616
	IF(AON.LT.0.) AON = 0.	R1T4617
	BDOL = BETA * DIA / BLN	R1T4618
C		R1T4619
C	FIGURE 4.2.2.1-23A DATCOM *****	R1T4620
	XCPOL = DLNT(BDOL, AON, XBDL, YACN, FCP, 5, 6, 5, 2, 2)	R1T4621
	XACN = XLE / CRX * (XCPOL - 1.)	R1T4622
C		R1T4623
	BDOC = BETA * DIA / CRX	R1T4624
	BCOT = BETA * TAN(SWPLEI)	R1T4625
	A1 = 0.5845	R1T4626
	IF(BCOT.GE.1.) A1 = 0.5985 + 0.00214 * ALOG(BCOT)	R1T4627
	IF(BCOT.LT.1.0.AND.BCOT.GT.0.1) A1 = 0.5985 + 0.00607 * ALOG(BCOT)	R1T4628
C		R1T4629
C	FIGURE 4.3.2.1-37A DATCOM *****	R1T4630
	XACBW = 0.5 + A1 * BDOC - 0.1057 * BDOC**2 + 0.0172 * BDOC**3	

```

      IF( SPEED.GE.1.2 ) GO TO 290
C
      XACN  = XACN1 + (XACN -XACN1) *(SPEED-FMCRO)/(1.2-FMCRO)
      XACBW = XACBW1+ (XACBW-XACBW1)*(SPEED-FMCRO)/(1.2-FMCRO)
C
290  CLAB  = C1(55)
      CLAW  = C1(54)
      FKBW  = 0.0004 +1.2662*DOB +.6018*DOB**2 +.1263*DOB**3
      FKWB  = 1.0028 +.7116*DOB +.42*DOB**2 -.1366*DOB**3
      CLABW = FKBW/(FKBW + FKWB) * CLAW
      CLAWB = FKWB/(FKBW + FKWB) * CLAW
C
300  XACR  = (XACN * CLAB + XACWB * CLAWB + XACBW * CLABW)/
1      (CLAB + CLAW)
C
      XACS  = (XACN * CLAB + XACSW * CLAWB + XACBW * CLABW)/
1      (CLAB + CLAW)
      C1(08) = XACS
C
      IF( KPRINT(18).EQ.0 ) GO TO 400
      WRITE(6,1000) XACR, XACN,CLAB, XACWB,CLAWB, XACBW,CLABW,
1      XACS, SPEED, XLEQ, FDOB
C
400  RETURN
1000 FORMAT(5X,*WBAC DUMP* /(5X,6F15.5) )
      END

```

```

R1T46301
R1T46302
R1T46303
R1T46304
R1T4631
R1T4632
R1T4633
R1T4634
R1T4635
R1T4636
R1T4637
R1T4638
R1T4639
R1T4640
R1T4641
R1T4642
R1T4643
R1T4644
R1T4645
R1T4646
R1T4647
R1T4648
R1T4649
R1T4650
R1T4651
R1T4652

```

CC = 00138

	SUBROUTINE ACCR(SPEED,AR,SWPLE,SWPMC,TR,SPLAN,TOC,TW,FMCRC,	R1T4654
	1 XACR, CLAX, XACS)	R1T4655
C		R1T4656
C	COMPUTES AERODYNAMIC CENTER OF SINGLE PANEL WINGS	R1T4657
C		R1T4658
	COMMON /BLKBD3/ BOT(12), ATSW(6), TPRT(6), FXAC(12,6,6)	R1T4659
C		R1T4660
	COMMON /BLKBD6/ XBD6(4), YBD6(4), ZBD6(6), FBD61(4,4,6),	R1T4661
1	FBD62(4,4,6), FBD63(4,4,6)	R1T4662
C		R1T4663
	COMMON /BLKCLA/ X(11)	R1T4664
	COMMON /BLKPRT/ KPRINT(50)	R1T4665
	COMMON /BLKCO1/ C1(100)	R1T4666
	COMMON /BLKMAP/ MAP, TRANS, DY, AMAP(22), BMAP(11)	R1T4667
C		R1T4668
	DIMENSION XDY(6), YAC(6), XVAL(3), VAL(3)	R1T4669
	DATA XVAL / 0.0, 0.2, 0.5 /	R1T4670
	DATA XDY / .2, .4, .6, .8, 1.2, 1.6 /,	R1T4671
1	YAC / .67,.585,.55,.555,.59,.61 /	R1T4672
C		R1T4673
	NI = TW	R1T4674
	DY = AMAP(NI) * TOC	R1T4675
	ARLO = C1(71)	R1T4676
C		R1T4677
	SB = 2.	R1T4678
	IF(SPEED.EQ.1.0) GO TO 20	R1T4679
	IF(SPEED.GT.1.0) GO TO 10	R1T4680
C		R1T4681
	TANOB = TAN(SWPLE)/SQRT(1.-SPEED**2)	R1T4682
	IF(TANOB.LE.1.0) SB = TANOB	R1T4683
	IF(TANOB.GT.1.0) SB = 2. - 1./TANOB	R1T4684
	GO TO 20	R1T4685
C		R1T4686
10	TANOB = TAN(SWPLE)/SQRT(SPEED**2 -1.)	R1T4687
	IF(TANOB.LE.1.) SB = 4.-TANOB	R1T4688
	IF(TANOB.GT.1.) SB = 2.+1./TANOB	R1T4689
C		R1T4690
20	ARTSW = AR * TAN(SWPLE)	R1T4691
C		R1T4692
	CALL TLNT(SB,ARTSW,TR, XAC1, BOT,ATSW,TPRT,FXAC, 12,6,6, 12,6)	R1T4693
C		R1T4694
	KPR11 = KPRINT(11)	R1T4695
	KPRINT(11) = 0	R1T4696
	X(1) = SPLAN	R1T4697
	X(2) = 0.0	R1T4698
	X(3) = TR	R1T4699
	X(4) = AR	R1T4700
	X(5) = 0.0	R1T4701
	IF(TW.EQ.8.) X(5) = 0.0334	R1T4702
	X(6) = 0.0	R1T4703
	X(7) = 0.0	R1T4704
	IF(TW.EQ.8.) X(7) = 0.09	R1T4705
	X(8) = 0.0	R1T4706
	X(9) = 0.0	R1T4707
	IF(TW.EQ.8.) X(9) = 1.173763 * TOC	R1T4708
	X(10) = SWPMC	R1T4709

30	X(11) = SPLAN	R1T4710
C		R1T4711
	CALL AER2(SPEED, CLAX)	R1T4712
C		R1T4713
	XAC2 = XAC1	R1T4714
	VBAR = (SPEED**2 - 1.)/TOC**0.6667	R1T4715
	ABAR = AR * TOC**0.3333	R1T4716
C		R1T4717
	IF(VBAR.GE.1.0.OR.VBAR.LE.-2.) GO TO 40	R1T4718
C		R1T4719
	TRANSONIC AERODYNAMIC CENTER	R1T4720
C		R1T4721
	CALL TLNT(ABAR,VBAR,ARTSW,VAL(1),XBD6,YBD6,ZBD6,FBD61,4,4,6,4,4)	R1T4722
	CALL TLNT(ABAR,VBAR,ARTSW,VAL(2),XBD6,YBD6,ZBD6,FBD62,4,4,6,4,4)	R1T4723
	CALL TLNT(ABAR,VBAR,ARTSW,VAL(3),XBD6,YBD6,ZBD6,FBD63,4,4,6,4,4)	R1T4724
C		R1T4725
	CALL LNTP(TR, XAC2, XVAL, VAL, 3, 2)	R1T4726
C		R1T4727
40	XACR = XAC1	R1T4728
	IF(SPEED.LE.FMCRO.OR.SPEED.GE.1.2) GO TO 50	R1T4729
	FM1 = FMCRO + 0.05	R1T4730
	FM2 = SQRT(1. + TOC**-.66667)	R1T4731
	XACR = XAC2	R1T4732
C		R1T4733
	IF(SPEED.GT.FM1.AND.SPEED.LT.FM2) GO TO 50	R1T4734
	IF(SPEED.LE.FM1) XACR = XAC1 + (XAC2-XAC1)*(SPEED-FMCRO)/.05	R1T4735
	IF(SPEED.GE.FM2) XACR = XAC2 + (XAC1-XAC2)*(SPEED-FM2)/	R1T4736
	1 (1.2 - FM2)	R1T4737
C		R1T4738
50	ARHI = 2.* ARLO	R1T4739
	XACS = (1.+2.*TR)/12.* ARTSW + (1. + TR**2/(1.+TR))/3.	R1T4740
	IF(AR.GT.ARHI) GO TO 60	R1T4741
	CALL LNTP(DY, XACS1, XDY, YAC, 6, 2)	R1T4742
	XACS = XACS1 + (1.+2.2*TR)/17.544 * ARTSW - 0.2	R1T4743
C		R1T4744
60	IF(KPRINT(18).EQ.2) WRITE(6,1000) SPEED, AR, SWPLE, SWPMC,	R1T4745
1	TR, SPLAN, TOC, TW, FMCRO, XACR, CLAX,	R1T4746
2	SB, ARTSW, XAC1, ABAR, VBAR, VAL, XAC2,	R1T4747
3	FM1, FM2, ARHI, XACS	R1T4748
C		R1T4749
1000	FORMAT(5X, *ACCR DUMP* /15X,6F15.5))	R1T4750
C		R1T4751
	KPRINT(11) = KPR11	R1T4752
	RETURN	R1T4753
	END	R1T4754

CC = 00101

	OVERLAY(3,4)	R1T4756
	PROGRAM LSHL	R1T4757
C	COMMON /BLKHLS/ NHLSV, DFI(3,5), CPF1(5), DSI(5), CPS1(5),	R1T4758
	1 DELCD(5), H(5), DFI2(3,5), CPF2(5), DSI2(5),	R1T4759
	2 CPS2(5)	R1T4760
	COMMON /BLKA04/ IHLS,NXF,NXS, A2(62), BF11, BF10, CF1(3), BF21,	R1T4761
	1 BF20, CF2(3), BS11, BS10, CS1, BS21, BS20, CS2	R1T4762
	COMMON /BLKCO1/ C1(85), CPF, CPS, XACS, DCDLG, C2(11)	R1T4763
	COMMON /BLKG01/ G1(200)	R1T4764
	COMMON /BLKCO2/ CO2(11)	R1T4765
C		R1T4766
	COMMON /BLKOV3/ ITYP, OV3A(7), JPASS, RCT, OTE, ITRIM	R1T4767
C		R1T4768
	DIMENSION DF(3), CFOC(3), COL(10)	R1T4769
	DATA DF / 3*0.0 /, CFOC / 3*0.0 /	R1T4770
	DATA COL / 10*0.0 /	R1T4771
C		R1T4772
	DO 700 I = 1, NHLSV	R1T4773
	DF(1) = DFI(1,I)	R1T4774
	DF(2) = DFI(2,I)	R1T4775
	DF(3) = DFI(3,I)	R1T4776
	DS = DSI(1)	R1T4777
	CPF = CPF1(1)	R1T4778
	CPS = CPS1(1)	R1T4779
	IF(CPF.EQ.0.) CPF = 1.	R1T4780
	CFOC(1)= CF1(1) / CPF	R1T4781
	CFOC(2)= CF1(2) / CPF	R1T4782
	CFOC(3)= CF1(3) / CPF	R1T4783
	IF(CPS.EQ.0.) CPS = 1.	R1T4784
	CSOC = CS1 / CPS	R1T4785
	DCDLG = DELCD(1)	R1T4786
	HGT = H(1)	R1T4787
	DFLAP = DF(1) + DF(2) + DF(3)	R1T4788
C		R1T4789
	BF1 = BF11	R1T4790
	BF0 = BF10	R1T4791
	CF = CF1(1) + CF1(2) + CF1(3)	R1T4792
	BS1 = BS11	R1T4793
	BS0 = BS10	R1T4794
C		R1T4795
	CALL SSET(DF, DS, CFOC, CSOC, OTE, RCT)	R1T4796
	CALL DSET(BF1, BF0, BS1, BS0, CF)	R1T4797
	CALL MSET(BF1, BF0, BS1, BS0)	R1T4798
	IF(IHLS.EQ.1) GO TO 500	R1T4799
C		R1T4800
C	CALCULATION OF THE EFFECT OF THE SECOND HIGH LIFT SEGMENT	R1T4801
C		R1T4802
	DF(1) = DFI2(1,I)	R1T4803
	DF(2) = DFI2(2,I)	R1T4804
	DF(3) = DFI2(3,I)	R1T4805
	DS = DSI2(1)	R1T4806
	CPF = CPF2(1)	R1T4807
	CPS = CPS2(1)	R1T4808
	CFOC(1)= CF2(1) * CPF	R1T4809
	CFOC(2)= CF2(2) * CPF	R1T4810
		R1T4811

	CFNC(3) = CF2(3) * CPF	R1T4812
	CSOC = CS2 * CPS	R1T4813
C		R1T4814
	DA = CO2(8) - G1(82)	R1T4815
	DR = CO2(11) - 1.	R1T4816
	DO 300 J = 1, 10	R1T4817
300	COL(J) = CO2(J)	R1T4818
C		R1T4819
	BFI = BF2I	R1T4820
	BFO = BF2O	R1T4821
	CF = CF2(1) + CF2(2) + CF2(3)	R1T4822
	BSI = BS2I	R1T4823
	BSO = BS2O	R1T4824
C		R1T4825
	CALL SSET(DF, DS, CFC, CSOC, OTE, ROT)	R1T4826
	CALL DSET(BFI, BFO, BSI, BSO, CF)	R1T4827
	CALL MSET(BFI, BFO, BSI, BSO)	R1T4828
C		R1T4829
	CO2(8) = CO2(8) + DA	R1T4830
	CO2(11) = CO2(11) + DR	R1T4831
	DO 400 J = 1, 10	R1T4832
400	IF(J.NE.8) CO2(J) = CO2(J) + COL(J)	R1T4833
C		R1T4834
	500 ALPHA = -1.	R1T4835
	CL = - 10.	R1T4836
600	ALPHA = ALPHA + 1.	R1T4837
	CL1 = CL	R1T4838
	CALL DAFO(ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG, HGT, DFLAP)	R1T4839
	IF(ALPHA.EQ.0.0) WRITE(6,2000)	R1T4840
	IF(HGT.EQ.0.0) WRITE(6,2001) ALPHA, CL, CD, CM	R1T4841
	IF(HGT.GT.0.0) WRITE(6,2001) ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG	R1T4842
C		R1T4843
	IF(ALPHA.GT.60.0) GO TO 700	R1T4844
	IF(CL.GT.CL1) GO TO 600	R1T4845
	700 CONTINUE	R1T4846
2000	FORMAT(1H1, 20X, *HIGH LIFT SURVEY* // 15X, *ALPHA* 10X,	R1T4847
1	* CL * 10X, * CD * 10X, * CM * 10X, *AGRD* 10X, * CLG* 10X, * CDG*	R1T4848
2	10X, *CMG * /)	R1T4849
2001	FORMAT(5X, F15.2, 3F15.5, F15.3, 3F15.5)	R1T4850
	END	R1T4851

CC = 00096

	SUBROUTINE SSET(DF, DS, CFDC, CSOC, DTE, ROT)	R1T4853
C		R1T4854
C	COMPUTES SECTION INCREMENTS FOR FLAPS AND SLATS	R1T4855
C		R1T4856
	COMMON /BLKPRT/ KPRINT(50)	R1T4857
C		R1T4858
	COMMON /BLKCO1/ C1(77), DC1F, DC1MF, DCDFS, DCMFS, DC1S,	R1T4859
1	DC1MS, DCDSS, DCMSS, CPF, CPS, C2(13)	R1T4860
C		R1T4861
	COMMON /BLKA04/ IHLS, NXF, NXS, TF, TS, XDF(5), FCPF(5),	R1T4862
1	FDC1F(5), FDCMXF(5), FCDF(5), FCMF(5),	R1T4863
2	XDS(5), FCPS(5), FDC1S(5), FDCMS(5), FDCDS(5),	R1T4864
3	FOMS(5), A04(16)	R1T4865
C		R1T4866
	DIMENSION DF(3), CFDC(3), ETA(3), C10(3), DCL(3), RMAX(3),	R1T4867
1	DCLM(3), PCM(3), DCM(3), X1(4), Y1(8), F1(4,8),	R1T4868
2	X2(5), Y2(6), F2(5,6), X3(6), Y3(4), F3(6,4),	R1T4869
3	A0(4), A1(4), XROT(5), YEMAX(5,3), XXDS(4), YED(4,3)	R1T4870
C		R1T4871
	DATA A0 / 0.915, 1.215, 1.39, 1.545 /,	R1T4872
1	A1 / -.01186, -.0115, -.0125, -.0130 /,	R1T4873
2	X1 / 0., .1, .2, .3 /,	R1T4874
3	Y1 / 0., 10., 20., 30., 40., 50., 60., 70. /,	R1T4875
4	F1 / 5*0.0,.001,.004,.0085, 0.,.009,.02,.032,	R1T4876
5	0.,.019,.0425,.07, 0.,.0325,.071,.118,	R1T4877
5	0.,.049,.104,.17, 0.,.063,.14,.237,	R1T4878
5	0.,.084,.18,.295 /,	R1T4879
6	X2 / 0., .1, .2, .3, .4 /,	R1T4880
7	Y2 / 0., 10., 20., 30., 40., 50. /,	R1T4881
8	F2 / 6*0.0,.001,.003,.006,.012, 0.,.003,.0075,.013,.023,	R1T4882
9	0.,.011,.024,.0385,.059, 0.,.02,.045,.079,.117,	R1T4883
9	0.,.03,.069,.127,.205 /,	R1T4884
1	X3 / 0., 1., 2., 4., 6., 8. /,	R1T4885
2	Y3 / .2, .3, .4, .656 /,	R1T4886
3	F3 / 0.,.01,.02,.058,.155,.28,0.,.01,.02,.05,.12,.24,	R1T4887
4	0.,.01,.02,.046,.091,.2,0.,.008,.016,.032,.057,.106/	R1T4888
	DATA XROT / 0.0, 0.07, 0.08, 0.1, 0.2 /,	R1T4889
1	YEMAX / 0.56, 0.98, 1.02, 0.95, 0.0,	R1T4890
2	.56,1.34,1.74,1.72,.8,.56,1.09,1.2,1.15,.2 /,	R1T4891
3	XXDS / 0.0, 15.0, 25.0, 45.0 /,	R1T4892
4	YED / 1.0, 1.0, .96, .25, 1.0,1.0,.75,0.,	R1T4893
5	1.0, 1.0, .97, .67 /	R1T4894
C		R1T4895
	DC1F = 0.0	R1T4896
	DC1MF = 0.0	R1T4897
	DCDFS = 0.0	R1T4898
	DCMFS = 0.0	R1T4899
	DC1S = 0.0	R1T4900
	DC1MS = 0.0	R1T4901
	DCDSS = 0.0	R1T4902
	DCMSS = 0.0	R1T4903
C		R1T4904
	IF(TF.EQ.0.) GO TO 100	R1T4905
	IF(TF.GE.5.) GO TO 60	R1T4906
C		R1T4907
C	TRAILING-ENGINE DEVICE *****	R1T4908

	NF = TF	R1T4909
	CFP = CFOC(1) + CFOC(2) + CFOC(3)	R1T4910
	IF (TF.EQ.1.) ZKM = 1.0031 -.928*CFP + 9.9869*CFP**2	R1T4911
1	-7.1843*CFP**3	R1T4912
	IF(TF.EQ.1.0.AND.CFP.LT.0.1) ZKM = 1.0	R1T4913
	IF(TF.GT.1.) ZKM = 1.2029 -1.1834*CFP +13.145*CFP**2	R1T4914
1	-11.5455*CFP**3	R1T4915
	IF(TF.GT.1.0.AND.CFP.LT.0.1) ZKM = 1.2	R1T4916
	ZKD = 1.9 - 0.0225 * (DF(1)+DF(2)+DF(3))	R1T4917
	IF(ZKD.LT.1.0) ZKD = 1.0	R1T4918
	ZKT = 1.23 + 4.5 * ROT	R1T4919
	ZKMAX = ZKT * ZKD	R1T4920
C	IF(TF.LE.2.) NT = 1	R1T4921
	IF(TF.EQ.3.) NT = 2	R1T4922
	IF(TF.EQ.4.) NT = 3	R1T4923
C		R1T4924
	PHI = DF(1)	R1T4925
C		R1T4926
	DO 50 I = 1, NT	R1T4927
	IF(I.GT.1) PHI = DF(I) + PHI	R1T4928
	IF(I.EQ.NT) PHI = PHI + OTE	R1T4929
	IF(I.GT.1) CFP = CFP - CFOC(I-1)	R1T4930
	ETA(I) = AO(NF) + A1(NF)*PHI	R1T4931
	IF(TF.EQ.1.0.AND.PHI.GT.35.0) ETA(I) = 1.5912 -.05068*PHI	R1T4932
1	+ .000754*PHI**2 -.0000039*PHI**3	R1T4933
	IF(ETA(I).GT.0.77) ETA(I) = 0.77	R1T4934
C		R1T4935
	DFR = DF(1)/57.2556	R1T4936
	OF = ACOS(1. - 2.*CFP)	R1T4937
	C10(I) = 2.*(OF + SIN(OF))	R1T4938
C		R1T4939
	DCL(I) = ETA(I) * C10(I) * DFR	R1T4940
	DC1F = DC1F + DCL(I)	R1T4941
C		R1T4942
	XS = 0.0	R1T4943
	IF(TS.GT.0.0.AND.DS.GT.0.0) XS = CSOC	R1T4944
	X = ACOS(2.*XS - 1.)	R1T4945
	RMAX(I) = 1. -(OF/(OF + SIN(OF))) * (1.+ ALOG(ABS(SIN(.5*(X+OF)))/	R1T4946
1	SIN(.5*(X-OF))))/(OF * TAN(.5*X)))	R1T4947
C		R1T4948
	DCLM(I) = DCL(I) * RMAX(I) * ZKMAX	R1T4949
	IF(DCLM(I).GT.DCL(I)) DCLM(I) = DCL(I)	R1T4950
	DC1MF = DC1MF + DCLM(I)	R1T4951
C		R1T4952
	RCM(I) = - CFP * SIN(OF)/C10(I)	R1T4953
C		R1T4954
	DCM(I) = DCL(I) * RCM(I) * ZKM	R1T4955
	DCMFS = DCMFS + DCM(I)	R1T4956
C		R1T4957
	50 CONTINUE	R1T4958
C		R1T4959
	DCL2 = DC1F*DC1F	R1T4960
	CFP = CFOC(1) + CFOC(2) + CFOC(3)	R1T4961
	IF(TF.EQ.1.) DCDFS = DLNT(CFP, DF(1), X1, Y1, F1, 4,8,4,4,2)	R1T4962
	IF(TF.EQ.2.) DCDFS = DLNT(CFP, DF(1), X2, Y2, F2, 5,6,5,4,2)	R1T4963
		R1T4964

	IF(TF.GF.3.) DCDFS = DLNT(DCL2, CFP, X3, Y3, F3, 6.4, 6.4, 2)	R1T4965
C	GO TO 100	R1T4966
60	CALL LNTP(DF(1), DC1F, XDF, FDC1F, NXF, 4)	R1T4967
	CALL LNTP(DF(1), DC1MF, XDF, FDCMXF, NXF, 4)	R1T4968
	CALL LNTP(DF(1), DCDFS, XDF, FCDF, NXF, 4)	R1T4969
	CALL LNTP(DF(1), DCMFS, XDF, FCMF, NXF, 4)	R1T4970
	CALL LNTP(DF(1), CPF, XDF, FCPF, NXF, 4)	R1T4971
		R1T4972
C		R1T4973
100	IF(TS.LE.0.) GO TO 200	R1T4974
	IF(TS.GE.4.) GO TO 150	R1T4975
C	LEADINE-EDGE DEVICE	R1T4976
C	****	R1T4977
	OS = ACOS(1. - 2.*CSOC)	R1T4978
	C1LE = 2.*(SIN(OS) - OS)	R1T4979
	CMXLE = 2.* SIN(OS)	R1T4980
	CMLE = (1.-CSOC)*SIN(OS) C1LE	R1T4981
	DSR = DS/57.2956	R1T4982
	NS = TS	R1T4983
	CALL LNTP(ROT, EMAX, XROT, YEMAX(1,NS), 5,2)	R1T4984
	CALL LNTP(DS, ED, XXDS, YED(1,NS), 4, 2)	R1T4985
C		R1T4986
	DC1S = C1LE * DSR	R1T4987
	DC1MS = CMXLE * EMAX * ED * DSR	R1T4988
	DCDSS = 0.0	R1T4989
	DCMSS = CMLE * DC1S	R1T4990
C		R1T4991
	GO TO 200	R1T4992
150	CALL LNTP(DS, DC1S, XDS, FDC1S, NXS, 4)	R1T4993
	CALL LNTP(DS, DC1MS, XDS, FDCMS, NXS, 4)	R1T4994
	CALL LNTP(DS, DCDSS, XDS, FDCDS, NXS, 4)	R1T4995
	CALL LNTP(DS, DCMSS, XDS, FDCMS, NXS, 4)	R1T4996
	CALL LNTP(DS, CPS, XDS, FCPS, NXS, 4)	R1T4997
C		R1T4998
200	IF(KPRINT(27).EQ.1) HRITE(6,1000) DC1F, DC1MF, DCDFS, DCMFS,	R1T4999
1	DC1S, DC1MS, DCDSS, DCMSS, CPF, CPS, DCL, ETA, C10,	R1T5000
2	DCLM, RMAX, ZKMAX, DCM, RCM, ZKM, C1LE, CMXLE, EMAX,	R1T5001
3	ED, TF, TS, DF, DS, CFCF, CSOC, OTE, ROT	R1T5002
C		R1T5003
1000	FORMAT(5X, *SSET DUMP* /(5X, 7F15.5))	R1T5004
C		R1T5005
	RETURN	R1T5006
	END	R1T5007

CC = 00155

	SUBROUTINE DSET(BF, BFO, BSI, BSO, CF)	RIT5009
C		RIT5010
C	AERODYNAMIC CONTROL ROUTINE FOR HIGH LIFT SYSTEMS	RIT5011
C		RIT5012
	COMMON /BLKCO2/ DCLOF, DCLOS, DCLMF, DCLMS, DCDMIN, DCLF, DCLS,	RIT5013
1	SPLANX, DCMO, DCMCL, RCLA	RIT5014
	COMMON /BLKCO1/ C1(11), AK, DCL, C2(64), DC1F, DC1MF, DCDFS,	RIT5015
1	DCMFS, DC1OS, DC1MS, DCDSS, DCMSS, CPF, CPS, C3(13)	RIT5016
	COMMON /BLKGO1/ G1(200)	RIT5017
C		RIT5018
	COMMON /BLKMX3/ SPAN(6), YTPR(4), FKB(6,4), FKS(6), FKD(6,4)	RIT5019
	COMMON /BLKMX9/ SPAN4(5), YBFI1(5), FKA(5,5),	RIT5020
1	SPAN5(5), YBFI2(3), FKF(5,3)	RIT5021
C		RIT5022
	ZKT = 1.0	RIT5023
	SWPLE = G1(103)	RIT5024
	SWPMC = G1(49)	RIT5025
	TPR = G1(81)	RIT5026
	AR = G1(80)	RIT5027
	SPLAN = G1(82)	RIT5028
	DCLS = 0.0	RIT5029
C		RIT5030
	ZKBI = DLNT(BFI, TPR, SPAN, YTPR, FKB, 6,4, 6,2,2)	RIT5031
	ZKBO = DLNT(BFO, TPR, SPAN, YTPR, FKB, 6,4, 6,2,2)	RIT5032
	ZKB = ZKBO - ZKBI	RIT5033
C		RIT5034
	ASC = SQRT(1. - (1.-CF)**1.61)	RIT5035
	ZKC = 1. + 1.875 * (1.-ASC)**2 /AR	RIT5036
C		RIT5037
	CALL LNTP(BSI, ZKSI, SPAN, FKS, 6, 4)	RIT5038
	CALL LNTP(BSO, ZKSO, SPAN, FKS, 6, 4)	RIT5039
	ZKS = ZKSO - ZKSI	RIT5040
C		RIT5041
C		RIT5042
	SPLANX = SPAN * (1. + (CPF-1.)/(1.+TPR) * (2.-(1.-TPR)	RIT5043
1	* (BFO + BFI)) * (BFO - BFI)	RIT5044
2	+ (CPS-1.)/(1.+TPR) * (2.-(1.-TPR)*(BSO-BSI))	RIT5045
3	* (BSO - BSI))	RIT5046
C		RIT5047
20	CLOC1 = AR/(2. + SQRT(4. + (SPLAN * AR/SPLANX /COS(SWPMC))**2))	RIT5048
	CLOC1R = AR/(2. + SQRT(4. + (AR/COS(SWPMC))**2))	RIT5049
	RCLA = CLOC1/CLOC1R	RIT5050
C		RIT5051
	DCLOF = ZKT * DC1F * CLOC1 * ZKC * ZKB	RIT5052
	DCLMF = ZKT * DC1MF * CLOC1 * ZKC * ZKB * COS(SWPMC)	RIT5053
C		RIT5054
	DCLOS = DC1OS * CLOC1 * ZKS	RIT5055
	DCLMS = DC1MS * CLOC1 * ZKS * COS(SWPLE)	RIT5056
C		RIT5057
C		RIT5058
	30 WRITE(6,1000) ZKB, ZKC, ZKS, CLOC1, DC1F, DC1MF, DC1OS,	RIT5059
1	DC1MS, DCLOS, DCLOF	RIT5060
C		RIT5061
C	CALCULATION OF DRAG PARAMETERS	RIT5062
C		RIT5063
	SWPHL = ATAN(TAN(SWPLE) - 4./AR * (1.-TPR)/(1.+TPR) * (1.-CF))	RIT5064

C	ZKDI = DLNT(BFI, TPR, SPAN, YTPR, FKD, 6, 4, 6, 2, 2)	RIT5065
	ZKDO = DLNT(BFO, TPR, SPAN, YTPR, FKD, 6, 4, 6, 2, 2)	RIT5066
	ZKD = ZKDO - ZKDI	RIT5067
C		RIT5068
	40 CONTINUE	RIT5069
	E = AR/6.283	RIT5070
	D = BFO	RIT5071
C		RIT5072
	ZKA = DLNT(E, BFI, SPAN4, YBFI1, FKA, 5, 5, 5, 4, 2)	RIT5073
	ZKF = DLNT(D, BFI, SPAN5, YBFI2, FKF, 5, 3, 5, 2, 2)	RIT5074
C		RIT5075
C		RIT5076
C		RIT5077
	45 WRITE(6, 1001) SWPHL, ZKD, ZKA, ZKF, CDO, SWPLE, AR, TPR, AK, DCL	RIT5078
	EK = 0.31831/AR/AK	RIT5079
C		RIT5080
	50 CONTINUE	RIT5081
	DCLF = (1.-EK) * DCLOF / (1.+1.16*CLOC1*(.5-CF))	RIT5082
	CDC = 0.0	RIT5083
	IF(EK.LT.1.0) CDC = (DCLF**2 + 2.*DCLF*DCL)*0.31831/	RIT5084
	1 (AR * (1.-EK))	RIT5085
C		RIT5086
	DCDF = DCDFS * CGS(SWPHL) * ZKD	RIT5087
	DCDS = DCDS * CGS(SWPLE) * ZKD	RIT5088
	CDI = ZKA * ZKF * DCLOF**2 * 0.31831/AR	RIT5089
C		RIT5090
	DCDMIN = DCDF + DCDS + CDI + CDC	RIT5091
	WRITE(6, 1002) DCDMIN, CD%, DCDS, DCDF, DCL, DCLF, CDC	RIT5092
C		RIT5093
	60 RETURN	RIT5094
	1000 FORMAT(1H1, 10X, *HIGH LIFT CONSTANTS*, // 10X	RIT5095
	1 *ZKB ** F10.6, 10X *ZKC ** F10.6, 10X *ZKS **	RIT5096
	2 F10.6 / 10X *CLOC1** F10.6, 10X *DCLF ** F10.6,	RIT5097
	3 10X *DC1MF** F10.6 / 10X *DC1OS** F10.6, 10X	RIT5098
	4 *DC1MS ** F10.6, 10X	RIT5099
	6 *DCLOS** F10.6 / 10X *DCLOF** F10.6)	RIT5100
	1001 FORMAT(/// 10X *SWPHL** F10.6, 10X *ZKD ** F10.6,	RIT5101
	1 10X *ZKA ** F10.6 / 10X *ZKF ** F10.6, 10X	RIT5102
	2 *CDO ** F10.6, 10X *SWPLE** F10.6 / 10X	RIT5103
	3 *AR ** F10.6, 10X *TPR ** F10.6, 10X	RIT5104
	4 *AK ** F10.6 / 10X *DCL ** F10.6)	RIT5105
	1002 FORMAT(/// 10X, *DCDMIN ** F10.6, 10X *CDI ** F10.6, 10X	RIT5106
	1 *DCDS ** F10.6 / 10X *DCDF ** F10.6, 10X	RIT5107
	2 *DCL ** F10.6, 10X *DCLF ** F10.6 / 10X *CDC ** F10.6)	RIT5108
	1003 FORMAT (6F10.6)	RIT5109
	END	RIT5110
		RIT5111

CC = 00103

C	SUBROUTINE MSET(BFI, BFO, BSI, BSO)	RIT5113
C	COMPUTES MOMENT SHIFT DUE TO FLAPS AND SLATS AT CL = 0	RIT5114
C	COMMON /BLKC02/ DCLOF, DCLOS, DCLMF, DCLMS, DCOMIN, DCLF, DCLS,	RIT5115
	1 SPLANX, DCMO, DCMCL, RCLA	RIT5116
	COMMON /BLKC01/ C1(77), DC1F, DC1MF, DCDFS, DCMFS,	RIT5117
	1 DC1S, DC1MS, DC0SS, DCMSS, CPF, CPS, C2(13)	RIT5118
	COMMON /BLKG01/ G1(200)	RIT5119
C	COMMON/BLKMX7/ SPAN2(11), YTPR2(5), FKSW(11,5)	RIT5120
	COMMON/BLKMX8/ SPAN3(6), YTPR3(4), FKM(6,4)	RIT5121
C	TPR = G1(81)	RIT5122
	AR = G1(80)	RIT5123
	SWPMC = G1(49)	RIT5124
	CMCL = C1(15)	RIT5125
	CLA = C1(17)	RIT5126
	ALO = C1(18)	RIT5127
C	FLAP CALCULATIONS START HERE	RIT5128
C	DCMAF = 0.0	RIT5129
	DCMCLF = 0.0	RIT5130
	IF(DCMFS.EQ.0.0.AND.DC1F.EQ.0.0.AND.CPF.EQ.0.0) GO TO 40	RIT5131
C	ZKSWI = DLNT(BFI, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)	RIT5132
	ZKSWO = DLNT(BFO, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)	RIT5133
	ZKSW = ZKSWO - ZKSWI	RIT5134
C	ZKMO = DLNT (BFO, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)	RIT5135
	ZKMI = DLNT (BFI, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)	RIT5136
	ZKM = ZKMO - ZKMI	RIT5137
C	DCMAF = DCMFS * CPF**2 * ZKM + 0.5 * AR * TAN(SWPMC)	RIT5138
	1 * DC1F * ZKSW	RIT5139
C	DCMCLF = (-0.25 * CPF *(CPF-1.) + CMCL * (CPF**2-1.))* ZKM	RIT5140
	40 CONTINUE	RIT5141
C	SLAT CALCULATIONS START HERE	RIT5142
C	DCMAS = 0.0	RIT5143
	DCMCLS = 0.0	RIT5144
	IF(DCMSS.EQ.0.0.AND.DC1S.EQ.0.0.AND.CPS.EQ.0.0) GO TO 50	RIT5145
	ZKMSO = DLNT(BSO, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)	RIT5146
	ZKMSI = DLNT(BSI, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)	RIT5147
	ZKMS = ZKMSO - ZKMSI	RIT5148
C	ZKSHSO = DLNT(BSO, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)	RIT5149
	ZKSHSI = DLNT(BSI, TPR, SPAN2, YTPR2, FKSW, 11, 5, 11, 2, 2)	RIT5150
	ZKSW = ZKSHSO - ZKSHSI	RIT5151
C	DCMAS = DCMSS * CPS**2 * ZKMS + 0.5 * AR * TAN(SWPMC)	RIT5152
	1 * DC1S * ZKSW	RIT5153
C		RIT5154
		RIT5155
		RIT5156
		RIT5157
		RIT5158
		RIT5159
		RIT5160
		RIT5161
		RIT5162
		RIT5163
		RIT5164
		RIT5165
		RIT5166
		RIT5167
		RIT5168

C	DCMCLS = (-0.25 * CPS * (CPS-1.) + CMCL * (CPS**2-1.)) * ZKMS	R1T5169
C		R1T5170
C		R1T5171
C	ADD FLAP AND SLAT MOMENTS	R1T5172
C		R1T5173
50	DCMA = DCMAF + DCMA5	R1T5174
	DCMCL = DCMCLF + DCMCLS	R1T5175
C		R1T5176
	CLO = -CLA * ALO	R1T5177
	DCMO = DCMA - CMCL * (DCLOF + DCLOS) - CMCL * (CLO+DCLOF+DCLOS)	R1T5178
C		R1T5179
C		R1T5180
	WRITE(6,1001) DCMAF, DCMA5, DCMO	R1T5181
C		R1T5182
1001	FORMAT(///10X,*DCMOF = *F10.6/10X,*DCMOS = *F10.6/	R1T5183
1	10X,*DCMO = *F10.6)	R1T5184
	RETURN	R1T5185
	END	R1T5186

CC = 00074

C	SUBROUTINE DAERO(ALPHA, CL, CD, CM, AGRD, CLG, CDG, CMG, H, DF)	R1T5188
C	COMPUTES LIFT MOMENT AND DRAG FOR HIGH-LIFT SYSTEMS IN	R1T5189
C	FREE AIR OR GROUND EFFECT	R1T5190
C		R1T5191
	COMMON /BLKCO2/ DCLOF, DCLOS, DCLMF, DCLMS, DCDMIN, DCLF, DCLS,	R1T5192
1	SPLANX, DCMO, DCMCL, RCLA	R1T5193
	COMMON /BLKG01/ G1(200)	R1T5194
	COMMON /BLKA02/ A1(433)	R1T5195
	COMMON /BLKA03/ A03(12), CMAC, A03A(7)	R1T5196
	COMMON /BLKA04/ A2(66), BF1C, A3(4), BF20, A4(9)	R1T5197
	COMMON /BLKC01/ C1(4), CDMINB, C2(6), FK, DELCL, CMO, CMCL,	R1T5198
1	XACWB, CLA, ALD, C3(19), CDWING(4), C7(12),	R1T5199
1	CLAW, CLAN, CLAT, AH, BH, HSTAR,	R1T5200
2	C4(5), CLMAXB, ABRK, AMAX, DAMAX,	R1T5201
3	C5(25), DEDA, C6(6)	R1T5202
C		R1T5203
	DIMENSION XDX(6), YA(6), YB(6), XTR(6), YAR(4), Z1(6,4),	R1T5204
1	XBF(6), Z2(6)	R1T5205
C		R1T5206
	DATA XDX / -1., -.6, -.2, .2, .6, 1. /,	R1T5207
1	YA / -3.71, -3.03, -2.63, -2.34, -2.13, -2. /,	R1T5208
1	YB / .466, .5, .664, .901, 1.1, 1.171 /	R1T5209
C		R1T5210
C	FIG.S 4.7.1-18A&B IN DATCOM	R1T5211
	DATA XTR / 1., 2., 3., 4., 5., 6. /,	R1T5212
1	YAR / 4., 6., 8., 10. /,	R1T5213
2	Z1 / .825, .772, .745, .726, .715, .712, .855, .772, .738, .715,	R1T5214
3	.7, .695, .68, .775, .727, .702, .69, .685,	R1T5215
4	.895, .776, .723, .697, .68, .675 /,	R1T5216
5	XBF / 0., .2, .4, .6, .8, 1. /,	R1T5217
6	Z2 / 0., .34, .58, .76, .89, 1. /	R1T5218
C		R1T5219
	SPLAN = G1(82)	R1T5220
	AR = G1(80)	R1T5221
	TR = G1(81)	R1T5222
	DCDLG = C5(21)	R1T5223
	ESWQC = G1(102)	R1T5224
	OMEGA = G1(110)	R1T5225
	XH = G1(109)	R1T5226
C		R1T5227
	CLO = -CLA*ALD + DCLOF + DCLOS	R1T5228
	CLMAX = CLMAXB + DCLMF + DCLMS	R1T5229
	CLABL = CLAN + CLAT + CLAW * RCLA	R1T5230
	AK = FK * SPLAN/SPLANX	R1T5231
	DCL = DELCL + DCLF + DCLS	R1T5232
	CDMIN = CDMINB + DCDMIN + DCDLG	R1T5233
C		R1T5234
	ALIN = (CLMAX - CLO)/CLABL - DAMAX	R1T5235
C		R1T5236
	CL = CLABL * ALPHA + CLO	R1T5237
	IF(ALPHA.LE.ALIN) GO TO 20	R1T5238
C		R1T5239
	DELM = CLABL * (ALIN + 2.* DAMAX) + CLO - CLMAX	R1T5240
	DEL = ((ALPHA - ALIN) * 0.5/DAMAX)**2 * DELM	R1T5241
C		R1T5242
		R1T5243

	CL = CL - DEL	R1T5244
C		R1T5245
	20 CD = CDMIN + AK*(CL - DCL)**2	R1T5246
	IF(ALPHA.LE.ALIN) GO TO 30	R1T5247
C		R1T5248
	EO = 0.31831/AR/AK	R1T5249
	E = EO + (0.3 - EO) * 0.5 * (ALPHA - ALIN)/DAMAX	R1T5250
	IF(E.LE.0.0) E = 0.1	R1T5251
	CD = CDMIN + 0.31831/AR/E * (CL - DCL)**2	R1T5252
C		R1T5253
	30 CMOTOT = CMO + DCMO	R1T5254
	DCMDCL = CMCL + DCMCL	R1T5255
	CLTAIL = CLAT * (ALPHA-HSTAR)	R1T5256
	CLWB = CL - CLTAIL	R1T5257
	CM = CMOTOT + DCMDCL * CLWB	R1T5258
	1 -CLTAIL * XH * COS(OMEGA - ALPHA/57.3)/CMAC	R1T5259
C		R1T5260
C	GROUND EFFECTS ARE NOW CALCULATED	R1T5261
C		R1T5262
	DCLWG = 0.0	R1T5263
	DCLTG = 0.0	R1T5264
	DCMWG = 0.0	R1T5265
	DCMTG = 0.0	R1T5266
	DCDG = 0.0	R1T5267
	AGRD = ALPHA	R1T52671
	OMEGA = G1(110)	R1T5268
	BO2 = A1(240)	R1T5269
	DXOB = G1(106)	R1T5270
	CROB = G1(105)	R1T5271
	HTZ = A1(432)	R1T5272
C		R1T5273
	IF(H.EQ.0.0) GO TO 50	R1T5274
	CDWNG = CDWING(1) + CDWING(2) + CDWING(3) + CD - CDMIN	R1T5275
	CALL LNTP(DXOB, A, XDX, YA, 6, 4)	R1T5276
	CALL LNTP(DXOB, B, XDX, YB, 6, 4)	R1T5277
	SIGP = 2.71828 ** (A * (H/BO2)**B)	R1T5278
C		R1T5279
	HOC = H/(CROB*BO2) * 0.5	R1T5280
	A20 = 9.1189 * CL / (COS(ESWQC)**2)	R1T5281
	DLOLO = 0.058/(HOC**1.7) - 0.0085/HOC * A20	R1T5282
	IF(HOC.LT.0.4) DLOLO = DLOLO - 0.02/(HOC**2) + 0.05/HOC	R1T52821
C		R1T5283
	DDCLF = -.142 + .2272*HOC -.0902*HOC**2	R1T5284
C		R1T5285
	SIG = 2.71828 ** (-2.48 * (H/BO2)**.768)	R1T5286
C		R1T5287
	R = SQRT(1. + (H/BO2)**2) - H/BO2	R1T5288
	T = 0.03979 * HOC / (HOC**2 - .015625)	R1T5289
	DCLWG = (9.12/AR + 7.16*CROB) * CLWB * CLAW * SIGP * RCLA	R1T5290
	1 + 0.5 * AR * CROB * DLOLO * R * CLWB	R1T5291
	2 + DDCLF * (DF/50.)**2	R1T5292
C		R1T5293
	DCDG1 = -SIG * 0.31831*CLWB**2/AR	R1T5294
	DCDG = DCDG1 - (CDWNG + DCDG1) * R * T * CLWB	R1T5295
C		R1T5296
C	GROUND EFFECTS ON HORIZONTAL TAIL ARE CALCULATED	R1T5297

C	IF(CLAT.EQ.0.0) GO TO 40	R1T5298
	TRIV = 6.	R1T5299
	IF(TR.NE.0.0) TRIV = 1./TR	R1T5300
C	BPWOB = DLNT(TRIV, AR, XTR, YAR, Z1, 6, 4, 6, 4, 2)	R1T5301
	BF = BF10	R1T5302
	IF(BF20.GT.BF) BF = BF20	R1T5303
C	CALL LNTP(BF, BFOBWP, XBF, Z2, 6, 4)	R1T5304
	SPAN = 802 * 2.	R1T5305
	BWP = BPWOB * SPAN	R1T5306
	BFP = BWP * BFOBWP	R1T5307
C	BEFF = BWP	R1T5308
	IF(BFP.NE.0.0) BEFF = CLO/((CLO -DCLOF)/BWP + DCLOF/BFP)	R1T5309
C	HH = (H + HTZ)*COS(ALPHA/57.3) - XH*SIN(ALPHA/57.3)	R1T5310
	DE = DEDA * ALPHA * (BEFF**2 + (HH-H)**2)/(BEFF**2 + (HH+H)**2)	R1T5311
C	DCLTG = DE * CLAT/(1.-DEDA)	R1T5312
40	AGRD = ALPHA -(DCLWG +DCLTG)/CLAH	R1T5313
C	DCMWG = DCMDCL * DCLWG	R1T5314
	DCMTG = -DCLTG * XH * COS(OMEGA -AGRD/57.2956)	R1T5315
C	50 CLG = CLWB + CLAT *(AGRD-HSTAR) + DCLTG	R1T5316
	CMG = CMOTOT + DCMDCL * CLWB	R1T5317
1	-(CLG-CLWB)* XH *COS(OMEGA -AGRD/57.2956)	R1T5318
C	CDG = CD + DCDG	R1T5319
	RETURN	R1T5320
	END	R1T5321
		R1T5322
		R1T5323
		R1T5324
		R1T5325
		R1T5326
		R1T5327
		R1T5328
		R1T5329

CC = 00145

```

C SUBROUTINE ATMOS (ZFT, TM, SIGMA, RHO, THETA, CA, AMU, K) CS
C THIS IS A SUBROUTINE TO COMPUTE CERTAIN ELEMENTS OF THE 1962 CS
C U.S. STANDARD ATMOSPHERE UP TO 90 KILOMETERS. CS
C ***** CS
C THIS MODEL REPRESENTS MEAN ANNUAL, MID-LATITUDE, DRY AIR CS
C CONDITIONS. DATA ON LATITUDINAL AND SEASONAL VARIATIONS ARE CS
C AVAILABLE FROM THE OFFICE OF THE STAFF METEOROLOGIST (ASD/WE). CS
C CARE MUST BE EXERCISED WHEN USING THIS AND OTHER MODELS FOR CS
C ENGINEERING PURPOSES AND IT IS STRONGLY URGED THAT USERS CONTACT CS
C ASD/WE FOR INTERPRETATION AND ASSISTANCE. CS
C ***** CS
C CALLING SEQUENCE... CS
C CALL ATMOS (ZFT, TM, SIGMA, RHO, THETA, DELTA, CA, AMU, K) CS
C ZFT = GEOMETRIC ALTITUDE (FEET) CS
C TM = MOLECULAR SCALE TEMPERATURE (DEGREES RANKINE) CS
C SIGMA = RATIO OF DENSITY TO THAT AT SEA LEVEL CS
C RHO = DENSITY (LB-SEC**2*FT**(-4) OR SLUGS-FT**3) CS
C THETA = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL CS
C DELTA = RATIO OF PRESSURE TO THAT AT SEA LEVEL CS
C CA = SPEED OF SOUND (FT/SEC) CS
C AMU = VISCOSITY COEFFICIENT (LB-SEC/FT**2) CS
C K = 1 NORMAL CS
C = 2 ALTITUDE LESS THAN -5000 METERS OR GREATER THAN 90 KM CS
C = 3 FLOATING POINT OVERFLOW CS
C ALL DATA AND FUNDAMENTAL CONSTANTS ARE IN THE METRIC SYSTEM AS CS
C THESE QUANTITIES ARE DEFINED AS EXACT IN THIS SYSTEM. CS
C THE RADIUS OF THE EARTH (REFT59) IS THE VALUE ASSOCIATED WITH THE CS
C 1959 ARDC ATMOSPHERE SO THAT PROGRAMS CURRENTLY USING THE LIBRARY CS
C ROUTINE WILL NOT REQUIRE ALTERATION TO USE THIS ROUTINE. CS
C DIMENSION HB(10),TMB(10),DELTAB(10),ALM(10),ARAY(10,4) CS
C EQUIVALENCE(ARAY(1,1),HB),(ARAY(1,2),TMB),(ARAY(1,3),DELTAB), CS
C (ARAY(1,4),ALM) CS
C * DATA ((ARAY(I,J),J=1,4),I=1,10) / CS
C X -5.0 , 320.65 , 1.75363 E 00 , -6.5 , CS
C X 0.0 , 286.15 , 1.00000 E 00 , -6.5 , CS
C X 11.0 , 216.65 , 2.23361 E-01 , 0.0 , CS
C X 20.0 , 216.65 , 5.40328 E-02 , 1.0 , CS
C X 32.0 , 220.65 , 0.56663 E-03 , 2.8 , CS
C X 47.0 , 270.65 , 1.09455 E-03 , 0.0 , CS
C X 52.0 , 270.65 , 5.82289 E-04 , -2.0 , CS
C X 61.0 , 252.65 , 1.79718 E-04 , -4.0 , CS
C X 79.0 , 180.6 , 1.02410 E-05 , 0.0 , CS
C X 88.743 , 180.65 , 1.62230 E-06 , 0.0 , / CS
C DATA REFT59/2.0855531E 07, GZ /9.80665/, CS
C A AMZ /28.9644 /, RSTAR /8.31432/, CS
C B FTTOKM/3.048E-04 /, S /110.4 /, CS

```

UBROUTINE ATMOS TRACE

CDC 6600 FTH V3.0-P351 OPT=1 06/22

```

      C      AMUZ /1.2024E-05 /, CAZ /1116.45/,      CS1
      D      RHOZ /0.076474 /, GZENG /32.1741/      CS1
C  CONVERT GEOMETRIC ALTITUDE TO GEOPOTENTIAL ALTITUDE      CS1
      HFT = (REFT59/(REFT59+ZFT))*ZFT      CS1
C  CONVERT HFT AND ZFT TO KILOMETERS      CS1
      Z = FTTOKM*ZFT      CS1
      H = FTTOKM*HFT      CS1
      K = 1      CS1
      TMZ = TMB(2)      CS1
      IF (H.LT.-5.0.OR.Z.GT.90.0) GO TO 16      CS1
      DO 10 M=1,10      CS1
      IF (H-HB(M)) 11,12,10      CS1
10  CONTINUE      CS1
      GO TO 16      CS1
11  M = M-1      CS1
12  DELH = H-HB(M)      CS1
      IF (ALM(M).EQ.0.0) GO TO 13      CS1
      TMK = TMB(M)+ALM(M)*DELH      CS1
C  GRADIENT IS NON ZERO, PAGE 10, EQUATION I.2.10-(3)      CS1
      DELTA = DELTAB(M)*((TMB(M)/TMK)**(GZ*AMZ/(RSTAR*ALM(M))))      CS1
      GO TO 14      CS1
13  TMK = TMB(M)      CS1
C  GRADIENT IS ZERO, PAGE 10, EQUATION I.2.10-(4)      CS1
      DELTA = DELTAB(M)*EXP(-GZ*AMZ*DELH/(RSTAR*TMB(M)))      CS1
14  THETA = TMK/TMZ      CS1
      SIGMA = DELTA/THETA      CS1
      ALPHA = SQRT(THETA**3)*((TMZ+S)/(TMK+S))      CS1
C  CONVERSION TO ENGLISH UNITS      CS1
      TH = 1.8*TMK      CS1
      RHO = RHOZ*SIGMA/GZENG      CS1
      CA = CAZ*SQRT(THETA)      CS1
      AMU = AMUZ*ALPHA/GZENG      CS1
C  CALL OVERFL(J)      CS1
C  GO TO (15,17), J      CS1
C  15 K = K+2      CS1
      GO TO 17      CS1
16  K = 2      CS1
17  RETURN      CS1
      END      CS1

```



```

SUBROUTINE PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQT)
PLSQ      POLYNOMIAL LEAST SQUARE CURVE FIT

      PLSQ WILL FIT A GIVEN SET OF DATA TO A
      POLYNOMIAL OF DEGREE K OF THE FORM...

      
$$Y=C(K+1)+C(K)*X+C(K-1)*X**2+...+C(2)*X**(K-1)+C(1)*X**K$$


      PLSQ THEN COMPUTES THE MAXIMUM ERROR AND ROOT
      MEAN SQUARE ERROR OBTAINED BY USING THE C
      COEFFICIENTS TO RE-COMPUTE Y FROM X

      USAGE...

      DIMENSION X(N), Y(N), C(L)
      WHERE L IS K+1
      CALL PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQT)

      WHERE,

      X  IS THE ARRAY OF N INDEPENDENT VARIABLES
      Y  IS THE ARRAY OF N DEPENDENT VARIABLES
      N  IS THE NUMBER OF INDEPENDENT(DEPENDENT)
          VARIABLES
      K  IS THE DEGREE OF THE LEAST SQUARES POLYNOMIAL
      C  IS THE ARRAY OF THE COEFFICIENTS,HIGH ORDER
          TO LOW ORDER, OF THE LEAST SQUARES POLYNOMIAL

      LIST =0  SUPPRESSES THE ERROR ANALYSIS OUTPUT
            =1  GIVES THE ERROR ANALYSIS OUTPUT

      EMAX IS THE MAXIMUM ABSOLUTE ERROR OBTAINED
            BY USING THE COMPUTED C COEFFICIENTS TO
            APPROXIMATE THE DEPENDENT VARIABLE

      ERMS IS THE ROOT MEAN SQUARE ERROR OBTAINED
            BY USING THE COMPUTED C COEFFICIENTS TO
            APPROXIMATE THE DEPENDENT VARIABLE

      EMEQT IS THE MAXIMUM DEVIATION FROM UNITY
            IN THE LINEAR SYSTEM CHECK SOLUTION

      PLSQ CALLS SUBROUTINES FXEM AND MTXEQT
      PLSQ USES 1309 CELLS OF BLANK COMMON

      COMMON  MTXEQT(664), CF, DIF, I, J, JC, JK,

```

```

      *          L, LL, LU, M, SUM, XI, XM(576),          CS1
      *          XMAX, XMIN, XP, YC, YH(48)              CS1
      DIMENSION  X(N), Y(N), C(24),                     CS1
      *          XDP(48), XYDP(24)                      CS1
      EQUIVALENCE (MTXEQT(1),XDP(1)), (MTXEQT(97),XYDP(1)) CS1
      DATA      KMAX/ 23/                               CS1
C
C      CHECK K AND N FOR PROPER RANGE                    CS1
C
      IF (K .GT. KMAX .OR. N .LE. K .OR. K .LE. 0)      CS1
        GO TO 200                                         CS1
      L=K+1                                               CS1
C
C      FIND MINIMUM AND MAXIMUM VALUES FOR X            CS1
C
      XMIN=X(1)                                           CS1
      XMAX=X(1)                                           CS1
      DO 10 I=2,N                                         CS1
      XMIN = AMIN1(XMIN,X(I))                             CS1
10    XMAX = AMAX1(XMAX,X(I))                             CS1
C
C      ZERO DOUBLE PRECISION ARRAYS FOR SUMMING         CS1
C
      M=2*K+1                                             CS1
      DO 20 I=1,M                                         CS1
20    XDP(I) = 0.0                                         CS1
      DO 25 I=1,L                                         CS1
25    XYDP(I) = 0.0                                       CS1
C
C      TRANSFORM RANGE OF X TO (-1,+1) AND              CS1
C      COMPUTE SUMS OF POWERS OF X AND SUMS             CS1
C      OF Y TIMES POWERS OF X                           CS1
C
      C1 = 2.0 / (XMAX - XMIN)                           CS1
      C2 = (XMAX + XMIN) / (XMAX - XMIN)                  CS1
      LL=K+2                                              CS1
      LU=2*K+1                                            CS1
      DO 40 I=1,N                                         CS1
      XP = 1.0                                            CS1
      XI = C1 * X(I) - C2                                CS1
      DO 30 J=1,L                                         CS1
      XDP(J)=XDP(J)+XP                                    CS1
      XYDP(J)=XYDP(J)+XP*Y(I)                            CS1
30    XP=XP*XI                                           CS1
      DO 40 J=LL,LU                                       CS1
      XDP(J)=XDP(J)+XP                                    CS1
40    XP=XP*XI                                           CS1
C
C      STORE ABOVE COMPUTED SUMS IN ARRAY XM            CS1
C      AND COMPUTE ROW SUMS FOR CHECK SOLUTION          CS1
C
      DO 50 I=1,L                                         CS1
      LL=I+L                                              CS1
      YH(LL)=0.0                                          CS1
      LU=(I-1)*L                                          CS1
      JK=I-1                                              CS1

```

```

      DO 50 J=1,L                      CS1
      JK=JK+1                          CS1
      JC=LU+J                          CS1
      XM(JC)=XDP(JK)                   CS1
50    YH(LL)=YH(LL)+XM(JC)            CS1
      DO 60 I=1,L                      CS1
60    YH(I)=XYP(I)                    CS1
      C                                CS1
      C    SOLVE THE SYSTEM XM*C=YH    CS1
      C                                CS1
      C    CALL MTXEQ(XH,YH,YH,L,2)    CS1
      C                                CS1
      C    REORDER AND MOVE SOLUTION TO C AND FIND    CS1
      C    MAXIMUM ERROR IN CHECK SOLUTION    CS1
      C                                CS1
      EMEQ=0.0                         CS1
      DO 70 I=1,L                      CS1
      JK=K-I+2                         CS1
      C(JK)=YH(I)                     CS1
      JC=I+L                           CS1
70    EMEQ=AMAX1(EMEQ,ABS(YH(JC)-1.0)) CS1
      C                                CS1
      C    ADJUST COEFFICIENTS FOR ORIGINAL RANGE    CS1
      C    OF X                          CS1
      C                                CS1
      DO 80 I=1,K                      CS1
      DO 80 J=1,I                      CS1
80    C(J) = C(J) * C1                CS1
      C1 = (XMAX + XMIN) / 2.0        CS1
      DO 90 I=1,K                      CS1
      M=L-I+1                          CS1
      DO 90 J=2,M                      CS1
90    C(J) = -C1 * C(J-1) + C(J)      CS1
      C                                CS1
      C    INITIATE PRINT OF ERROR ANALYSIS IF LIST .NE. 0    CS1
      C                                CS1
      IF (LIST.EQ.1) PRINT 1001        CS1
      C                                CS1
      C    COMPUTE MAXIMUM AND ROOT MEAN SQUARE ERRORS    CS1
      C    AND OUTPUT ERROR ANALYSIS IF LIST .NE. 0    CS1
      C                                CS1
      EMAX=0.0                         CS1
      SUM=0.0                          CS1
      DO 130 I=1,N                     CS1
      YC=C(1)                          CS1
      DO 100 J=1,K                     CS1
100   YC=YC*X(I)+C(J+1)               CS1
      DIF=YC-Y(I)                     CS1
      IF (LIST.EQ.0) GO TO 120          CS1
      IF (I .GT. L) GO TO 110          CS1
      PRINT 1002, I, X(I), Y(I), YC, DIF, C(I) CS1
      GO TO 120                        CS1
110   PRINT 1002, I, X(I), Y(I), YC, DIF CS1
120   EMAX=AMAX1(EMAX,ABS(DIF))        CS1
130   SUM=SUM+DIF**2                  CS1

```

SUBROUTINE PLSQ

TRACE

CDC 6600 FTM V3.0-P351 OPT=1 06/22

	ERMS=SQRT(SUM/FLOAT(N))	CS1
	IF (LIST.EQ.1) PRINT 1003, EMAX, ERMS, EMEQ	CS1
	RETURN	CS1
C		CS1
C	GIVE ERROR MESSAGE AND RETURN TO	CS1
C	SYSTEM VIA FXEM	CS1
C		CS1
200	PRINT 1000, N, K	CS1
	CALL SYSTEM (200,1L)	CS1
	RETURN	CS1
1000	FORMAT (3H0N=,I12,3H K=,I12,29HINCORRECT FOR SUBROUTINE PLSQ)	CS1
1001	FORMAT (1H1,20X,32HPLSQ POLYNOMIAL LEAST SQUARE	CS1
	*24HCURVE FIT ERROR ANALYSIS//	CS1
	*6H0 1,11X,9HX - GIVEN,11X,9HY - GIVEN,11X,	CS1
	*10HY - FITTED,12X,5HERROR,16X,4HB(I)//)	CS1
1002	FORMAT (1X,I5,8X,5(1PE14.6,6X))	CS1
1003	FORMAT (1H0,9X,5HEMAX=,1PE15.6,9X,5HERMS=,E15.6,	CS1
	*9X,5HEMEQ=,E15.6)	CS1
	END	CS1

```

SUBROUTINE MTXEQ(A,X,B,N,K)
C
C MATRIX EQUATION SOLVER (7094 FORTRAN IV)
C
C USAGE...
C
C TO SOLVE THE LINEAR SYSTEM AX=B
C
C CALL MTXEQ(A,X,B,N,K)
C
C WHERE A MUST BE DIMENSIONED N X N
C X MUST BE DIMENSIONED N X K
C B MUST BE DIMENSIONED N X K
C N IS THE NO. OF EQUATIONS (ROWS IN A,X,B)
C K IS THE NO. OF SOLUTION VECTORS (COLS. IN X,B)
C
C 664 CELLS OF BLANK COMMON ARE USED.
C
C NOTE... TO CHANGE DIMENSIONS OF ARRAYS C AND PIV, ALSO
C CHANGE VALUES OF NMAX AND NKMAX IN DATA STATEMENT.
C
C
C DIMENSION A(N,N), B(N,K), X(N,K)
COMMON ATPE, I, IFROM, IP1, IPIV, ITO,
1 J, KP, L, NP, NP1, NPJ, NPK, RM
COMMON PIV(26), C(24,26)
DATA NMAX, NKMAX/ 24, 26/
C
C GET ARGUMENTS N AND K.
C
C NP=N
C KP=K
C
C ST N AND K FOR CORRECT RANGE
C
C IF (NP.LE.0.OR.NP.GT.NMAX) GO TO 190
C IF (KP.LE.0.OR.(NP+KP).GT.NKMAX) GO TO 190
C
C MOVE ARRAYS A(I,J) AND B(I,J) INTO C(I,J)
C
C DO 10 J=1,NP
C DO 10 I=1,NP
10 C(I,J)=A(I,J)
C DO 20 J=1,KP
C NPJ=NP+J
C DO 20 I=1,NP
20 C(I,NPJ)=B(I,J)
C
C SET TO PERFORM N ELIMINATION SWEEPS (I=1,N)
C
C NP1=NP+1
C NPK=NP+KP
C DO 120 I=1,NP
C IP1=I+1
C

```

```

C      SEARCH FOR NEXT PIVOT ROW (I-TH PIVOT IS IN COL. I)      CS1
C      ATPE=0.                                                    CS1
      DO 60 J=I,NP                                                CS1
      IF (ABS(C(J,I))-ATPE) 40,30,30                               CS1
30     ATPE=ABS(C(J,I))                                           CS1
      IPIV=J                                                       CS1
40     CONTINUE                                                    CS1
C      OPERATE ON THE PIVOT ROW                                    CS1
C      IF (ATPE) 210,210,50                                       CS1
50     DO 60 J=IPIV,NPK                                           CS1
60     PIV(J)=C(IPIV,J)/C(IPIV,I)                                 CS1
C      PERFORM ELIMINATIONS BELOW THE DIAGONAL (COL. I)         CS1
C      IFROM=NP                                                    CS1
      ITO=NP                                                        CS1
70     IF (IFROM-IPIV) 80,100,80                                   CS1
80     RH=-C(IFROM,I)                                             CS1
      DO 90 J=IPIV,NPK                                           CS1
90     C(ITO,J)=C(IFROM,J)+RH*PIV(J)                               CS1
      ITO=ITO-1                                                    CS1
100    IFROM=IFROM-1                                              CS1
      IF (IFROM-I) 110,70,70                                       CS1
C      PUT THE I-TH PIVOT ROW IN THE VACATED ROW I              CS1
C      DO 120 J=IPIV,NPK                                           CS1
110    DO 120 J=IPIV,NPK                                           CS1
120    C(I,J)=PIV(J)                                              CS1
C      NOW DO THE BACK SOLUTION                                  CS1
C      I=NP                                                        CS1
130    IP1=I                                                       CS1
      I=I-1                                                         CS1
      IF (I) 150,150,150                                           CS1
140    DO 150 J=NP,IP1                                           CS1
      DO 150 L=IP1,NP                                              CS1
150    C(I,J)=C(I,J)-C(I,L)*C(L,J)                               CS1
      GO TO 130                                                    CS1
C      MOVE THE SOLUTION TO ARRAY X(I,J)                         CS1
C      DO 170 J=1,KP                                               CS1
160    NPJ=NP+J                                                    CS1
      DO 170 I=1,NP                                                CS1
170    X(I,J)=C(I,NPJ)                                           CS1
180    RETURN                                                    CS1
C      PRINT 1000, NP, KP                                         CS1
190    PRINT 1000, NP, KP                                         CS1
      CALL SYSTEM (200,1L )                                       CS1
      STOP                                                         CS1
210    PRINT 1001                                                 CS1

```

SUBROUTINE MTXEQ TRACE

CDC 6600 FTM V3.0-P351 OPT=1 06/22

CALL SYSTEM (200,1L)	CS1
STOP	CS1
1000 FORMAT (3H0N=I12,3H K=I12,3H ARE INCORRECT FOR SUBROUTINE MTXEQ)	CS1
1001 FORMAT (37H0DET(A)=0 IN CALL TO SUBROUTINE MTXEQ)	CS1
END	CS1