

12 LEVEL III  
SC

AD-E300 442

DNA 4531F-2

AD A 0641 35  
**AERODYNAMIC ACCOUNTING TECHNIQUE  
FOR DETERMINING EFFECTS OF NUCLEAR  
DAMAGE TO AIRCRAFT**

**Volume II - Program User Guide**

General Dynamics Corporation  
Fort Worth Division  
P.O. Box 748, Grants Lane  
Fort Worth, Texas 76101

THIS DOCUMENT IS BEST QUALITY PRACTICABLE  
THE COPY FURNISHED TO DDC CONTAINED A  
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.

28 February 1978

Final Report for Period 15 March 1977-30 January 1978

CONTRACT No. DNA 001-77-C-0075

APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY  
UNDER RDT&E RMSS CODE B342077464 N99QAXAE50602 H2590D.

Prepared for  
Director  
DEFENSE NUCLEAR AGENCY  
Washington, D. C. 20305

DDC  
RECEIVED  
FEB 5 1979  
RECEIVED

JA

B

79 01

DDC FILE COPY

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY,  
ATTN: TISI, WASHINGTON, D.C. 20305, IF  
YOUR ADDRESS IS INCORRECT, IF YOU WISH TO  
BE DELETED FROM THE DISTRIBUTION LIST, OR  
IF THE ADDRESSEE IS NO LONGER EMPLOYED BY  
YOUR ORGANIZATION.



## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY  
PRACTICABLE. THE COPY FURNISHED  
TO DDC CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 4531F-2 ✓	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) AERODYNAMIC ACCOUNTING TECHNIQUE FOR DETERMINING EFFECTS OF NUCLEAR DAMAGE TO AIRCRAFT, Volume II, Program User Guide.	5. TYPE OF REPORT & PERIOD COVERED Final Report, for Period 15 Mar 77-30 Jan 78.	
7. AUTHOR(s) Roy/Schemensky George/Howell	8. CONTRACT OR GRANT NUMBER(s) DNA 001-77-C-0075	6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS General Dynamics Corporation Fort Worth Division, P.O. Box 748, Grants Lane Fort Worth, Texas 76101	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask N99QAXAE506-02 (12) E506	
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305	12. REPORT DATE 28 February 1978	13. NUMBER OF PAGES 252
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 253p.	15. SECURITY CLASS (of this report) UNCLASSIFIED	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. (18) DNA, SBIE (19) 4531F-2, AD-E300 442		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B342077464 N99QAXAE50602 H2590D.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Drag Prediction      Roughness      Computer Program Lift Prediction      Blast      Nuclear Damage Aircraft      Overpressure      This volume Damage      Thermal Effects		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report (Volume II, Program User Guide) describes the computer program developed for the aerodynamic evaluation of aircraft, including the effects of roughness and damage. The computer code has the capability of assessing the aerodynamic effects of damage, such as rough, bent, and burnt skins, boundary-layer thickness effects, and loss of radomes, panels, doors and covers. Also, the computer code has the capability of analyzing changes in drag due to lift and trim caused by asymmetric loss of parts of the wing or trim surfaces.		

402709

LB

over

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued)

\*The computer program is coded in the FORTRAN Extended Version IV language to operate on a CDC 6600 or CYBER 172 computer. The computer code structure is explained and a computer code input utilization is presented along with sample input and output with a FORTRAN source deck listing. Details of the methods, equations, and substantiating data for this computer code are contained in Volume I, Empirical Methods.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS  
Volume II

		<u>Page</u>
Section 1.	GENERAL DESCRIPTION	3
Section 2.	AAT COMPUTER CODE UTILIZATION	5
Section 3.	SAMPLE PROBLEMS	39
Section 4.	PROGRAM DESCRIPTION	91
	4.1 Program Structure	91
	4.2 Subroutine Descriptions	91
	4.3 Common Block Descriptions	111
	4.4 Program Listing	126

ACCESSION for		
NTIS	Volume Section	<input checked="" type="checkbox"/>
DDC	Page Section	<input type="checkbox"/>
UNANNOUNCED		<input type="checkbox"/>
JCS/JRNL/ER		
BY		
DISTRIBUTION/CLASSIFICATION CODES		
Dist: REPRODUCIBLE SPECIAL		
A	23	



## 1. GENERAL DESCRIPTION

The Aerodynamic Accounting Technique Computer Code was developed to provide a computerized, systematic method to evaluate the aerodynamic effects of nuclear damage to an aircraft. In addition, it can be used to evaluate the aerodynamic characteristics of an undamaged configuration or any type damage that can be modeled into one of the fourteen damage modes provided by the program.

The computer program is coded in the Fortran Extended Version IV language to operate on the CDC 6600 computer at Wright-Patterson Air Force Base. It is also operational on General Dynamics' CDC CYBER 172 computer as procedure R7F. Four primary overlays are used by the program to keep the central memory core requirements below 54,000 bytes. The main program controls the calling of the four overlay programs: XINPT, GEOM, SURVEY, and NUCDAM. Figure 1.1 shows the program overlay structure and the arrangement of the 57 subprograms and subroutines that comprise the AAT computer code. Program XINPT controls the reading of input data and the printout of data that will be used in the problem. Program GEOM computes the geometric parameters that are required in the calculations. Program SURVEY controls the calculation of aerodynamic characteristics of the undamaged aircraft, and program NUCDAM determines the aerodynamic effects that result from the damage. The third primary overlay, SURVEY, calls four secondary overlay programs VGEOM, MCRIT, AEROA, and AEROB. These programs and subprograms call the appropriate subroutines necessary to make the computations.

The program utilization describes the inputs that are required. Aerodynamic characteristics of an undamaged configuration can be estimated by entering only the basic geometric parameters and the survey conditions. Aircraft damage evaluation is optional. The damage input data are contained in a separate data block and the effects of damage are determined on an incremental basis and integrated into the aerodynamics of the undamaged configuration.



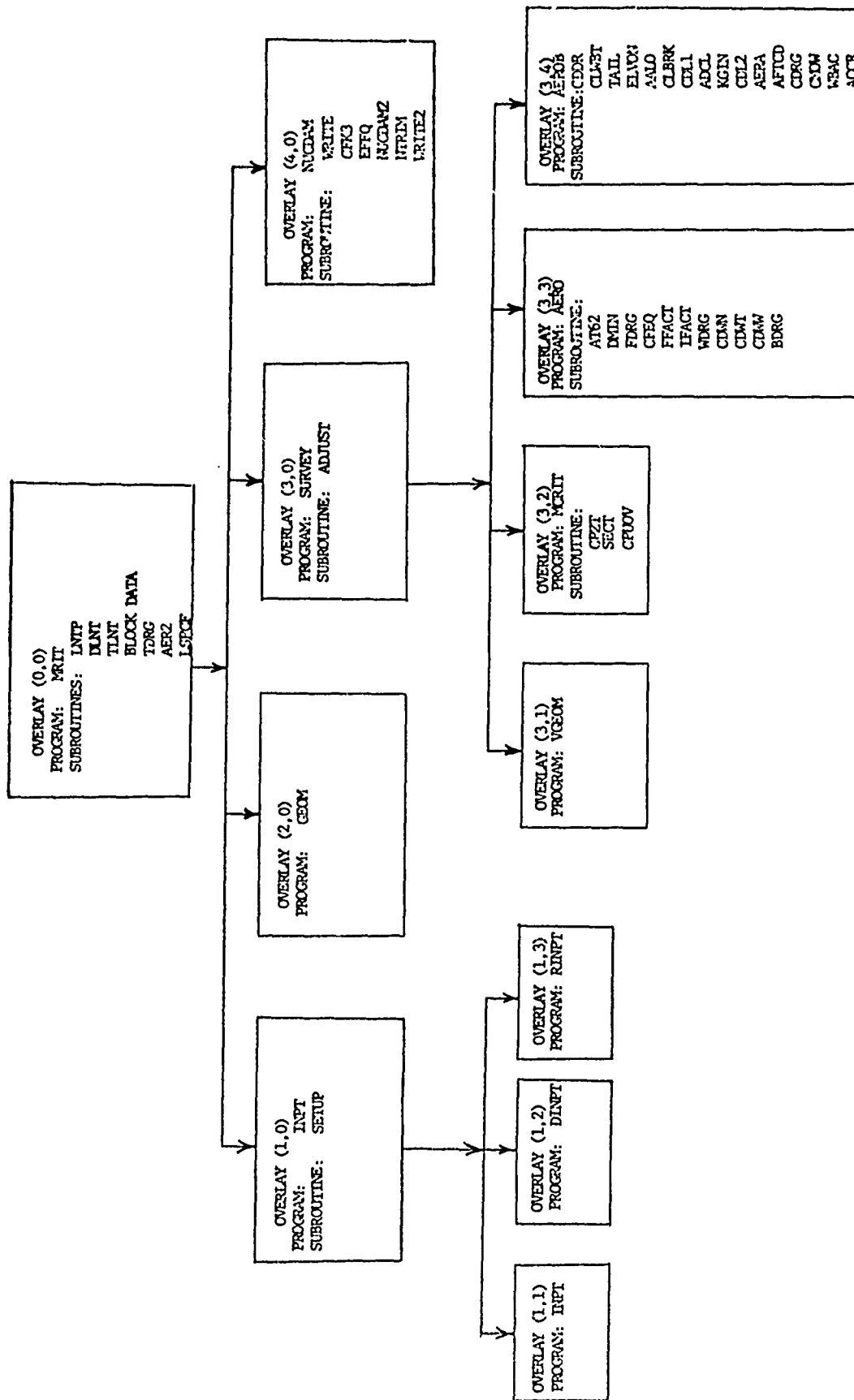


FIGURE 1-1 PROGRAM OVERLAY STRUCTURE

## 2. AAT COMPUTER CODE UTILIZATION

The first step in preparing program input is to determine how to best represent the configuration with circular bodies and surfaces. Components such as the fuselage, canopy, stores, external fairings, and nacelles are represented as bodies, while the wing, horizontal tail, vertical tail, pylons, and ventral fins are represented as surfaces. A straight-wing planform is represented by one panel, and a cranked- or complex-wing planform is approximated with two panels. Up to seven bodies and seven surfaces may be used to represent a configuration. Figures 1 and 2 define some of the geometric parameters required to define the individual bodies and surfaces.

Input requirements for the Aerodynamic Accounting Technique computer code are partially determined by the options selected by the user. The input format is divided into 12 blocks to assist the user in determining his input requirements. The user may decide which blocks are necessary to complete his problem while not concerning himself with the remaining blocks. The user determines which data blocks are required from the following descriptions:

1. General Information. This block contains information that the AAT program needs to read the remainder of the input data. Indicators, such as the number of bodies and surfaces, are included, which tell the program which blocks of data will follow. Certain reference data and geometric dimensions are also included. This block is always required.
2. Body Geometry. The geometric parameters of each component which is represented as a body is contained in this block. A maximum of seven bodies may be used.
3. Surface Geometry. The geometry of each surface is represented in this block. A maximum of seven panels may be used, with either one or two panels for the main wing.
4. Arbitrary Airfoil. This block is used to describe the camber and thickness distribution of an arbitrary airfoil section if one of the standard sections contained internally is not sufficient.
5. Variable Sweep. Aircraft with movable wing panels such as the F-111 and B-1, may be evaluated at any selected sweep angle. This block of data is required to describe the wing movement.
6. Survey Conditions. This block of data contains the Mach number and Reynolds number conditions at which the analysis will be conducted.

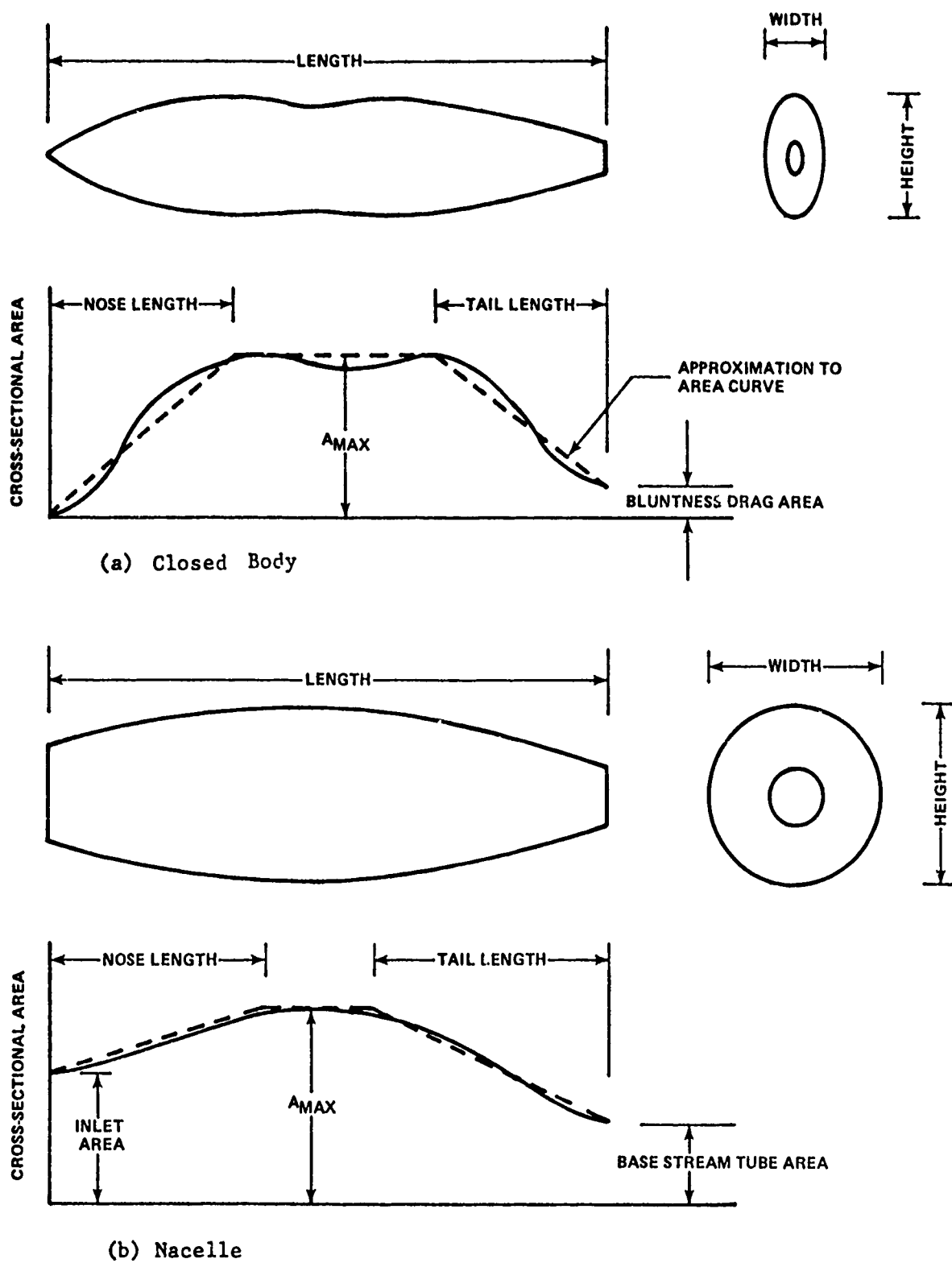
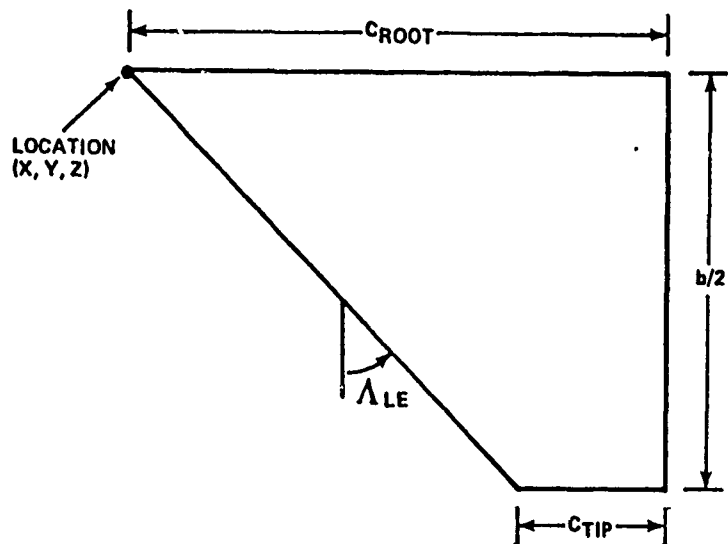
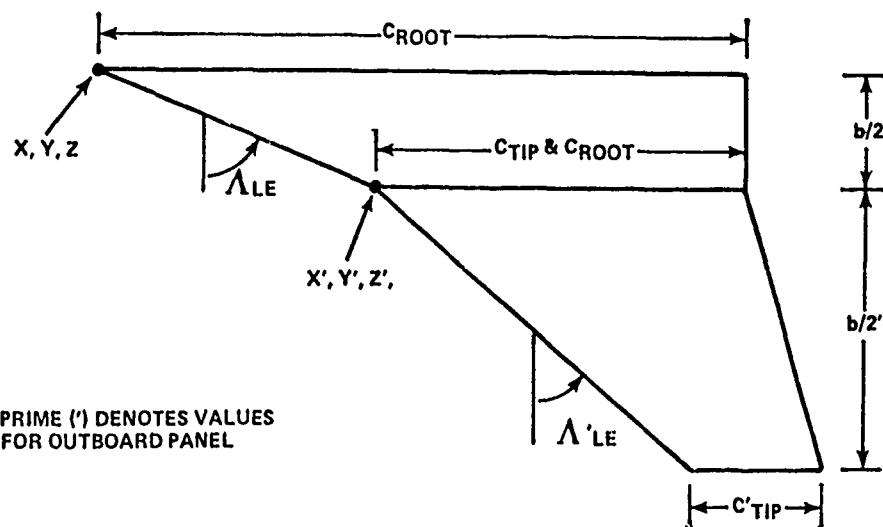


Figure 2-1 Geometry for Closed Bodies and Nacelles



(a) Single-Panel Wing or Other Surface (Exposed Area)



(b) Two-Panel Wing (Exposed Area)

Figure 2-2 Surface Geometry

7. Adjustment Factors. A table of adders and multipliers can be specified in this block to correlate the AAT predictions with known test levels. At least one card is always required.

8. Expansion Provisions. This data block is provided for future expansion.

9. Damage Mode Indicators. Provisions are included for fourteen modes of nuclear damage in the AAT program. Seven modes are for body damage and seven for surface damage. This block, which requires three cards, allows the user to select only those modes that are desired.

10. Body Damage Parameters. This block of data contains parameters that define nuclear damage to the bodies. It is required only if a body damage mode is indicated in Block number 9.

11. Surface Damage Parameters. This block describes nuclear damage to the surfaces and is similar in format to Block 10. It is required only if a surface damage mode is indicated in Block 9.

12. End of Problem. Describes how one may end the problem or change the input and run new problems.

These twelve blocks of data represent a wide range of options for defining the geometry and the manner in which the data are handled. Most aircraft analyses will not require all blocks.

It should be noted that there is an option that allows the program to compute the values of certain parameters that are shown as input. These items are indicated with an asterisk (\*). If a zero or blank is input, the program will use other geometric parameters to compute these values. All dimensions are input in either inches, feet or meters, except for the roughness factor, which is always in inches. All angles are in degrees. The utilization specifies one of the following formats for each item of input:

"I" Format     Input must be in integer form and right adjusted in the specified field.

"F" Format     Input must include a decimal and may be anywhere within the specified field.

"A" Format     Input may include any alphanumeric characters.

Data Block No. 1, General Information

Data Block No. 1 requires from 5 to 7 cards, depending on the options selected.

CARD 1.1 - Title. Enter any alphanumeric characters in columns 1 through 60 to identify each problem. These characters will be printed out at the top of each page of output for the undamaged configuration.

CARD 1.2 - Print-out options. This card should normally be left blank; however, certain seldom-used data and subroutine dumps will be printed if KPRINT(I)=1. The subroutine dumps are normally for diagnostic purposes and can be used only in conjunction with the programming statements. If KPRINT(31)=1, the aerodynamic data for both the damaged and undamaged configurations will be printed at constant values of angle of attack.

<u>COLUMN</u>	<u>SYMBOL</u>	<u>PRINT-OUT</u>
1	KPRINT(1)	Airfoil ordinates and pressure distribution
11	(11)	Dump Subroutine AER2
12	(12)	GEOM
13	(13)	AALO
14	(14)	CDL2
15	(15)	BDRG
16	(16)	CLBRK
17	(17)	AERA
18	(18)	WBAC
19	(19)	CFEQ
20	(20)	CDWW
21	(21)	TAIL
22	(22)	CDL1
23	(23)	CDDR
24	(24)	ADJUST
25	(25)	CMOW
26	(26)	CPZT
27	(27)	SSET
28	(28)	NUCDAM
29	(29)	NUCDAM2
30	(30)	NTRIM
31	(31)	WRITE2

CARD 1.3 - Configuration Definition (I5 Format).

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
5	NPODS	Number of different <u>types</u> of bodies. NPODS $\leq$ 7.
10	NPNS	Number of surfaces used to represent the main wing. Must be 1 or 2.
15	NHT	Number of lifting surfaces other than main wing, i.e., horizontal tail or canard. Program assumes that these surfaces are symmetric with respect to configuration centerline.
20	NVT	Number of non-lifting surface <u>types</u> , i.e., vertical tail, ventral fin, or pylon. For example, a twin vertical tail should be entered as 1 surface type; symmetry will be indicated in Data Block No. 3. NPNS+NHT+NVT $\leq$ 7.
25	ISWP	Variable-sweep indicator. 0 for fixed wing 1 for variable-sweep wing
30	IREF	Reference angle-of-attack indicator. 0 reference to wing root-chord plane 1 reference to fuselage centerline
35	IWNG	WING-definition indicator. 0 omit card 1.6 1 enter reference planform area, taper ratio, and leading-edge sweep on Card 1.6 See note number 1 at end of input instructions for Data Block No. 1.
39-40	NAFO	Number of stations at which thickness and camber are to be defined for an arbitrary airfoil, $\leq$ 30. If NAFO $>$ 0, Data Block No. 4 must be input.
45	METER	Unit-system indicator 0 use units of feet 1 use units of meters 2 use units of inches, except for reference area, planform area, wetted areas, Reynolds number, and altitude which are in feet.
50	ITRIM	Trim indicator. 0 trim configuration 1 do not trim

CARD 1.4 - "F" Format.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	SREF	Reference area.
11-20	CMAC*	Reference aerodynamic chord.
21-30	XMAC*	Location for leading-edge of CMAC. Does not change with sweep.
31-40	ZCG	Height of moment reference point relative to wing root-chord plane.
41-50	UPFUS	Upsweep angle of the aft fuselage (deg). For use with transport-type aircraft.
51-60	AOB	Ratio of width to height of the aft fuselage in the upswept region.

CARD 1.5 - (F10 Format).

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	ROUGHK	Average surface roughness height for friction drag, (inches).
11-20	FMISC	Miscellaneous drag factor as a percentage of total friction and form drag.
21-30	TWIST	Wing twist between exposed root chord and tip chord. Negative for washout (wing-tip leading edge down).
31-40	CONCL	Wing conical camber design lift coefficient.

CARD 1.6 (F10 format). This card is required only if IWNG=1 on Card 1.3.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	SPLAN	Wing theoretical planform area.
11-20	TAPR	Reference wing theoretical taper ratio.
21-30	SWP	Reference wing leading-edge sweep.

\*An asterisk beside a symbol indicates that the value of that item will be computed internally if a blank or zero is input.



CARD 1.7 (F10 Format) Elevon Definition. Required if NHT=0 and ITRIM=0. This card defines the elevon, which will be used to trim the configuration if there is no horizontal tail or canard.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	CFOC	Elevon chord length, $\Delta c/c$
11-20	EI	Elevon inboard edge, expressed as a fraction of semi-span.
21-30	EO	Elevon outboard edge, expressed as a fraction of semi-span.

Notes pertaining to the input for Data Block No. 1 are as follows:

1. Card 1.6 is used to input the wing planform area, taper ratio, and leading-edge sweep when the user desires to override the internal calculations that normally compute these values based on other wing parameters. Equations used to make these calculations are presented in Section 2.1.3 of Volume I. Card 1.6 should not be required for a single-panel wing or for a two-panel wing with a moderate amount of crank. It is recommended, however, when two panels are used to represent a wing with an extreme amount of crank or a strake/wing planform where the strake is highly swept.
2. The input reference area is arbitrary and does not have to be related to the theoretical wing planform area.

## Data Block No. 2, Body Geometry

Thirteen cards are used to input the name and 12 parameters for each of the bodies. The name that is input on Card 2.1 will be used in the printout to identify the various components. Data for each body are listed vertically in the input. The main body (fuselage) should be listed to the extreme left, (Column 1-10), followed next by other bodies with zero inlet area. Nacelle data should be listed last. Up to seven columns for seven different body types can be input.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
2.1	7A10	BNAME(I)	Body I name, I=1, NPODS
2.2	7F10	BOD(I,1)	Length
2.3		(I,2)	Width
2.4		(I,3)	Height
2.5		(I,4)*	Wetted area (total for all bodies of type I).
2.6		(I,5)	Interference factor
2.7		(I,6)	Number of bodies of type I
2.8		(I,7)*	Maximum cross-sectional area
2.9		(I,8)	Base streamtube area
2.10		(I,9)	Nose length
2.11		(I,10)	Boattail length
2.12		(I,11)	Base area for bluntness drag
2.13		(I,12)	Inlet area.

There are several points that should be mentioned pertaining to Data Block No. 2.

1. The name assigned to each body will be used in the printout of results.

2. The following is presented as a guide for determining the interference factor (BOD(I,5)):

=1.0 for nacelles and external stores mounted out of the local velocity field of the wing and fuselage.

=1.25 for external stores mounted symmetrically on the wing tip.

=1.3 for nacelles and external stores if mounted in moderate proximity of the wing.

=1.5 for nacelles and external stores mounted flush to the wing.

(2., continued)

The same variation of the interference factor applies in the case of a nacelle or external store strut-mounted to or flush-mounted on the fuselage.

3. The length, width, height, cross-sectional area, etc. listed for each body type describes a single body. The total number of bodies of each type must also be specified even though they may be paired symmetrically with respect to the centerline. One exception is that the wetted area represents the total for all bodies of a particular type.

4. Bodies must be defined for I=1 to NPODS. The main body must be first, closed bodies second, and bodies with inlet area last.

5. When fewer than seven bodies are defined, columns to the right of the last field may be conveniently used to identify the parameter listed on that card.

6. The interference factor for the fuselage that is entered on Card 2.6 is overridden by internal calculations for wing/body interference.

### Data Block No. 3, Surface Geometry

Thirteen cards are used to input the name and 12 parameters for each of the surfaces. The format for the surface data is similar to that for the bodies.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
3.1	7A10	SNAME(I)	Surface name I, I=1, NPNLS+NHT+NVT
3.2	7(A3,7X)	SUR(I,1)	Airfoil section
3.3	7F10	(I,2)	2-D design lift coefficient
3.4		(I,3)	Thickness ratio, (t/c) <sub>RMS</sub>
3.5		(I,4)	Leading-edge sweep
3.6		(I,7)*	Total surface(s) wetted area
3.7		(I,8)	Exposed root chord
3.8		(I,9)	Tip chord
3.9		(I,10)	Exposed semi-span
3.10		(I,11)	X-station at LE of exposed root chord
3.11		(I,12)	Y-station at LE of exposed root chord
3.12		(I,13)	Z-station at LE of exposed root chord
3.13		(I,14)	Incidence with respect to main body

Several points that should be noted regarding Data Block No. 3 are:

1. The name assigned to each surface will be used in the printout of results.
2. Surfaces must be input from left to right in the following order: NPNLS main wing panels, NHT lifting surfaces, and finally, NVT non-lifting surfaces.
3. Twenty options are available for defining the airfoil section type:

<u>INPUT</u>	<u>AIRFOIL TYPE*</u>	<u>INPUT</u>	<u>AIRFOIL TYPE*</u>
63-	63-YXX(6 Series)	-62	00XX-62(4 Digit)
64-	64-YXX "	-63	00XX-63 "
65-	65-YXX "	-64	00XX-64 "
66-	66-YXX "	-65	00XX-65 "
63A	63AYXX(6A Series)	-33	00XX-33 "
64A	64AYXX "	-34	00XX-34 "
65A	65AYXX "	-35	00XX-35 "
BIC	Biconvex	-93	00XX-93 "
INPUT	Arbitrary Airfoil	-94	00XX-94 "
		-95	00XX-95 "

\*Y= Section Camber in Tenths

XX\* Section t/c in % Z/C

(3., continued)

Any of these sections may be input to the program with the identification shown above. The characters must, however, be left-adjusted in the data field.

An arbitrary section may be input for any surface by entering "INPUT". This will be used for both wing panels if NPNLS=2. Use of "INPUT" requires that Data Block No. 4 be included and that NAFO > 0 on Card 1.4. Only one arbitrary section can be defined, but this section can be used for any of the surfaces by showing "INPUT" on Card 3.2.

4. Symmetry is controlled in the following manner: NPNLS and NHT surfaces are automatically assumed to have counterparts on the opposite side of the fuselage. The NVT non-lifting surfaces are assumed to be single surfaces if  $y=0$  (SUR(I,12)). If  $y > 0$ , an identical surface is assumed to be on the opposite side. Irregardless of symmetry, note that the wetted area must be the total value, not a single, one-sided value.

### Data Block No. 4, Arbitrary Airfoil

Data Block No. 4 must follow immediately after Data Block No. 3 if NAFO > 0 on Card 1.4,

#### CARD 4.1 - (F10 format)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	RLE	Leading-edge radius divided by chord (RLE/CHORD), at reference t/c (TOCR).
11-20	TOCR	Reference t/c of AFT (see Card 4.4).
21-30	CLDR	Reference design lift coefficient of AFC (see Card 4.3).
31-40	TECH	Technology Factor (See Sections 5.1 and 6.2 of Volume I) = 0.0 Conventional airfoil = 1.0 Advanced supercritical wing section

#### CARD 4.2

<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
7F10	AFX(I)	Chordwise stations at which the camber and thickness data are to be entered. Enter as x/c. (I=1,NAFO)

Card 4.2 should be repeated until NAFO (Card 1.4) ordinates have been entered.

#### CARD 4.3

<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
7F10	AFC(I)	Camber of the arbitrary airfoil section. Defined by Z/C where Z is the distance from the chord line to the camber line. (I =1, NAFO)

Card 4.3 should be repeated in the same format as Card 4.2 until NAFO values of AFC(I) have been input. The camber ordinates must correspond with the chordwise stations on Card 4.2.

CARD 4.4

<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
7F10	AFT(I)	Thickness of the arbitrary airfoil section. Defined by $Z/C$ , where $Z$ is the distance from the chord line to the upper or lower surface on an uncambered airfoil. (I=1,NAFO)

Card 4.4 should be repeated in the same format as Card 4.2 until NAFO values of AFT(I) have been input. The thickness ordinates must correspond with the chordwise stations on Card 4.2.

Data Block No. 5., Variable Sweep

Data Block No. 5 is required only if the variable-sweep option is indicated by ISWP=1 on Card 1.4. This Block contains only one card. Use of the variable-sweep option does not change the reference area or the reference moment center, which are defined in Data Block No. 1.

CARD 5.1 - (F10 format)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	XPIVOT	X-location of wing pivot
11-20	YPIVOT	Y-location of wing pivot
21-30	AFTSW	Maximum aft sweep
31-40	AFTCB*	Mean aerodynamic chord of movable panel in aft-sweep position.
41-50	AFTOC*	Thickness ratio (t/c) of movable panel in aft-sweep position.
51-60	AFTAW*	Wetted area of movable panel in aft-sweep position.



### Data Block No. 6, Survey Conditions

This data block is always required, and will consist of from 2 to 22 cards, dependent upon the number of survey conditions specified.

#### CARD 6.1 - (I5 format)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
4-5	NSURV	Number of surveys (sets of conditions) at which the problem is to be run ( $NSURV \leq 20$ ).
9-10	NCLAS	Number of evenly spaced $C_L$ values for which data will be computed ( $NCLAS \leq 21$ ).

#### CARD 6.2 - (F10 format)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	MSURV(I)	Mach number
11-20	ALT(I)	Altitude
21-30	CG(I)	Position for trim or moment reference. Measured as a fraction of the mean aerodynamic chord (CMAC) relative to leading-edge of MAC.
31-40	SWPV(I)	Leading-edge sweep.
41-50	CLLO(I)	Low $C_L$
51-60	CLHI(I)	High $C_L$

Card 6.2 is repeated until NSURV conditions have been specified. Several items that may prove useful to the user are:

1. The Reynolds number may be specified instead of altitude by entering the negative value of  $RN/ft$  divided by  $10^6$ . For example, if  $RN/ft=3.0 \times 10^6$ , enter -3.0 in Column 11-20.

2. Model-scale predictions can be obtained even though full-scale geometric data are loaded in the program. Enter the negative of the  $RN/ft$  multiplied by model scale divided by  $10^6$ . For example, if  $RN/ft=3.0 \times 10^6$  and model scale desired is 1/15, enter -0.2 in Columns 11-20 of Card 6.2.

3. The low  $C_L$  and high  $C_L$  should be defined in conjunction with NCLAS so that the calculated values of  $C_L$  will be rounded off to convenient numbers. For example,

<u>INPUT</u>	<u>Data Calculated at</u>
NCLAS=21	$C_L = 0.00$
CLLO=0.0	$C_L = 0.05$
CLHI=1.0	$C_L = 0.10$
	$C_L = 0.15$
	.
	.
	.
	$C_L = 1.00$

4. If low  $C_L$  and high  $C_L$  are left blank, the program will automatically set CLLO=0.0 and CLHI=1.0.

### Data Block No. 7, Adjustment Factors

Data Block No. 7 must always be entered in the input; it may, however, contain only one card which indicates that no adjustments will be made.

#### CARD 7.1

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 "NO ADJUSTMENTS" beginning in Column 1 and continue with the next data block.

The adjustment options allow 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 (APRED) can be adjusted to match an experimental value ( $A_{EXP}$ ) by the equation

$$A_{EXP} = (APRED) \cdot YM + YA$$

where YM is an input correlation multiplier and YA is an adder. Each is a function of Mach number or lift coefficient.

#### CARD 7.2 - (I format)

Set a given IVAL indicator equal to 1 to identify it as being an aerodynamic parameter to be adjusted.

<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 $\alpha_{LO}$ as a function of Mach number.
4	IVAL(4)	Adjust MCR as a function of lift coefficient.
5	IVAL(5)	Adjust $C_{DL}$ as a function of lift coefficient.
24-25	NXVAR	Number of Mach values in the table of Mach function adjust factors ( $\leq 15$ ).

(CARD 7.2 - (I format), continued)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
29-30	NADJ	Number of parameters to be adjusted as a function of Mach number ( $\leq 3$ ).
34-35	NXCL	Number of $C_L$ values in the table of lift function adjust factors ( $\leq 15$ ).
39-40	NADJ2	Number of parameters to be adjusted as a function of lift coefficient ( $\leq 2$ ).

CARD 7.3 - (7F10.0 format) Required if NADJ > 0.

<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)	
61-70	X(7)	

Repeat Card 7.3 until NXVAR values of X are read in.


CARD 7.4 (6F10.0 format) (NADJ > 0)

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
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)	Multiplier factor.
31-40	YA(J=2, I=1)	Adder factor.
41-50	YM(J=3, I=1)	Multiplier factor.
51-60	YA(J=3, I=1)	Adder factor.

Repeat Card 7.4 until NXVAR values are read in (J=NXVAR) for each aerodynamic parameter to be adjusted (I=1 to NADJ).

CARD 7.5 - (7F10 format) Required if NADJ2 > 0.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	XCL(1)	CL 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)	
61-70	XCL(7)	



Repeat Card 7.5 until NXCL values of XCL are read in.

CARD 7.6 - (6F10.0 format) NADJ2 0.

<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)	Multiplier factor.
31-40	YA(J=2, I=NXVAR+1)	Adder factor.
41-50	YM(J=3, I=NXVAR+1)	Multiplier factor.
51-60	YA(J=3, I=NXVAR+1)	Adder factor.

Repeat Card 7.6 until NXCL values are read in (J=NXCL).

Data Block No. 8, Reserved for Expansion

Data Block No. 8 represents a section of the input common block that has been reserved for future expansion. The user should ignore this block.

Data Block No. 9, Damage Mode Indicators

If no damage parameters are to be loaded, skip the remaining input and go to "End of Problem" instructions in Data Block No. 12. If damage parameters are to be loaded for either a body or a surface, complete the following three cards.

CARD 9.1

Enter "DAMAGE CASES FOLLOW" beginning in Column 1.

CARD 9.2

Enter any alphanumeric title in Columns 1 through 60. This title will be printed at the top of each page of output related to damage calculations.

CARD 9.3 - (I format)

This card allows the user to select the modes of damage that will be used. Set IDAM(I)=1 if mode I input are to be entered in Blocks 10 and 11.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1	IDAM(1)	Mode 1, Roughness on bodies
2	(2)	2, Fwd-facing steps on bodies
3	(3)	3, Aft-facing steps on bodies
4	(4)	4, Holes in bodies
5	(5)	5, Surface waviness on bodies
6	(6)	6, Protuberances on bodies
7	(7)	7, Nose bluntness on bodies
11	IDAM(11)	Mode 11, Roughness on surfaces
12	(12)	12, Fwd-facing steps on surfaces
13	(13)	13, Aft-facing steps on surfaces
14	(14)	14, Holes in surfaces

(CARD 9.3 - I format), continued

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
15	IDAM(15)	Mode 15, Surface waviness on surfaces
16	↓ (16)	↓ 16, Protuberances on surfaces
17	↓ (17)	↓ 17, Missing parts of surfaces



Data Block No. 10, Body Damage

This block is required if one or more of the indicators IDAM(1) through IDAM(7) equals 1, otherwise, it can be omitted. Several cards are required to define the body damage for each mode selected. The input format is similar to that in Block No. 2. If only part of the bodies are damaged, the columns for the undamaged bodies are to be left blank. The space to the right of the last field of data may be used to identify the damage parameter on that card.

CARD 10.1

Enter the body names in 10-column fields and in the same order as on Card 2.1

Body roughness, required if IDAM(1)=1.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.2	7F10	DBOD(I,1)	Roughness factor on body I <u>before</u> damage is incurred.
10.3		(I,2)	Roughness factor on body I <u>after</u> damage is incurred.
10.4		(I,3)	x/1 where damage roughness starts
10.5		(I,4)	x/1 where damage roughness ends
10.6		(I,5)	Fraction of area affected by increased roughness (includes area forward and aft of damaged region).

Table 1 is presented as an aid in selecting the roughness factors after damage has been incurred.

Fwd-facing steps, required if IDAM(2)=1

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.7	7F10	DBOD(I,11)	Number of fwd-facing steps on body I.
10.8		(I,12)	Width of each step.
10.9		(I,13)	Height of each step.
10.10		(I,14)	x/1 at first step.
10.11		(I,15)	x/1 at last step.

Aft-facing steps, required if IDAM(3)=1.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.12	7F10	DBOD(I,21)	Number of aft-facing steps on body I.
10.13		(I,22)	Width of each step.
10.14		(I,23)	Height of each step.
10.15		(I,24)	x/1 at first step.
10.16		(I,25)	x/1 at last step.

Holes, required if IDAM(4)=1.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.17	7F10	DBOD(I,31)	Total no. of holes in body I.
10.18		(I,32)	Number of these holes over wing.
10.19		(I,33)	Length of each hole.
10.20		(I,34)	Width of each hole.
10.21		(I,35)	Depth of each hole.
10.22		(I,36)	x/1 where first hole starts.
10.23		(I,37)	x/1 where last hole starts.
10.24		(I,38)	Type hole 1.0 Missing Panel 2.0 Caved-in Panel

Waviness, required if IDAM(5)=1.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.25	7F10	DBOD(I,41)	Total no. of waves in body I.
10.26		(I,42)	No. of these waves over wing.
10.27		(I,43)	Length of each wave.
10.28		(I,44)	Width of each wave.
10.29		(I,45)	Amplitude of each wave.
10.30		(I,46)	x/1 where first wave starts.
10.31		(I,47)	x/1 where last wave starts.

Protuberances, required if IDAM(6)=1.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.32	7F10	DBOD(I,51)	No. of protuberances on body I.
10.33		(I,52)	Height of each protuberance.
10.34		(I,53)	Parasite area of each protuberance, $\Delta f$ .
10.35		(I,54)	x/l of first protuberance.
10.34		(I,55)	x/l of last protuberance.

Bluntness, required if IDAM(7)=1.

<u>Card</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
10.37	7F10	DBOD(I,61)	Frontal area of blunted body at point of damage. For example, a fuselage with the radome missing might have a cross-section area of 10 sq ft at the point where the radome is detached.
10.38		(I,62)	l/d at point of damage. d is the body diameter at most forward body section that is not damaged by the bluntness. l is the remaining nose length forward of the point where d is measured. For example, l/d=0.0 for a nose section that is flat and l/d=0.5 for a hemi-spherical section.

All input parameters describe damage to only one body of type I. If all bodies of type I have sustained the damage described, a negative sign on the first parameter in each mode of damage will indicate this. For example, if there is increased roughness on only one of four wing-mounted pods, DBOD(I,1) through DBOD(I,5) are all positive values. If all four pods are damaged, this is indicated by entering DBOD(I,1) as a negative number.

The parasite area for a protuberance is the drag divided by the dynamic pressure (D/q) in freestream air flow. Effects of local boundary layer are computed. Units for  $\Delta f$  must be consistent with those selected on Card 1.3.

TABLE 1 TYPICAL ROUGHNESS VALUES

<u>Surface or Condition of Surface</u>	<u>Roughness k ~ Inch</u>
Average aircraft wing or tail surface	.0006
Average fuselage, nacelle surface	.0012
Aluminum skin with blistered paint	.0012
Fiberglass/Enamel with blistered paint	.0025
Fiberglass/Thick Coatings and Graphite with blistered paint	.0030
Broken Skin	.01
Exposed Honeycomb	.1

Data Block No. 11, Surface Damage

This block is required if one or more of the indicators IDAM(11) through IDAM(17) equals 1, otherwise, it can be omitted. The format is identical to that in Data Block No. 10.

CARD 11.1

Enter the surface names in 10-column fields and in the same order as on Card 3.1

Surface Roughness, required if IDAM(11)=1

<u>Card</u>	<u>Symbol</u>	<u>Definition</u>
11.2	DSUR(I,1)	Roughness factor on surface I <u>before</u> damage is incurred.
11.3	(I,2)	Roughness factor on surface I <u>after</u> damage is incurred.
11.4	(I,3)	x/c where damage roughness starts.
11.5	(I,4)	x/c where damage roughness ends.
11.6	(I,5)	Fraction of area affected by increased roughness (includes area forward and aft of damaged region).
11.7	↓ (I,6)	0.0 for Lower Surface. 1.0 for Upper Surface.

Table 1 is presented as an aid in selecting the roughness factors after damage has been incurred.

Fwd-facing steps, required if IDAM(12)=1.

<u>Card</u>	<u>Symbol</u>	<u>Definition</u>
11.8	DSUR(I,11)	Number of fwd-facing steps on lower surface.
11.9	(I,12)	Number of fwd-facing steps on upper surface.
11.10	(I,13)	Width of each step.
11.11	(I,14)	Height of each step.
11.12	(I,15)	x/c at first step.
11.13	↓ (I,16)	x/c at last step.

Aft-facing steps, required if IDAM(13)=1.

<u>Card</u>	<u>Symbol</u>	<u>Definition</u>
11.14	DSUR(I,21)	Number of aft-facing steps on lower surface.
11.15	(I,22)	Number of aft-facing steps on upper surface.
11.16	(I,23)	Width of each step.
11.17	(I,24)	Height of each step.
11.18	(I,25)	x/c at first step.
11.19	↓ (I,26)	x/c at last step.

Holes, required if IDAM(14)=1.

<u>Card</u>	<u>Symbol</u>	<u>Definition</u>
11.20	DSUR(I,31)	Number of holes in lower surface.
11.21	(I,32)	Number of holes in upper surface.
11.22	(I,33)	Length of each hole.
11.23	(I,34)	Width of each hole.
11.24	(I,35)	Depth of each hole.
11.25	(I,36)	x/c where first hole starts.
11.26	(I,37)	x/c where last hole starts.
11.27	(I,38)	Type Hole    1.0 Missing Panel 2.0 Caved-in Panel
11.28	↓ (I,39)	Porosity Factor (must have holes on both upper & lower surface)

Waviness, required if IDAM(15)=1.

<u>Card</u>	<u>Symbol</u>	<u>Definition</u>
11.29	DSUR(I,41)	Number of waves in lower surface.
11.30	(I,42)	Number of waves in upper surface.
11.31	(I,43)	Length of each wave.
11.32	↓ (I,44)	Width of each wave.

<u>Card</u>	<u>Symbol</u>	<u>Denifition</u>
11.33	DSUR(I,45)	Amplitude of each wave.
11.34	↓ (I,46)	x/c where first wave starts.
11.35	↓ (I,47)	x/c where last wave starts.

Protuberances, required if IDAM(16)=1

<u>Card</u>	<u>Symbol</u>	<u>Definition</u>
11.36	DSUR(I,51)	Number of protuberances on Surface I.
11.37	↓ (I,52)	Height of each protuberance.
11.38	↓ (I,53)	Parasite area of each protuberance, $\Delta f$ .
11.39	↓ (I,54)	x/c of first protuberance.
11.40	↓ (I,55)	x/c of last protuberance.

Missing Wing Parts, required if IDAM(17)=1

CARD 11.41

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	DWING(1)	$\Delta c/c$ fraction of wing chord that is missing from leading edge.
11-20	↓ (2)	$\eta_i$ , inboard edge of missing leading edge, expressed as a fraction of semispan.
21-30	↓ (3)	$\Delta \eta$ , span of missing leading edge, expressed as a fraction of semispan.
31-40	↓ (4)	$z/t$ ratio of leading edge to maximum thickness. $z$ represents the wing thickness at the point where the missing leading-edge section is detached. $t$ is the wing maximum thickness.

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
41-50	DWING(5)	Corner sharpness factor for leading edge. Enter 1.0 if corners on leading edge are rounded. Enter 2.0 if corners on leading edge are sharp.
51-60	↓ (6)	
		Indicates that leading edge is missing on:
		1.0, left side only
		2.0, right side only
		3.0, both sides

CARD 11.42

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	DWING(7)	$\Delta c/c$ , fraction of wing chord that is missing from trailing edge.
11-20	↓ (8)	$\eta_i$ , inboard edge of missing trailing edge expressed as a fraction of semispan.
21-30	↓ (9)	$\Delta \eta$ , span of missing trailing edge, expressed as a fraction of semispan.
31-40	↓ (10)	Indicates that trailing edge is missing on:
		1.0, left side only
		2.0, right side only
		3.0, both sides

CARD 11.43

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	DWING(11)	$\Delta \eta$ , fraction of semispan that is missing from the left wing tip.
11-20	↓ (12)	$\Delta \eta$ , right wing tip.
21-30	↓ (13)	Moment arm of device that trims rolling moment. Entered as a fraction of wing semispan.



CARD 11.44

<u>Column</u>	<u>Symbol</u>	<u>Definition</u>
1-10	DWING(14)	The first NHT surface requires input for the left side (Column 1-10) and the right side (Column 11-20). Remaining surfaces must be in the same order as those on Card 3.1. If NHT=0, list NVT surfaces in order beginning with Columns 1-10.
11-20	(15)	
21-30	(16)	
31-40	(17)	
41-50	(18)	
51-60	(19)	
61-70	Y (20)	

There are several points that should be noted regarding the surface damage parameters:

1. All input parameters describe damage to only one surface (such as the right wing). If the damage is symmetric (left wing also damaged), then a negative sign should be used on the first parameter in each mode of damage. The first parameter will be zero in some cases. When this happens, a negative sign on the second parameter will indicate symmetric damage.

2. Porosity factor specifies the fraction of hole area that allows air to flow from the lower to the upper surface. The hole area is the smaller of the area of upper surface holes and lower surface holes.

3. If IDAM(17)=1, Cards 11.42 through 11.45 must all be entered. A zero or blank should be entered for the parameters that are not applicable to the damage case being described.

4. The parasite area for a protuberance is the drag divided by the dynamic pressure ( $D/q$ ) in freestream air flow. Effects of local boundary layer are computed. Units for  $\Delta f$  must be consistent with those selected on Card 1.3.

### Data Block No. 12, End of Problem

Several options are available when the end of the initial problem is attained. The user may either end the problem, add more damage cases, or change the basic input for an additional run.

#### CARD 12.1

- Option 1. Input "END" in columns 1 through 3 and the problem will be terminated.
- Option 2. Input "DAMAGE CASES FOLLOW" beginning in Column 1 and continue input with Card 9.2 to run additional damage cases.
- Option 3. "CHANGE INPUT" beginning in Column 1 allows the user to change any variable in the basic input (Data Block No. 1 through 7). The variable-location is its position in the input common block. This location is given in Volume II, Section 4 of the AAT Computer Code Final Report. The remaining cards in this block must be input.

#### CARD 12.2

Enter any alphanumeric characters in Columns 1 through 60 to identify the new problem. This title will replace the one originally entered on Card 1.1 when the new problem is run.

#### CARD 12.3

<u>Column</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
4-5	I2	M	Number of integer variables to be changed.
9-10	I2	N	Number of floating point variables to be changed.

#### CARD 12.4 (Req'd if M > 0)

<u>Column</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
1-5	I5	M0(1)	Location of 1st integer variable.
6-10	I5	IA(1)	New value of 1st integer variable.

<u>Column</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
11-15	I5	MO(2)	Location of 2nd integer variable.
16-20	I5	IA(2)	New value of 2nd integer variable.
21-25	I5	MO(3)	Location of 3rd integer variable.
26-30	I5	IA(3)	New value of 3rd integer variable.

This pattern is repeated for 6 variables per card, until M pairs of variables have been loaded.

CARD 12.5 - (Req'd if N > 0)

<u>Column</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
1-5	I5	NO(1)	Location of 1st floating point variable.
11-20	F10	AA(1)	New value of 1st floating point variable.
21-25	I5	NO(2)	Location of 2nd floating point variable.
31-40	F10	AA(2)	New value of 2nd floating point variable.
41-45	I5	NO(3)	Location of 3rd floating point variable.
51-60	F10	AA(3)	New value of 3rd floating point variable.

Card 12.5 is repeated until N pairs of variables have been loaded.

CARD 12.6

Input "END" in columns 1 through 3 to terminate problem or omit this card and go to Card 12.1 to run additional problems.

### 3. SAMPLE PROBLEMS

This section contains the computer printout for the following sample problems:

1. F-16A
2. FB-111
3. C-141 with damage

The output is self-explanatory, except that in the C-141 problem it should be noted that the damage imposed is strictly hypothetical and is not an attempt to simulate an actual damage case. Damage inputs are entered for each of the fourteen AAT damage modes to demonstrate the flexibility of the program.

The F-16A and FB-111 problems presented in this section provided the estimates that are compared with data in Section 9, Volume I of this report.



EMPIRICALLY BASED COEFFICIENT PROGRAM  
TO PREDICT THE AERODYNAMIC CHARACTERISTICS  
OF AIRCRAFT  
INCLUDING  
ROUGHNESS, PROTRUSIONS, AND  
CORRECTION CALCULATIONS

F-16 WIND TUNNEL MODEL WITH TIP MISSILES AND LAUNCHERS REMOVED

FUSELAGE INPUT PARAMETERS

NPCCS = 3  
 NPALS = 1  
 NHT = 1  
 NVT = 2  
 IGMP = 0  
 TREF = 0  
 IMAG = 0  
 PPEC = 0  
 PETER = 0  
 ITRIM = 0

SFEE = 305.00  
 CPAC = 11.317  
 XPAC = 22.760  
 I:CG = 0.000  
 UPFUS = 0.000  
 ACE = 0.000

ROUGHN = 2.00000  
 FMI:CC = 0.00  
 TWIST = -3.000  
 CONCL = 0.000

DIMENSIONS ARE IN FEET

FUSELAGE CAPCY NACELLE

LENGTH = 42.070 11.675 23.557  
 WIDTH = 4.833 1.575 4.500  
 HEIGHT = 3.333 1.575 2.167  
 METEC AREA = 532.000 25.120 237.420  
 INTER.FACTCF = 1.000 1.200 1.000  
 NO.CF THIS TYPE = 1.000 1.000 1.000  
 PFXTRUP AREA = 15.972 2.517 0.750  
 BASE AREA = 8.056 0.000 1.517  
 NOSE LENGTH = 16.750 3.500 5.250  
 SCATTAIL LENGTH = 7.917 5.333 20.417  
 BASE DRAG AREA = 0.000 0.000 0.000  
 INLET AREA = 0.000 0.000 5.741

SECTION	WING	PCRIZ.	VERT.	VENTRAL
=	64A	BIC	BIC	BIC
CAMEE (2-C)	= .210	0.000	0.000	0.000
THICKNESS RATIO	= .046	.053	.045	.019
L.E. SWEET	= 40.000	40.000	47.500	45.000
WETTED AREA	= 344.850	58.000	117.000	32.200
RCOT CP-CRC	= 14.545	6.623	9.052	4.702
TIP CP-CRC	= 3.702	2.004	3.556	3.515
SFAP OF PANEL	= 12.917	5.642	0.417	1.618
X- LE RCOT CP-CRC	= 15.534	26.326	24.062	32.275
Y- LE RCOT CP-CRC	= 2.023	2.452	0.800	1.417
Z- LE RCOT CP-CRC	= 0.010	0.000	2.125	-2.133
INCIDENCE	= 0.000	0.000	0.000	0.000



\* CRAG SOLARS TO BE GENERATED BY THE FOLLOWING CONDITIONS  
 PACH NO. ALTITUDE C.C. SWEEP FRCT CL TO CL

.85	-.167	.25	40.0	0.00	1.00
.80	-.167	.25	40.0	0.00	1.00
1.28	-.167	.25	40.0	0.00	1.00
1.63	-.123	.25	40.0	0.00	1.00

NO ADJUSTMENTS \*\*\*\*\*

CALCULATED BODY PARAMETERS  
 FLSE/LACE CANCFY NACE/LE

FINENESS RATIO = 10.482 5.511 10.258  
 WETTED AREA = 532.000 25.120 231.420  
 MAXIMUM AREA = 15.972 2.917 6.750

CALCULATED SURFACE PARAMETERS

	WING	PCRF12.	VEFT.	VENTRAL
WETTED AREA	* 344.890	56.000	117.020	22.200
TAPER RATIO	= .255	.310	.427	.747
PLANFORM AREA	= 225.690	45.012	54.753	13.211
ASPECT RATIO	= 2.022	2.550	2.500	.787
CHARACT. LENGTH	= 10.150	4.764	6.020	4.142
S/P C/4	= 32.180	22.284	43.225	25.224
S/P C/2	= 22.753	22.558	30.263	22.216
S/P TE	= -0.017	.561	25.525	14.250
S/P MAX T/C	= 27.100	22.558	30.263	22.216
X/C AT MAX T/C	= .290	.500	.500	.500

TOTAL AIRPLANE PARAMETERS

WING SEMI-SPAN	*RCS =	15.000
EDLY CIA/ WING SPAN	*LCE =	.129
WING FCCT CHFC AT CENTERLINE	*CF =	16.000
THEO. WING PLANFORM AREA	*SPLAN =	233.924
Y-STA. CF RCCT CHFC (CR) T.E.	*XCFTE =	34.075
EGLIV. WING ASPECT RATIO	*AR =	3.001
EGLIV. WING TAPER RATIO	*TAPR =	.227
EGLIV. WING LEADING EDGE SWEEP	*SALE =	40.000
EGLIV. WING QUARTER-CHORD SWEEP	*SAQCC =	32.100
EGLIV. WING PIC-CHORD SWEEP	*SAPMC =	22.707
EGLIV. WING TAILING-EDGE SWEEP	*SATE =	-0.017
EGLIV. WING MAX. THICKNESS SWEEP	*SMT =	27.100
EGLIV. WING EXPOSED MAC	*SEMA =	235.000
EGLIV. WING THICKNESS RATIO	*CCR =	10.100
EGLIV. WING 2-C DESIGN CL	*CLC =	.040
CHFC LENGTH AT WING FCCT	*ARX =	.200
Y-STA. AT L.E. CF WING FCCT	*CR =	14.505
Y-STA. AT L.E. CF WING TIF	*CY =	3.702
Y-STA. AT L.E. CF WING TIF	*XR =	35.524
Y-STA. AT L.E. CF WING TIF	*XT =	30.273
Y-STA. AT L.E. CF WING TIF	*YR =	2.003
Y-STA. AT L.E. CF WING TIF	*YT =	15.000
Y-STA. AT L.E. CF WING PAC	*XR =	22.708
DELTA-X C/4 WING TO C/4 TAIL	*YE =	5.526
INCLINATION LINE BET. WING AND TAIL	*YF =	18.000
ENTER SURVEY	*CIEGA =	0.000

6A SERIES AIRCRAFT SECTION

PACF CRITICAL TABLE

CL	PACF CRITICAL
0.0000	.0024
.1000	.0522
.2000	.0794
.3000	.0864
.4000	.0825
.5000	.0750
.6000	.0651
.7000	.0554
.8000	.0441
.9000	.0302
1.0000	.0142

F-16 WING TUNNEL MODEL WITH TIF MISSILES AND LALICHERS FENCY  
 PACH NO. = .85C RN/FT = 1.6667E+05 L.E. SHEEP ANGLE = 4.00 DEG. TRIMMED CONNECTION  
 CL AT CP=C CC AT CP=C

CL	TOTAL CC	CF	NLFPA	CC LIFT	CD RAFT	CL AT CP=C	CC AT CP=C
-.019	.02160	-1.75130	-.58	.0073	.00130	4.66000	.02320
.024	.02028	-1.47213	.11	.0005	.00030	.66000	.02661
.087	.02053	-1.19315	.70	.00018	.00000	.16000	.02074
.140	.02180	-.51426	3.48	.00112	.00000	.15000	.02376
.153	.02352	-.03322	2.17	.00290	.00000	.20000	.02447
.246	.02607	-.35228	2.66	.00548	.00000	.25000	.02405
.259	.02546	-.07651	3.54	.00155	.00000	.30000	.02546
.405	.02745	.20342	4.23	.00312	.00000	.35000	.02368
.458	.04472	.48307	4.52	.00816	.00000	.40000	.02872
.512	.05160	.76417	5.62	.02482	.00000	.45000	.04482
.565	.06054	1.04725	6.45	.03270	.00000	.50000	.05166
.617	.07579	1.32250	7.42	.03858	.00000	.55000	.06075
.668	.09088	1.52566	8.54	.05251	.00000	.60000	.07584
.716	.10568	1.68221	9.75	.07104	.00000	.65000	.09088
.807	.12377	1.46526	11.19	.09454	.00000	.70000	.12354
.885	.15829	1.25354	13.72	.12332	.00000	.75000	.15826
.947	.19570	.00000	14.42	.15648	.00000	.80000	.19568
.985	.23074	-.22133	16.24	.19491	.00000	.85000	.23087
.923	.27260	-1.27187	18.21	.23832	.00000	.90000	.27074
.972	.44150	-2.43516	21.32	.28171	.00000	.95000	.37176
		-2.51054	21.57	.34007	.00000	1.00000	.44052

CLA = .07265 PER DEG. ALC = -.57503 DEG.  
 K = .16362 CACL = .06894  
 CND = -.02817\* CM/CL = .07630\* (\*TAIL CFF)

FRAG BREAKDOWN	FUSELAGE	CANOPY	NACELLE	VENTRAL
FRICITION = .0021	.00231	.00044	.00246	.00049
FORM = .0021	.00042	.00023	.00008	.00001
INTERF = .0021	.00022	.00017	.00000	.00015
WAVE = .0000	.00000	.00000	.00000	.00000
BASE = .00000	.00000	.00000	.00000	.00000
FRICITION = .00792	.00429	.00144	.00160	.00049
FORM = .0046	.00027	.00009	.00000	.00001
INTERF = .0020	.00100	.00048	.00053	.00015
WAVE = .00000	.00000	.00000	.00000	.00000
CANOPER = .00135	.00000	.00000	.00000	.00000
FRAG RISE = .0000	.00000	.00000	.00000	.00000
PISC = .000000	.00000	.00000	.00000	.00000
CEMIN = .02056				

F-16 WING TUNNEL MODEL WITH TIF MISSILES AND LAUNCHERS FEMCA  
 PACH NO. = .9CC RN/PI = 1.667E+05 U.S. SNEEP ANGLE = 41.6( DEG).

CL	TOTAL CD	CF	ALPHA	CD LIFT	CE MAFY	CL AT CF=0	CC AT CF=C
-.022	.02158	-1.92064	-.58	.00067	0.00000	0.00000	.02152
.032	.02125	-1.80575	.02	.00000	.00001	.05000	.02451
.085	.02123	-1.25534	.74	.00020	.00012	.10000	.02418
.129	.02260	-.58860	1.40	.00117	.00041	.15000	.02444
.192	.02475	-.07007	2.06	.00295	.00085	.20000	.02466
.246	.02751	-.32744	2.72	.00552	.00148	.25000	.02487
.299	.03205	-.05661	3.38	.00891	.00228	.30000	.02504
.353	.03726	.25451	4.04	.01305	.00322	.35000	.02527
.406	.04369	.56654	4.70	.01807	.00456	.40000	.02545
.460	.05123	.87537	5.37	.02381	.00632	.45000	.02554
.514	.05970	1.20255	6.06	.03045	.00837	.50000	.02560
.568	.07001	1.54151	7.08	.03785	.01125	.55000	.02555
.621	.08253	1.88462	8.12	.04601	.01456	.60000	.02542
.673	.10674	2.23152	9.22	.05476	.01820	.65000	.02522
.726	.13216	2.67421	10.63	.06371	.02210	.70000	.02496
.772	.15840	3.19470	12.09	.07281	.02623	.75000	.02464
.818	.18505	3.69561	13.67	.08200	.03066	.80000	.02424
.862	.21221	4.17220	15.20	.09120	.03541	.85000	.02372
.902	.24027	4.62611	17.23	.10040	.04045	.90000	.02322
.938	.26960	-1.02551	19.21	.10960	.04564	.95000	.02260
.991	.46771	-.80447	26.32	.34177	.10452	1.00000	.02115

CLA = .07574 PER DEG. ALC = -.58392 DEG.  
 CLK = .16054 DELCL = .06447  
 CPO = -.02782\* CTA/CL = .07442\*

CLCB = .55211  
 CLMFX = .95522

DFAG BREAKDOWN -----  
 FRICTION = .00217  
 FCRP = .00062  
 INTERF = .00021  
 WAVE = 0.00000  
 BASE = 0.00000  
 FRICTION = .00708  
 FCRP = .00046  
 INTERF = .00262  
 WAVE = 0.00000  
 CAMBER = .00130  
 DFAG RISE = 0.00000  
 PISC = 0.00000  
 CCFIA = .02086

FUSELAGE  
 .00020  
 .00041  
 .00028  
 0.00000  
 0.00000  
 WING  
 .00437  
 .00027  
 .00122  
 0.00000  
 CANOPY  
 .00002  
 .00012  
 .00017  
 0.00000  
 0.00000  
 FORTIZ  
 .00143  
 .00000  
 .00006  
 0.00000  
 NACELLE  
 .00445  
 .00000  
 0.00000  
 0.00000  
 VENTRAL  
 .00160  
 .00000  
 .00062  
 0.00000

F-16 NAC TUNNEL MODEL WITH TIF MISSILES AND LAUNCHERS REMOVED  
 PACH AC. = 1.2222 RN/FT = 1.6667E+05 L.E. SWEEP ANGLE = 40.00 DEG. TRIMMED CONCUSSION

CL	TOTAL CC	CF	ALPHA	CC LIFT	CC R'AFT	CL AT CP=C	CC AT CP=C
0.000	.03978	0.0000	0.00	.00000	0.00000	0.00000	.03978
.047	.03962	-.23427	.05	.00000	0.00000	.05000	.02561
.095	.04005	-.46266	1.20	.00161	0.00000	.10000	.04005
.142	.04248	-.72228	1.55	.00464	0.00000	.15000	.04227
.190	.04709	-.99227	2.01	.00756	0.00000	.20000	.04650
.237	.05162	-1.17273	2.28	.01225	0.00000	.25000	.05152
.285	.05718	-1.40979	2.91	.01791	0.00000	.30000	.05725
.332	.06486	-1.60657	4.56	.02472	0.00000	.35000	.06457
.379	.07216	-1.86420	5.21	.03265	0.00000	.40000	.07155
.427	.08159	-2.18279	5.86	.04167	0.00000	.45000	.08161
.474	.09224	-2.56247	6.51	.05179	0.00000	.50000	.09112
.522	.10311	-2.80226	7.17	.06200	0.00000	.55000	.10234
.569	.11641	-2.94559	7.82	.07221	0.00000	.60000	.11465
.617	.12071	-3.06930	6.47	.08230	0.00000	.65000	.12162
.664	.12747	-3.23460	5.12	.10992	0.00000	.70000	.14516
.711	.16572	-3.50163	5.77	.12260	0.00000	.75000	.16293
.759	.18514	-3.82052	10.42	.14262	0.00000	.80000	.18155
.806	.20544	-4.00141	11.07	.16269	0.00000	.85000	.20243
.853	.22782	-4.23444	11.72	.18442	0.00000	.90000	.22272
.900	.25110	-4.50975	12.20	.21115	0.00000	.95000	.24653
.948	.27583	-4.824747	12.03	.23122	0.00000	1.00000	.27255

CLA = .07676 PER DEG. ALC = 0.00000 DEG. CLCB = .63496  
 K = .21964 DELCL = .01462 CLPX = 1.2825E  
 CMC = 0.000000\* CAYCL = -.05207\* (9\*TAIL CFF)

FRAG BREAKDOWN	FLSEFACE	CANOPY	NACELLE	VENTRAL
FRICITION = .00787	.00000	.00042	.00236	.00047
FCP = 0.00000	0.00000	0.00000	0.00000	.00000
INTEFF = 0.00000	0.00000	0.00000	0.00000	.00000
PAVE = .01282	.00720	.00250	.00212	.00000
FASE = 0.00000	0.00000	0.00000	0.00000	.00005
FRICITION = .00762	.00422	.00138	.00154	.00047
FCP = 0.00000	0.00000	0.00000	0.00000	.00000
INTEFF = 0.00000	0.00000	0.00000	0.00000	.00000
PAVE = .01099	.00359	.00121	.00123	.00005
CANOPY = .00004				
IFAG RISE = 0.00000				
MISC = 0.00000				
CDMIN = .00222				





FD-311

2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
551.0	0.0			40.1											
C-00173	15.0			-5.6											
551.0	0.05			15.0											
FUSELAGE	FACELE														
70.8	37.5														
11.07	3.05														
5.5	2.55														
1020.0	442.0														
1.2	2.5														
1.0	2.0														
27.15	9.55														
15.97	7.47														
26.42	7.58														
22.75	28.08														
0.0	2.6														
	2.51														
ELCIVE	VARIABLE														
64	64														
0.0	0.2														
0.05	0.10														
70.0	16.0														
21.0	60.0														
25.58	10.50														
10.00	3.13														
5.5667	20.5333														
23.33	28.42														
2.55	0.0667														
0.42	0.0														
1.0	1.0														
40.64	5.86														
	21														
0.0	20000.0														
0.0	20000.0														
0.0	20000.0														
0.0	20000.0														
1.2	20000.0														
1.0	20000.0														
2.0	20000.0														
NO ADJUSTMENTS															
ENC															

SPECIFICALLY BASED COMPUTER PROGRAM  
TO PREDICT THE AERODYNAMIC CHARACTERISTICS  
OF AIRCRAFT  
INCLUDING  
ROLL-RATE, PITCH-RATE, AND  
LARGE CALCULATIONS

FE-111

PROCELP INPUT PARAMETERS

APCDS = 2  
 NFALS = 2  
 PHT = 1  
 NVT = 2  
 ISAP = 1  
 IREF = 0  
 INAG = 1  
 NAFK = 0  
 PETER = 0  
 ITRIP = 0  
  
 SREF = 550.00  
 CFAC = 0.000  
 YFAC = 40.000  
 ZCC = 0.000  
 UFRUS = 0.000  
 LICE = 0.000

ROUGHK = 0.0072  
 FRISC = 15.00  
 THIST = -5.000  
 CONCL = 0.000

DIMENSIONS ARE IN FEET

REFERENCE WING FOR TWO PANEL CASE

SPLAN = 550.000    TAPER = 0.2500    SARP = 10.000

FUSELAGE NOSE

LENGTH = 70.800    33.500  
 WIDTH = 11.370    3.550  
 HEIGHT = 5.900    3.550  
 NETTED AREA = 1020.000    443.000  
 INTER.FACTOR = 1.200    1.500  
 A.C. OF THIS TYPE = 1.000    2.000  
 MAXIMUM AREA = 37.150    5.550  
 BASE AREA = 15.970    7.470  
 NOSE LENGTH = 26.420    3.500  
 BOATTAIL LENGTH = 22.750    28.000  
 BASE CFAC AREA = 1.000    2.000  
 INLET AREA = 0.000    3.510

GLOVE VAR.ING P.T. V.T. STPANE

SECTION	=	64A	64A	61C	61C	61C
CAPEEF (2-D)	=	8.000	.200	0.000	0.000	0.000
THICKNESS RATIC	=	.055	.100	.025	.032	.020
L.E. SHEEP	=	70.000	10.000	27.500	55.000	40.000
WETTED AREA	=	313.000	800.000	334.000	223.000	50.000
RCOT CHCRE	=	25.000	10.000	15.000	17.790	14.000
TIF CFCFC	=	10.500	2.100	2.400	7.210	13.250
SFAP CF PANEL	=	5.000	20.000	0.000	0.000	1.440
X- LE RCOT CFCFC	=	23.000	10.400	26.100	53.000	50.000
Y- LE RCOT CFCFC	=	2.000	0.000	5.000	0.000	4.420
Z- LE FCOT CFCFC	=	.400	0.000	0.000	1.000	-4.170
INADIENCE	=	1.000	1.000	1.000	0.000	0.000

2 CRAG FOLARS TO BE GENERATED AT THE FOLLOWING CONDITIONS

MACH NO.	ALTITUDE	C.G.	SKEW	FRCP CL	TO CL
.20	3000.000	.45	16.0	C.CC	1.00
.20	3000.000	.45	26.0	C.CC	1.00
.20	3000.000	.45	35.0	C.CC	1.00
.20	3000.000	.45	50.0	C.CC	1.00
.20	3000.000	.45	72.5	C.CC	1.00
1.20	3000.000	.45	72.5	C.CC	1.00
1.60	3000.000	.45	72.5	C.CC	1.00
2.00	3000.000	.45	72.5	C.CC	1.00

NO ADJUSTMENTS \*\*\*\*\*

CALCULATED ECCY PARAMETERS  
FUSELAGE NACELLE

FINENESS RATIO = 2.644 9.437  
 WETTED AREA = 1220.000 443.000  
 MAXIMUM AREA = 37.150 5.550

CALCULATED SURFACE PARAMETERS

	GLOVE	VAR.	PIPING	P.T.	S.T.	STRAKE
WETTED AREA	= 313.000	800.000	334.000	223.000	50.000	50.000
T/REF RATIO	= .414	.256	.185	.411	.506	.506
PLANFORM AREA	= 201.292	369.256	156.670	111.695	40.205	40.205
ASPECT RATIO	= .616	7.658	2.055	1.418	.206	.206
CLIFFACT. LENGTH	= 15.112	7.520	10.240	13.280	13.671	13.671
S/P C/4	= 64.257	12.276	50.672	40.587	21.235	21.235
S/P C/2	= 54.466	8.442	41.071	40.610	20.457	20.457
S/P TE	= 2.027	.521	9.445	14.070	5.622	5.622
S/P MAX T/C	= 55.484	11.141	41.071	40.610	20.457	20.457
X/C AT MAX T/C	= .390	.350	.500	.500	.500	.500

TOTAL AIRPLANE PARAMETERS

WING SEMI-SPAN	'SECS	=	35.000
FCOY CIA/ WING SPAN	'LCE	=	.071
WING FOOT CHCC AT CENTERLINE	'CR	=	32.716
WING PLANFORM AREA	'SFLAN	=	20.000
Y-STA. CF ROOT CHCC (CFI) T.E.	'XCRTE	=	20.070
ECLIV. WING ASPECT RATIO	'AR	=	6.809
ECLIV. WING TAPER RATIO	'TAPR	=	.250
ECLIV. WING LEADING EDGE SWEEP	'SLE	=	49.112
ECLIV. WING CLARTEP-CHCCD SWEEP	'SFLCC	=	20.221
ECLIV. WING PIC-CLCCD SWEEP	'SFRMC	=	22.204
ECLIV. WING TRAILING-EDGE SWEEP	'SAPTE	=	1.445
ECLIV. WING MAX THICKNESS SWEEP	'SEDT	=	21.251
ECLIV. WING EXPOSED MAC	'CSAF	=	570.547
ECLIV. WING EXPOSED MAC	'TCCA	=	11.618
ECLIV. WING THICKNESS RATIO	'CLC	=	.092
ECLIV. WING 2-C DESIGN CL	'ARX	=	.161
ECLIV. WING EXPOSED ASPECT RATIO	'ARX	=	7.405
CHCCD LENGTH AT WING FOOT	'CFY	=	25.500
Y-STA. AT L.E. CF WING FOOT	'XR	=	2.120
Y-STA. AT L.E. CF WING TIP	'XT	=	2.805
Y-STA. AT L.E. CF WING FOOT	'YR	=	46.143
Y-STA. AT L.E. CF WING TIP	'YT	=	2.500
Y-STA. AT L.E. CF WING MAC	'YB	=	25.000
Y-STA. AT L.E. CF WING MAC	'YB	=	21.250
CELTA-X G/4 WING TO C/4 TAIL	'XP	=	14.000
INCLINATION LINE RET. WING AND TAIL	'XPECA	=	23.992
ENTER SURVEY			0.000

GA SERIES AIRCRAFT SECTION

MACH CRITICAL TABLE

CL	MACH CRITICAL
0.1000	.8199
.1000	.7982
.2000	.7850
.3000	.7726
.4000	.7600
.5000	.7472
.6000	.7324
.7000	.7161
.8000	.6974
.9000	.6759
1.0000	.6521



FR-111  
 PACH NO. = .80C ALTITUDE = 30000. FT. L.O. SWEEP ANGLE = 16.00 DEG. TRIANGLE CONDITION

CL	TOTAL CL	CH	ALPHA	CC LIFF	CC RAFT	CL AT CH = C	CC AT CP = C
.016	.02274	.02202	.25	.00000	.00000	.00000	.02251
.016	.02274	.03001	.75	.00000	.00000	.00000	.02251
.116	.02274	.05613	1.14	.00000	.00000	.00000	.02251
.167	.02274	.07244	1.52	.00000	.00000	.00000	.02251
.217	.02274	.08875	1.92	.00000	.00000	.00000	.02251
.267	.02274	.10506	2.32	.00000	.00000	.00000	.02251
.318	.02274	.12137	2.71	.00000	.00000	.00000	.02251
.368	.02274	.13768	3.10	.00000	.00000	.00000	.02251
.419	.02274	.15400	3.50	.00000	.00000	.00000	.02251
.469	.02274	.17031	3.89	.00000	.00000	.00000	.02251
.519	.02274	.18662	4.28	.00000	.00000	.00000	.02251
.569	.02274	.20293	4.67	.00000	.00000	.00000	.02251
.620	.02274	.21924	5.07	.00000	.00000	.00000	.02251
.670	.02274	.23555	5.46	.00000	.00000	.00000	.02251
.720	.02274	.25186	5.85	.00000	.00000	.00000	.02251
.771	.02274	.26817	6.25	.00000	.00000	.00000	.02251
.821	.02274	.28448	6.64	.00000	.00000	.00000	.02251
.872	.02274	.30079	7.02	.00000	.00000	.00000	.02251
.922	.02274	.31710	7.42	.00000	.00000	.00000	.02251
.972	.02274	.33341	7.82	.00000	.00000	.00000	.02251
		.34972	8.22	.00000	.00000	.00000	.02251
		.36603	8.62	.00000	.00000	.00000	.02251
		.38234	9.02	.00000	.00000	.00000	.02251
		.39865	9.42	.00000	.00000	.00000	.02251
		.41496	9.82	.00000	.00000	.00000	.02251
		.43127	10.22	.00000	.00000	.00000	.02251
		.44758	10.62	.00000	.00000	.00000	.02251
		.46389	11.02	.00000	.00000	.00000	.02251
		.48020	11.42	.00000	.00000	.00000	.02251
		.49651	11.82	.00000	.00000	.00000	.02251
		.51282	12.22	.00000	.00000	.00000	.02251
		.52913	12.62	.00000	.00000	.00000	.02251
		.54544	13.02	.00000	.00000	.00000	.02251
		.56175	13.42	.00000	.00000	.00000	.02251
		.57806	13.82	.00000	.00000	.00000	.02251
		.59437	14.22	.00000	.00000	.00000	.02251
		.61068	14.62	.00000	.00000	.00000	.02251
		.62699	15.02	.00000	.00000	.00000	.02251
		.64330	15.42	.00000	.00000	.00000	.02251
		.65961	15.82	.00000	.00000	.00000	.02251
		.67592	16.22	.00000	.00000	.00000	.02251
		.69223	16.62	.00000	.00000	.00000	.02251
		.70854	17.02	.00000	.00000	.00000	.02251
		.72485	17.42	.00000	.00000	.00000	.02251
		.74116	17.82	.00000	.00000	.00000	.02251
		.75747	18.22	.00000	.00000	.00000	.02251
		.77378	18.62	.00000	.00000	.00000	.02251
		.79009	19.02	.00000	.00000	.00000	.02251
		.80640	19.42	.00000	.00000	.00000	.02251
		.82271	19.82	.00000	.00000	.00000	.02251
		.83902	20.22	.00000	.00000	.00000	.02251
		.85533	20.62	.00000	.00000	.00000	.02251
		.87164	21.02	.00000	.00000	.00000	.02251
		.88795	21.42	.00000	.00000	.00000	.02251
		.90426	21.82	.00000	.00000	.00000	.02251
		.92057	22.22	.00000	.00000	.00000	.02251
		.93688	22.62	.00000	.00000	.00000	.02251
		.95319	23.02	.00000	.00000	.00000	.02251
		.96950	23.42	.00000	.00000	.00000	.02251
		.98581	23.82	.00000	.00000	.00000	.02251
		1.00212	24.22	.00000	.00000	.00000	.02251
		1.01843	24.62	.00000	.00000	.00000	.02251
		1.03474	25.02	.00000	.00000	.00000	.02251
		1.05105	25.42	.00000	.00000	.00000	.02251
		1.06736	25.82	.00000	.00000	.00000	.02251
		1.08367	26.22	.00000	.00000	.00000	.02251
		1.10000	26.62	.00000	.00000	.00000	.02251
		1.11631	27.02	.00000	.00000	.00000	.02251
		1.13262	27.42	.00000	.00000	.00000	.02251
		1.14893	27.82	.00000	.00000	.00000	.02251
		1.16524	28.22	.00000	.00000	.00000	.02251
		1.18155	28.62	.00000	.00000	.00000	.02251
		1.19786	29.02	.00000	.00000	.00000	.02251
		1.21417	29.42	.00000	.00000	.00000	.02251
		1.23048	29.82	.00000	.00000	.00000	.02251
		1.24679	30.22	.00000	.00000	.00000	.02251
		1.26310	30.62	.00000	.00000	.00000	.02251
		1.27941	31.02	.00000	.00000	.00000	.02251
		1.29572	31.42	.00000	.00000	.00000	.02251
		1.31203	31.82	.00000	.00000	.00000	.02251
		1.32834	32.22	.00000	.00000	.00000	.02251
		1.34465	32.62	.00000	.00000	.00000	.02251
		1.36096	33.02	.00000	.00000	.00000	.02251
		1.37727	33.42	.00000	.00000	.00000	.02251
		1.39358	33.82	.00000	.00000	.00000	.02251
		1.40989	34.22	.00000	.00000	.00000	.02251
		1.42620	34.62	.00000	.00000	.00000	.02251
		1.44251	35.02	.00000	.00000	.00000	.02251
		1.45882	35.42	.00000	.00000	.00000	.02251
		1.47513	35.82	.00000	.00000	.00000	.02251
		1.49144	36.22	.00000	.00000	.00000	.02251
		1.50775	36.62	.00000	.00000	.00000	.02251
		1.52406	37.02	.00000	.00000	.00000	.02251
		1.54037	37.42	.00000	.00000	.00000	.02251
		1.55668	37.82	.00000	.00000	.00000	.02251
		1.57299	38.22	.00000	.00000	.00000	.02251
		1.58930	38.62	.00000	.00000	.00000	.02251
		1.60561	39.02	.00000	.00000	.00000	.02251
		1.62192	39.42	.00000	.00000	.00000	.02251
		1.63823	39.82	.00000	.00000	.00000	.02251
		1.65454	40.22	.00000	.00000	.00000	.02251
		1.67085	40.62	.00000	.00000	.00000	.02251
		1.68716	41.02	.00000	.00000	.00000	.02251
		1.70347	41.42	.00000	.00000	.00000	.02251
		1.71978	41.82	.00000	.00000	.00000	.02251
		1.73609	42.22	.00000	.00000	.00000	.02251
		1.75240	42.62	.00000	.00000	.00000	.02251
		1.76871	43.02	.00000	.00000	.00000	.02251
		1.78502	43.42	.00000	.00000	.00000	.02251
		1.80133	43.82	.00000	.00000	.00000	.02251
		1.81764	44.22	.00000	.00000	.00000	.02251
		1.83395	44.62	.00000	.00000	.00000	.02251
		1.85026	45.02	.00000	.00000	.00000	.02251
		1.86657	45.42	.00000	.00000	.00000	.02251
		1.88288	45.82	.00000	.00000	.00000	.02251
		1.89919	46.22	.00000	.00000	.00000	.02251
		1.91550	46.62	.00000	.00000	.00000	.02251
		1.93181	47.02	.00000	.00000	.00000	.02251
		1.94812	47.42	.00000	.00000	.00000	.02251
		1.96443	47.82	.00000	.00000	.00000	.02251
		1.98074	48.22	.00000	.00000	.00000	.02251
		1.99705	48.62	.00000	.00000	.00000	.02251
		2.01336	49.02	.00000	.00000	.00000	.02251
		2.02967	49.42	.00000	.00000	.00000	.02251
		2.04598	49.82	.00000	.00000	.00000	.02251
		2.06229	50.22	.00000	.00000	.00000	.02251
		2.07860	50.62	.00000	.00000	.00000	.02251
		2.09491	51.02	.00000	.00000	.00000	.02251
		2.11122	51.42	.00000	.00000	.00000	.02251
		2.12753	51.82	.00000	.00000	.00000	.02251
		2.14384	52.22	.00000	.00000	.00000	.02251
		2.16015	52.62	.00000	.00000	.00000	.02251
		2.17646	53.02	.00000	.00000	.00000	.02251
		2.19277	53.42	.00000	.00000	.00000	.02251
		2.20908	53.82	.00000	.00000	.00000	.02251
		2.22539	54.22	.00000	.00000	.00000	.02251
		2.24170	54.62	.00000	.00000	.00000	.02251
		2.25801	55.02	.00000	.00000	.00000	.02251
		2.27432	55.42	.00000	.00000	.00000	.02251
		2.29063	55.82	.00000	.00000	.00000	.02251
		2.30694	56.22	.00000	.00000	.00000	.02251
		2.32325	56.62	.00000	.00000	.00000	.02251
		2.33956	57.02	.00000	.00000	.00000	.02251
		2.35587	57.42	.00000	.00000	.00000	.02251
		2.37218	57.82	.00000	.00000	.00000	.02251
		2.38849	58.22	.00000	.00000	.00000	.02251
		2.40480	58.62	.00000	.00000	.00000	.02251
		2.42111	59.02	.00000	.00000	.00000	.02251
		2.43742	59.42	.00000	.00000	.00000	.02251
		2.45373	59.82	.00000	.00000	.00000	.02251
		2.47004	60.22	.00000	.00000	.00000	.02251
		2.48635	60.62	.00000	.00000	.00000	.02251
		2.50266	61.02	.00000	.00000	.00000	.02251
		2.51897	61.42	.00000	.00000	.00000	.02251
		2.53528	61.82	.00000	.00000	.00000	.02251
		2.55159	62.22	.00000	.00000	.00000	.02251
		2.56790	62.62	.00000	.00000	.00000	.02251
		2.58421	63.02	.00000	.00000	.00000	.02251
		2.60052	63.42	.00000	.00000	.00000	.02251
		2.61683	63.82	.00000	.00000	.00000	.02251
		2.63314	64.22	.00000	.00000	.00000	.02251
		2.64945	64.62	.00000	.00000	.00000	.02251

MACH CRITICAL TABLE

CL	MACH CRITICAL
0.0000	.8160
.1000	.8021
.2000	.7899
.3000	.7778
.4000	.7661
.5000	.7547
.6000	.7439
.7000	.7335
.8000	.7236
.9000	.7141
1.0000	.7050



MACH CRITICAL TABLE

CL	MACH CRITICAL
0.5000	.8220
.1000	.8095
.2000	.7971
.3000	.7847
.4000	.7717
.5000	.7574
.6000	.7409
.7000	.7225
.8000	.7012
.9000	.6762
1.0000	.6450

FR-111

PACT NO. = 800

ALTITUDE = 30000. FT. I.E. SLEEP ANGLE = 35.00 DEG.

TRIMMED CONDITION  
CI AT EP = C  
CC AT EH = C

CL	TOTAL CC	EP	ALPHA	CD LIFT	CC RWAFT	CI AT EP = C	CC AT EH = C
.036	.02184	.32122	.42	.00005	C.CC000	C.CC000	.03175
.049	.05171	-.05155	.42	.00002	C.CC000	C.CC000	.05500
.093	.03178	-.12425	1.23	.00027	C.CC000	C.CC000	.03155
.136	.02207	-.19706	1.79	.00080	C.CC000	C.CC000	.05000
.180	.02257	-.116971	2.54	.00360	C.CC002	C.CC000	.02372
.223	.05241	-.154231	2.69	.00268	C.CC048	C.CC048	.04455
.267	.02461	-.151483	2.15	.00404	C.CC045	C.CC000	.04455
.311	.02618	-.238720	3.60	.00568	C.CC092	C.CC000	.02620
.354	.02012	-.265567	4.05	.00755	C.CC157	C.CC000	.02620
.398	.03054	-.322158	4.51	.00578	C.CC246	C.CC000	.03066
.442	.03725	-.340421	4.56	.00224	C.CC258	C.CC072	.03794
.485	.03678	-.377676	5.41	.00458	C.CC305	C.CC000	.03745
.528	.04068	-.414665	5.87	.00300	C.CC362	C.CC000	.04174
.572	.04522	-.452125	6.22	.00330	C.CC312	C.CC000	.04452
.613	.05519	-.563952	7.64	.00172	C.CC182	C.CC000	.05212
.634	.07129	-.670595	8.85	.04561	C.CC152	C.CC000	.06515
.667	.05558	-.713558	10.10	.00700	C.CC198	C.CC000	.06551
.700	.12555	-.872416	11.56	.00156	C.CC242	C.CC000	.06551
.733	.16277	-.968383	12.28	.001767	C.CC292	C.CC000	.06551
.767	.20676	-.106610	12.47	.00018	C.CC345	C.CC000	.06551
.801	.25667	-.115212	14.54	.02505	C.CC426	C.CC000	.06551

CLA = .01021 PEF DEC. ALC = .42447 DEG.

CM = .05325 DELCL = .05590

CMO = .02825\* CRACL = .01672\*

(\*TAIL OFF)

CLW = .65061

CIFPX = 1.57221

DRAG REPRESENTATION

FFRICTION = .00529

FCRM = .00647

INTERF = .00096

PAVE = .00000

BASE = .00114

GLUCVE = .00133

FRICION = .00782

FGPY = .00350

INTERF = .00225

PAVE = .00000

CPRES = .00015

IFRGRISE = .00000

PISC = .00266

LCMA = .00170

FLSELAGE

.00255

.00041

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

NACELLE

.00172

.00006

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

STRAME

.00181

.00004

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

.00000

6A SERIES AIRFOIL SECTION

PACH CRITICAL TABLE

CL	PACH CRITICAL
0.0000	.6323
.1000	.6566
.2000	.6846
.3000	.7132
.4000	.7415
.5000	.7685
.6000	.7942
.7000	.8186
.8000	.8417
.9000	.8635
1.0000	.8840

FR-111  
 PACH NC. = .800 ALTITUDE = 30000 FT. L.S. SNEEP ANGLE = 50.00 DEG.  
 TRIMMED CONDITION

CL	TOTAL CC	EF	ALPHA	CC LIFT	CD R-AFT	CL AT OP = C	CL AT LP = C
.000	.02102	-10566	.52	.0000	.00000	.00000	.00000
.000	.02102	-10566	1.05	.0000	.00000	.00000	.00000
.000	.02102	-10566	1.58	.0000	.00000	.00000	.00000
.070	.02051	-10500	1.64	.0000	.00000	.00000	.00000
.150	.01907	-10400	2.00	.0000	.00000	.00000	.00000
.157	.01800	-10300	2.76	.0000	.00000	.00000	.00000
.156	.01700	-10200	3.31	.0000	.00000	.00000	.00000
.156	.01600	-10100	3.87	.0000	.00000	.00000	.00000
.205	.01500	-10000	4.42	.0000	.00000	.00000	.00000
.274	.01400	-9900	4.98	.0000	.00000	.00000	.00000
.313	.01300	-9800	5.52	.0000	.00000	.00000	.00000
.322	.01200	-9700	5.82	.0000	.00000	.00000	.00000
.324	.01100	-9600	7.07	.0000	.00000	.00000	.00000
.414	.01000	-9500	8.05	.0000	.00000	.00000	.00000
.446	.00900	-9400	10.18	.0000	.00000	.00000	.00000
.477	.00800	-9300	11.22	.0000	.00000	.00000	.00000
.510	.00700	-9200	12.21	.0000	.00000	.00000	.00000
.543	.00600	-9100	12.22	.0000	.00000	.00000	.00000
.576	.00500	-9000	14.22	.0000	.00000	.00000	.00000
.610	.00400	-8900	15.29	.0000	.00000	.00000	.00000
.644	.00300	-8800	16.24	.0000	.00000	.00000	.00000
.679	.00200	-8700	17.16	.0000	.00000	.00000	.00000
.715	.00100	-8600		.0000	.00000	.00000	.00000

CLP = .0000 PER DEG. ALC = .52360 DEG.  
 K = .07700 DELCL = .02700  
 CMO = .01930 CPTCL = -.16262

(\*TAIL CFF)

FRAG BREAKDOWN	FUSELAGE	WING	STRAKE
FRICION = .00520	.00333	.00358	.00101
FCRM = .00047	.00041	.00107	.00004
INTERF = .00096	.00000	.00000	.00000
WAVE = .00000	.00000	.00000	.00000
EASE = .00114	.00114	.00114	.00000
FRICION = .00740	.00133	.00358	.00101
FCRM = .00095	.00012	.00107	.00004
INTERF = .00225	.00041	.00000	.00000
WAVE = .00000	.00000	.00000	.00000
CAMBER = .00020	.00000	.00000	.00000
CPAC RISE = .00000	.00000	.00000	.00000
PISC = .00256	.00256	.00256	.00000
CCMA = .00097	.00097	.00097	.00000

PACH CRITICAL TABLE

CL	PACH CRITICAL
0.0000	.8659
.1000	.8557
.2000	.8456
.3000	.8352
.4000	.8246
.5000	.8138
.6000	.8028
.7000	.7914
.8000	.7798
.9000	.7679
1.0000	.7557



FB-111  
MACH NO. = .880 ALTITUDE = 3000. FT. L.E. SLEEP ANGLE = 72.50 DEG. TRI-MEC CONDITION

CL	TOTAL CC	CF	ALPHA	CC LIFT	CC RAFT	CL AT CF = C	CC AT CF = C
.010	.01924	-.56710	.72	.0000	.00000	0.80000	.01952
.031	.01969	-1.10695	1.63	.0016	.00000	.05000	.01999
.072	.02035	-1.62210	2.55	.0030	.00000	.10000	.02032
.114	.02125	-2.13471	3.45	.0050	.00000	.15000	.02070
.155	.02240	-2.64042	4.35	.0075	.00000	.20000	.02106
.197	.02383	-3.14672	5.22	.0110	.00000	.25000	.02144
.239	.02554	-3.65362	6.11	.0150	.00000	.30000	.02182
.281	.02752	-4.16111	6.94	.0195	.00000	.35000	.02220
.324	.02978	-4.66915	7.78	.0245	.00000	.40000	.02257
.367	.03232	-5.17777	8.62	.0299	.00000	.45000	.02294
.409	.03514	-5.68692	9.48	.0357	.00000	.50000	.02331
.452	.03825	-6.19662	10.32	.0419	.00000	.55000	.02368
.495	.04167	-6.70689	11.17	.0485	.00000	.60000	.02405
.537	.04541	-7.21777	12.00	.0554	.00000	.65000	.02442
.579	.04947	-7.72927	12.87	.0627	.00000	.70000	.02479
.622	.05385	-8.24141	13.75	.0704	.00000	.75000	.02516
.664	.05855	-8.75422	14.65	.0785	.00000	.80000	.02553
.707	.06357	-9.26771	15.56	.0870	.00000	.85000	.02590
.749	.06891	-9.78187	16.48	.0959	.00000	.90000	.02627
.792	.07457	-10.29670	17.42	.1052	.00000	.95000	.02664
.834	.08065	-10.81221	18.37	.1149	.00000	.00000	.02701

CLIM = .1951E  
CLMAX = 1.0400E

CLM = .05422 REP DEG. ALC = .7161E DEG.  
K = .1919R CELCL = .0200E  
CMO = .00035\* CP/CL = .05540\*

(\*TAIL CFF)

CPAG BREAKDOWN	FUSELAGE	NACELLE	V.T.	STRAKE
FRICION = .0528	.00132	.00073	.00101	.00022
FCRP = .0047	.00041	.00019	.00024	.00001
INTEFF = .0056	.00000	.00000	.00028	.00000
PAVE = .00000	.00000	.00000	.00000	.00000
EASE = .00114	.00000	.00000	.00000	.00000
FRICION = .00560	.00132	.00073	.00101	.00022
FCRM = .0042	.00012	.00019	.00024	.00001
INTEFF = .00202	.00041	.00000	.00028	.00000
PAVE = .00000	.00000	.00000	.00000	.00000
CAMBER = .00026	.00000	.00000	.00000	.00000
CPAG RISE = .00000	.00000	.00000	.00000	.00000
PISC = .00024	.00000	.00000	.00000	.00000
COMIN = .01582				

FB-111 ALTITUDE = 3000. FT. L.O. SNEEP ANGLE = 72.50 DEG. TRIMMED CONDITION CC AT CP = 0  
 PACH NO. = 1.200 TOTAL CC CY ALPHA CC LIFT CD RWFT CL AT CP = 0 CC FT CP = 0

CL	TOTAL CC	CY	ALPHA	CC LIFT	CD RWFT	CL AT CP = 0	CC FT CP = 0
.004	.03878	-2.0314	.21	.00003	0.00000	0.00000	.02075
.027	.03905	-1.65275	1.07	.00042	0.00000	.05000	.02515
.070	.04061	-1.10427	1.62	.00224	0.00000	.10000	.04000
.119	.04366	-1.17502	2.75	.00546	0.00000	.15000	.04422
.160	.04733	-2.16572	3.65	.01009	0.00000	.20000	.04805
.200	.05259	-2.65648	4.51	.01614	0.00000	.25000	.05425
.241	.05959	-3.14722	5.27	.02360	0.00000	.30000	.06235
.282	.06717	-3.63826	6.24	.03242	0.00000	.35000	.07122
.323	.07533	-4.12931	7.10	.04275	0.00000	.40000	.08150
.364	.08408	-4.62047	7.96	.05464	0.00000	.45000	.09320
.404	.09341	-5.11172	8.82	.06755	0.00000	.50000	.10630
.445	.10322	-5.60225	9.68	.08137	0.00000	.55000	.12120
.486	.11355	-6.09278	10.54	.09610	0.00000	.60000	.13710
.527	.12438	-6.58331	11.41	.11173	0.00000	.65000	.15400
.568	.13571	-7.07384	12.27	.12826	0.00000	.70000	.17200
.609	.14754	-7.56437	13.13	.14569	0.00000	.75000	.19100
.649	.15987	-8.05490	13.99	.16402	0.00000	.80000	.21100
.690	.17270	-8.54543	14.85	.18325	0.00000	.85000	.23200
.731	.18603	-9.03596	15.71	.20348	0.00000	.90000	.25400
.772	.20086	-9.52649	16.58	.22471	0.00000	.95000	.27700
.812	.21719	-10.01702	17.44	.24694	0.00000	1.00000	.30100

CLA = .05803 PER DEG. ALC = .20531 DEG.  
 K = .02244 DELCL = .01095  
 CMO = 0.00000 CP/CL = -.02509\* (TAIL CFF)

CRAG BREAKDOWN -----

FRICION = .02457	FUSELAGE	WAGELLE	VAR. WING	H.T.	V.T.	STRAKE
FCRM = 0.00000	.00214	.00153	.00140	.00140	.00050	.00000
INTERF = 0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
WAVE = .01725	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
EASE = .00210	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
FRICION = .00611	.00115	.00282	.00140	.00140	.00050	.00000
FCRM = 0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
INTERF = 0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
WAVE = .00657	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
CAMBER = .00004	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
CRAG RISE = 0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
PISC = .00102	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

ADMIN = .03075

NOTE. HAVE DRAG FOR WDF.PINC IS INCLUDED IN HAVE DRAG OF CLCUE

FR-111  
MACH NO. = 1.600 ALTITUDE = 30000. FT. L.O.E. SNEEP ANGLE = 72.50 DEG. TRIPPED CONDITION  
TOTAL CC CH FLUFFS CC LIFT CC R4AF7 CL AT CP = C CC AT CP = C

-003	.03433	-15770	.20	-0000	0.0000	0.0000	.02434
.037	.03450	-17630	1.00	-0000	0.0000	0.0000	.02500
.078	.03685	-175510	1.50	-0000	0.0000	0.0000	.02500
.119	.04020	-170900	2.00	-0000	0.0000	0.0000	.02500
.160	.04454	-240111	3.00	-0000	0.0000	0.0000	.02500
.201	.05117	-300741	4.00	-0000	0.0000	0.0000	.02500
.241	.05608	-359400	5.00	-0000	0.0000	0.0000	.02500
.282	.06077	-418055	6.00	-0000	0.0000	0.0000	.02500
.322	.06544	-476712	7.00	-0000	0.0000	0.0000	.02500
.363	.06980	-535366	8.00	-0000	0.0000	0.0000	.02500
.404	.07409	-594019	9.00	-0000	0.0000	0.0000	.02500
.444	.07828	-652672	10.00	-0000	0.0000	0.0000	.02500
.485	.08234	-711324	11.00	-0000	0.0000	0.0000	.02500
.525	.08629	-770000	12.00	-0000	0.0000	0.0000	.02500
.566	.09016	-828676	13.00	-0000	0.0000	0.0000	.02500
.607	.09396	-887351	14.00	-0000	0.0000	0.0000	.02500
.647	.09768	-946026	15.00	-0000	0.0000	0.0000	.02500
.688	.10132	-1004700	16.00	-0000	0.0000	0.0000	.02500
.728	.10488	-1063174	17.00	-0000	0.0000	0.0000	.02500
.769	.10836	-1121648	18.00	-0000	0.0000	0.0000	.02500
.809	.11176	-1180122	19.00	-0000	0.0000	0.0000	.02500

CLA = .05662 PER DEG. ALC = .15716 DEG.  
K = .30261 DELCL = .00255  
CNC = 0.00000 GR/CL = -.00055\*

(\*TAIL CFF)

CRAG BREAKDOWN  
FRICTION = .00416  
FCRF = 0.00000  
INTEFF = 0.00000  
WAVE = .03551  
BASE = .00210  
GLCVC = .00000  
WAF.MINC  
N.T.  
V.T.  
STRAKE

CLER = .29630  
CLPFX = 1.42361

NOTE. HAVE DRAG FOR VFR-BING IS INCLUDED IN WAVE DRAG OF GLCVC



C-141A FLIGHT TEST ANALYSIS NASA CR - 1552

	4	2	1	3	C	1	1	6	0	1	1.0	
2220.	22.2	61.07										CARC 1.1
C.1005	5.0	-5.50										CARC 1.2
2220.0	0.373	24.38										CARC 1.3
FUSELAGE WHEEL WELL BULLET												CARC 1.4
132.29	33.67	24.95										CARC 1.5
14.17	5.85	3.65										CARC 1.6
4347.52	82.0	136.8										CARC 2.1
1.0	1.2	1.3										CARC 2.2
157.63	26.83	10.47										CARC 2.3
C.C	C.C	0.0										CARC 2.4
17.00	16.82	12.50										CARC 2.5
50.82	16.83	12.08										CARC 2.6
0.0	0.0	0.0										CARC 2.7
WING P1	WING P2	H. TAIL	V. TAIL									CARC 2.8
-62	-62	64A	64A									CARC 2.9
0.4	0.4	0.0	0.0									CARC 2.10
C.1155	0.11	0.105	C.11									CARC 2.11
23.72	25.02	29.0	40.0									CARC 2.12
3134.2	2922.2	693.74	219.4									CARC 2.13
23.82	20.06	14.2	23.0									CARC 3.1
20.06	10.59	5.0	13.6									CARC 3.2
27.1	46.4	25.17	22.72									CARC 3.3
37.75	51.1	123.7	110.2									CARC 3.4
6.5	33.6	6.0	8.0									CARC 3.5
0.0	0.0	0.0	23.5									CARC 3.6
4.69	0.0	0.0	0.0									CARC 3.7
0.2	3000.0	0.2										CARC 3.8
1.2	3000.0	0.2										CARC 3.9
NO ADJUSTMENTS												CARC 3.10
CAPACE CASES FOLLO												CARC 3.11
C-141A WITH NUCLEAR CAPAGE												CARC 3.12
111111	111111											CARC 3.13
FUSELAGE WHEEL WELL BULLET												CARC 6.1
C.1002												CARC 6.2
0.1												CARC 7.1
0.5												CARC 9.1
4720.0												CARC 9.2
												CARC 9.3
												CARC 10.1
												CARC 10.2
												CARC 10.3
												CARC 10.4
												CARC 10.5
												CARC 10.6
												CARC 10.7
												CARC 10.8
												CARC 10.9
												CARC 10.10
												CARC 10.11
												CARC 10.12
												CARC 10.13
												CARC 10.14
												CARC 10.15
												CARC 10.16
												CARC 10.17
												CARC 10.18
												CARC 10.19
												CARC 10.20
												CARC 10.21
												CARC 10.22
												CARC 10.23
												CARC 10.24
												CARC 10.25
												CARC 10.26
												CARC 10.27
												CARC 10.28
												CARC 10.29
												CARC 10.30
												CARC 10.31
												CARC 10.32
												CARC 10.33
												CARC 10.34
												CARC 10.35
												CARC 10.36
												CARC 10.37
												CARC 10.38
												CARC 10.39
												CARC 10.40
												CARC 10.41
												CARC 10.42
												CARC 10.43
												CARC 10.44
												CARC 10.45
												CARC 10.46
												CARC 10.47
												CARC 10.48
												CARC 10.49
												CARC 10.50
												CARC 10.51
												CARC 10.52
												CARC 10.53
												CARC 10.54
												CARC 10.55
												CARC 10.56
												CARC 10.57
												CARC 10.58
												CARC 10.59
												CARC 10.60
												CARC 10.61
												CARC 10.62
												CARC 10.63
												CARC 10.64
												CARC 10.65
												CARC 10.66
												CARC 10.67
												CARC 10.68
												CARC 10.69
												CARC 10.70
												CARC 10.71
												CARC 10.72
												CARC 10.73
												CARC 10.74
												CARC 10.75
												CARC 10.76
												CARC 10.77
												CARC 10.78
												CARC 10.79
												CARC 10.80
												CARC 10.81
												CARC 10.82
												CARC 10.83
												CARC 10.84
												CARC 10.85
												CARC 10.86
												CARC 10.87
												CARC 10.88
												CARC 10.89
												CARC 10.90
												CARC 10.91
												CARC 10.92
												CARC 10.93
												CARC 10.94
												CARC 10.95
												CARC 10.96
												CARC 10.97
												CARC 10.98
												CARC 10.99
												CARC 11.00
												CARC 11.01
												CARC 11.02
												CARC 11.03
												CARC 11.04
												CARC 11.05
												CARC 11.06
												CARC 11.07
												CARC 11.08



EMPIRICALLY BASED COMPUTER PROGRAM  
TO PREDICT THE AERODYNAMIC CHARACTERISTICS  
OF AIRCRAFT  
INCLUDING  
ROUGHNESS, PROTRUSIONS, AND  
DAMAGE CALCULATIONS

REFPRINT(28) = 1                      C-141A FLIGHT TEST ANALYSIS                      NASA CR - 1558

PROBLEP INPUT PARAMETERS

NCCS = 4  
 APALS = 2  
 RHT = 1  
 RVT = 3  
 ISNP = 0  
 IREF = 1  
 IIMG = 1  
 NAFI = 0  
 PETER = 0  
 ITRIP = 1

SREF = 3228.00  
 CPAC = 22.200  
 XPAC = 61.070  
 ICC = 0.000  
 UFFUS = 7.000  
 ACE = 1.000

ROUGHN = .00050  
 FMISC = 5.00  
 TWIST = -5.580  
 CONCL = 0.000

DIMENSIONS ARE IN FEET

REFERENCE WING FOR TWO PANEL CASE

SPLAN = 3228.000 TAPR = .3730 SWP = 24.300

	FUSELAGE	WHEEL	WELL	BULLET	NACELLE
LENGTH =	132.290	33.670	24.990	16.600	
WIDTH =	14.170	5.850	3.650	5.500	
HEIGHT =	14.170	5.850	3.650	5.500	
WETTED AREA =	4347.520	822.000	136.800	1045.960	
INTER.FACTOR =	1.000	1.000	1.300	1.200	
NO.OF THIS TYPE =	1.000	2.000	1.000	4.000	
MAXIMUM AREA =	157.630	26.830	10.470	23.760	
BASE AREA =	0.000	0.000	0.000	0.000	
MCSE LENGTH =	17.000	16.820	12.510	8.250	
BCATTAIL LENGTH =	50.830	16.820	12.000	8.330	
BASE DRAG AREA =	0.000	0.000	0.000	0.000	
INLET AREA =	0.000	0.000	0.000	12.500	



SECTION	=	-E3	-E3	E4A	E-A	E4A	E4A	E4A
CHIEF (2-D)	=	.400	.400	0.000	0.000	0.000	0.000	0.030
THICKNESS RATIO	=	.120	.110	.105	.130	.100	.100	.100
L.E. SHEEF	=	23.730	25.020	29.000	40.000	23.000	73.000	73.030
WETTED AREA	=	3134.200	2922.200	893.740	815.400	223.240	214.880	
RCCT CHCRC	=	33.200	30.000	14.200	23.000	14.000	14.000	14.000
TIP CHCRD	=	20.000	10.500	5.000	12.000	14.000	14.000	14.000
SFAN OF PANEL	=	27.100	46.400	25.170	22.720	3.200	3.170	
X- LE RCOT CHCRD	=	37.750	51.100	123.700	110.200	45.000	37.100	
Y- LE RCOT CHCRD	=	6.500	22.000	0.000	0.000	28.330	23.750	
Z- LE RCOT CHCRD	=	0.000	0.000	23.000	2.240	-4.400	-4.400	
INCIDENCE	=	4.890	0.000	0.000	0.000	0.000	0.000	

2 CRAG POLARS TO BE GENERATED AT THE FOLLOWING CONDITIONS

PACH NO.	ALTITUDE	C.G.	SHEEP	FROM CL	TO CL
020	30000.000	020	24.4	0.00	1.00
120	30000.000	020	24.4	0.00	1.00

NO ADJUSTMENTS \*\*\*\*\*

CALCULATED ECCY PARAMETERS

	FUSELAGE	WHEEL	HELL	WHEEL	WHEEL	WHEEL	WHEEL
FINENESS RATIO =	5.336	5.756	6.247	6.247	6.247	6.247	6.247
WETTED AREA =	4347.520	222.000	136.800	136.800	136.800	136.800	136.800
MAXIMUM AREA =	157.630	26.830	10.470	10.470	10.470	10.470	10.470

CALCULATED SURFACE PARAMETERS

	WING F1	WING F2	H. TAIL	V. TAIL	CUTED PYL	INBC FYL
WETTED AREA =	3134.200	2922.200	893.740	819.400	222.240	214.800
TAPER RATIO =	.664	.548	.250	.591	1.000	1.000
PLATFORM AREA =	1442.246	1440.720	485.781	415.776	55.776	52.564
ASPECT RATIO =	2.035	5.577	5.217	2.463	.445	.434
CHARACT. LENGTH =	27.172	15.567	10.357	18.702	14.601	14.601
S/P C/4 =	17.668	22.685	24.754	26.241	73.000	73.000
S/P C/2 =	11.153	20.265	20.263	22.202	73.000	73.000
S/P TE =	-2.592	15.122	10.471	23.042	73.000	73.000
S/P MAX T/C =	16.350	22.210	22.204	24.127	73.000	73.000
X/C AT MAX T/C =	.200	.300	.250	.250	.350	.390

TOTAL AIRPLANE PARAMETERS

WING SEMI-SPAN	*BC2 =	80.600
EGCY CIA/ WING SPAN	*DC2 =	.061
WING FOOT CHORD AT CENTERLINE	*CR =	36.252
THEO. WING PLATFORM AREA	*SELAN =	3228.800
X-STA OF RCCT CHORD (CR) T.E.	*XCITE =	72.857
EQUIV. WING ASPECT RATIO	*AR =	7.931
EQUIV. WING TAPER RATIO	*TAR =	.373
EQUIV. WING LEADING EDGE SWEEP	*SALE =	24.283
EQUIV. WING QUARTER-CHORD SWEEP	*SAFCQ =	20.322
EQUIV. WING MIC-CHORD SWEEP	*SAFPC =	16.226
EQUIV. WING TRAILING-EDGE SWEEP	*SAETE =	6.444
EQUIV. WING MAX THICKNESS SWEEP	*SMT =	15.500
EQUIV. WING EXPOSED AREA	*SEXA =	2084.066
EQUIV. WING EXPOSED MAC	*CBAR =	21.575
EQUIV. WING THICKNESS RATIO	*TCCH =	.115
EQUIV. WING 2-C DESIGN CL	*CLC =	.400
EQUIV. WING EXPOSED ASPECT RATIO	*ARXF =	7.453
CHORD LENGTH AT WING FCCT	*CR > =	33.200
CHORD LENGTH AT WING TIF	*CTX =	10.990
X-STA. AT L.E. OF WING RCCT	*XRX =	39.452
X-STA. AT L.E. OF WING TIF	*XTX =	72.766
Y-STA. AT L.E. OF WING FCCT	*YRX =	6.500
Y-STA. AT L.E. OF WING TIF	*YTX =	80.600
X-STA. AT L.E. OF WING PAC	*XE =	51.876
Y-STA. AT L.E. OF WING PAC	*YE =	33.911
CELEPA-X C/A WING TO C/4 TAIL	*XP =	74.869
INCLINATION LINE BET. WING AND TAIL	*CPEGA =	18.253
ENTER SURVEY		

4-1102, A75-FOIL SECTION

MACH CRITICAL TABLE

CL	MACH CRITICAL
0.0000	.6294
.1000	.6165
.2000	.6045
.3000	.5930
.4000	.5820
.5000	.5715
.6000	.5615
.7000	.5520
.8000	.5430
.9000	.5345
1.0000	.5265

FUSLAGE AFT-ENG UPSWEEP CRAG

WING ANGLE	DELCC
0.0000	.00189
1.0000	.00155
2.0000	.00126
3.0000	.00102
4.0000	.00081
5.0000	.00063
6.0000	.00048
7.0000	.00035
8.0000	.00024
9.0000	.00016
10.0000	.00010
11.0000	.00007
12.0000	.00006
13.0000	.00006
14.0000	.00007
15.0000	.00007
16.0000	.00007
17.0000	-.00001
18.0000	-.00007
19.0000	-.00017
20.0000	-.00025

C-141A FLIGHT TEST ANALYSIS NPSA CR - 1552  
 MACH NO. = .800 ALTITUDE = 30000. FT. SWEEP ANGLE = 24.38 DEG. TAIL DEFL. (OH) = 0.00 DEG.

CL	TOTAL CC	CR	ALPHA	CCL	CCR	COAFT	CC(L*F*H/FT)
0.000	.01829	.08021	-4.35	.00076	0.00000	.00171	.00445
.050	.01762	.07454	-3.60	.00024	0.00000	.00148	.00372
.100	.01716	.06884	-2.86	.00001	0.00000	.00128	.00312
.150	.01708	.06253	-2.04	.00000	0.00000	.00110	.00246
.200	.01729	.05602	-1.22	.00041	0.00000	.00094	.00180
.250	.01765	.04932	-.60	.00115	.00002	.00079	.00116
.300	.01850	.04244	-.09	.00214	.00000	.00066	.00050
.350	.02075	.03540	.63	.00243	.00000	.00055	.00000
.400	.02317	.02820	1.35	.00000	.00180	.00045	.00000
.450	.02633	.02086	2.07	.00000	.00315	.00036	.00000
.500	.03018	.01340	2.79	.00014	.00487	.00028	.00000
.550	.03455	.00562	3.51	.00166	.00719	.00021	.00000
.600	.04040	-.00147	4.22	.01448	.00996	.00015	.00000
.650	.04707	-.00964	4.94	.01760	.01347	.00011	.00000
.700	.05456	-.01749	5.66	.02107	.01751	.00008	.00000
.750	.06407	-.02541	6.38	.02456	.02245	.00007	.00000
.800	.07922	-.03371	7.10	.02852	.02825	.00006	.00000
.850	.10488	-.04312	7.82	.03274	.03495	.00006	.00000
.900	.13793	-.05385	8.53	.03754	.04243	.00007	.00000
.950	.17919	-.06602	9.25	.04301	.05022	.00007	.00000
1.000	.22811	-.07941	9.97	.04915	.05840	.00007	.00000

CLA = .06982 PER DEG. ALC = -4.35481 DEG.  
 K = .85102 DELCL = .11292  
 CM0 = -.05879\* CM/CL = .46019\*  
 CLM = .75056  
 CLPM = 1.41022

CIRAG BREAKDOWN	FUSELAGE	WHEEL WELL	BULLETS	NADELLE	CUIBEC FYL	INEC FYL
FRICITION = .00371	.00233	.00052	.00000	.00075	.00000	.00016
FCPP = .00051	.00023	.00012	.00002	.00005	.00016	.00003
INTERF = .00046	.00004	.00014	.00003	.00025	.00012	.00005
HAVE = 0.00000	0.00000	0.00000	0.00000	0.00000	.00000	.00000
CASE = 0.00000	0.00000	0.00000	0.00000	0.00000	.00000	.00000
FRICITION = .00579	.00209	.00211	H.TAIL	V.TAIL	CUIBEC FYL	INEC FYL
FCPP = .00128	.00051	.00047	.00009	.00050	.00016	.00016
INTERF = .00197	.00074	.00073	.00012	.00012	.00012	.00003
HAVE = 0.00000	0.00000	.00073	.00021	.00019	.00001	.00005
CAPBER. = .00150	0.00000	0.00000	.00000	.00000	.00000	.00000
CIRAG RISE = 0.00000	0.00000	0.00000	.00000	.00000	.00000	.00000
PISC = .00069						
GMNIN = .01569						

C-141A FLIGHT TEST ANALYSIS NASA CR - 1550 SNEEP ANGLE = 24.38 DEG. TAIL DEFL. (CH) = 0.00 DEG.

MACH NO. = 1.200 ALTITUDE = 30000. FT. SNEEP ANGLE = 24.38 DEG.

CL	TOTAL CC	CP	ALPHA	CDL	CDP	CCAFI	CD(L+R+AFFI)
0.000	.15077	.06868	-4.01	0.0000	0.0000	.00159	.00159
.050	.15861	.06710	-3.56	.0038	0.0000	.00145	.00145
.100	.15961	.06546	-3.11	.0075	0.0000	.00132	.00132
.150	.16127	.06275	-2.67	.0112	0.0000	.00120	.00120
.200	.16369	.06158	-2.22	.0150	0.0000	.00109	.00109
.250	.16717	.06015	-1.78	.0187	0.0000	.00099	.00099
.300	.17121	.05827	-1.33	.0225	0.0000	.00095	.00095
.350	.17600	.05622	-.89	.0262	0.0000	.00081	.00081
.400	.18155	.05424	-.44	.0299	0.0000	.00072	.00072
.450	.18726	.05231	.00	.0336	0.0000	.00065	.00065
.500	.19452	.05022	.45	.0375	0.0000	.00057	.00057
.550	.20274	.04810	.90	.0414	0.0000	.00051	.00051
.600	.21132	.04593	1.34	.0454	0.0000	.00045	.00045
.650	.22005	.04372	1.79	.0495	0.0000	.00039	.00039
.700	.23078	.04147	2.23	.0536	0.0000	.00034	.00034
.750	.24231	.03919	2.68	.0577	0.0000	.00029	.00029
.800	.25567	.03688	3.12	.0618	0.0000	.00024	.00024
.850	.27024	.03454	3.57	.0659	0.0000	.00020	.00020
.900	.28663	.03217	4.01	.0700	0.0000	.00017	.00017
.950	.30483	.02978	4.46	.0741	0.0000	.00014	.00014
1.000	.32426	.02736	4.90	.0782	0.0000	.00011	.00011

CLIP = 1.00000  
CLIPX = 1.44521

CL A = .1126 PER DEG.  
K = .15023  
CMC = 0.00000

ALC = -4.00517 DEG.  
DELCL = 0.00000  
CM/CL = .02016

(\*TAIL CFF)

DRAG BREAKDOWN

FRICITION = .00322	FUSELAGE	WHEEL WELL	BULLET	NACELLE	CUIBD FYL	IARC FYL
FCRM = 0.00000	.00210	.00040	.00000	.00007	.00015	.00014
INTERF = 0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
WAVE = .04112	.02270	.00000	.00000	.01402	0.00000	0.00000
EASE = 0.00000	0.00000	0.00000	0.00000	.00000	.00000	.00000
FRICITION = .00515	WING F1	WING F2	H.TAIL	V.TAIL	CUIBD FYL	IARC FYL
FCFP = 0.00000	.00127	.00189	.00000	.00052	.00015	.00014
INTERF = 0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
WAVE = .00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
CAMBER = 0.00000	.00000	0.00000	.00000	.00000	.00000	.00000
CRAG RISE = 0.00000	0.00000	0.00000	0.00000	.00000	.00000	.00000
PISC = .00042						
COMIN = .15679						

NOTE. WAVE CRAG FCR WING F2 IS INCLUDED IN WAVE CRAG CF WING F1

C-141A WITH NUCLEAR DSPACE

DSPACE PARAMETERS

IDAM( 1) = 1      ICAP(11) = 1  
 IDAM( 2) = 1      ICAP(12) = 1  
 ICAM( 3) = 1      ICAP(13) = 1  
 IDAM( 4) = 1      ICAP(14) = 1  
 IDAM( 5) = 1      ICAP(15) = 1  
 IDAM( 6) = 1      ICAP(16) = 1  
 IDAM( 7) = 1      ICAP(17) = 1

ECCY CAPAGE PARAMETERS, DECO(I,J)

I = 1    I = 2    I = 3    I = 4    I = 5    I = 6    I = 7

MCOE 1    ECCY, FOURNESS  
 J = 1    .000200    0.000000    0.000000    0.000000    0.000000  
 J = 2    .000500    0.000000    0.000000    0.000000    0.000000  
 J = 3    .100000    0.000000    0.000000    0.000000    0.000000  
 J = 4    .500000    0.000000    0.000000    0.000000    0.000000  
 J = 5    472.000000    0.000000    0.000000    0.000000    0.000000

MCOE 2    ECCY,FY-C-FACING STEP  
 J = 11    0.000000    0.000000    4.000000    0.000000    0.000000  
 J = 12    0.000000    0.000000    2.000000    0.000000    0.000000  
 J = 13    0.000000    0.000000    .100000    0.000000    0.000000  
 J = 14    0.000000    0.000000    .400000    0.000000    0.000000  
 J = 15    0.000000    0.000000    .500000    0.000000    0.000000

MCOE 3    ECCY,AFT-FACING STEP  
 J = 21    0.000000    2.000000    0.000000    0.000000    0.000000  
 J = 22    0.000000    0.000000    0.000000    0.000000    0.000000  
 J = 23    0.000000    .300000    0.000000    0.000000    0.000000  
 J = 24    0.000000    .300000    0.000000    0.000000    0.000000  
 J = 25    0.000000    .000000    0.000000    0.000000    0.000000

MCOE 4    ECCY, FCLES  
 J = 31    2.000000    0.000000    0.000000    1.000000  
 J = 32    2.000000    0.000000    0.000000    0.000000  
 J = 33    4.000000    0.000000    0.000000    4.000000  
 J = 34    2.000000    0.000000    0.000000    2.000000  
 J = 35    .500000    0.000000    0.000000    1.000000  
 J = 36    .200000    0.000000    0.000000    .500000  
 J = 37    .700000    0.000000    0.000000    .500000  
 J = 38    1.000000    0.000000    0.000000    2.000000

MCOE 5    ECCY, FAVINESS  
 J = 41    4.000000    0.000000    0.000000    1.000000  
 J = 42    2.000000    0.000000    0.000000    0.000000  
 J = 43    2.000000    0.000000    0.000000    6.000000  
 J = 44    4.000000    0.000000    0.000000    5.000000  
 J = 45    .500000    0.000000    0.000000    .400000  
 J = 46    .300000    0.000000    0.000000    .400000  
 J = 47    .300000    0.000000    0.000000    .400000



J	MCDE	6	ECCY	FRACTUREANCES
J = 51	0.000000	0.000000	2.000000	0.000000
J = 52	.300000	0.000000	.500000	0.000000
J = 53	.500000	0.000000	.750000	0.000000
J = 54	.400000	0.000000	.400000	0.000000
J = 55	.800000	0.000000	.500000	0.000000

J	MCDE	7	BCCY	ELUYNNESS
J = 61	12.570000	0.000000	8.000000	0.000000
J = 62	.100000	0.000000	.400000	0.000000

G-141A WITH NUCLEAR CAPAGE

SURFACE CAPAGE PARAMETERS, DSUR(I,J)

I = 1 I = 2 I = 3 I = 4 I = 5 I = 6 I = 7

J =	I = 1	I = 2	I = 3	I = 4	I = 5	I = 6	I = 7
MCDE 11	SURF, FROTHNESS						
J = 1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
MCDE 12	SURF, FLE-FACING STEP						
J = 11	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 12	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 13	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 14	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 15	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
MCDE 13	SURF, AFT-FACING STEP						
J = 21	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 22	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 23	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 24	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 25	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 26	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
MCDE 14	SURF, PGLES						
J = 31	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 32	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 33	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 34	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 35	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 36	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 37	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 38	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 39	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
MCDE 15	SURF, ADVJNESS						
J = 41	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 42	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 43	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 44	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 45	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 46	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 47	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
MCDE 16	SURF, FROTHREFERENCES						
J = 51	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 52	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 53	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 54	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J = 55	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

SURFACE MISSING PART PARAMETERS, ENINGII

OWING(1) =	.250	OWING( 7) =	.110	OWING(13) =	.600
OWING(2) =	.220	OWING( 8) =	.220	OWING(14) =	.050
OWING(3) =	.310	OWING( 9) =	.230	OWING(15) =	.050
OWING(4) =	.200	OWING(10) =	3.000	OWING(16) =	.250
OWING(5) =	1.000	OWING(11) =	.150	OWING(17) =	.300
OWING(6) =	3.000	OWING(12) =	1.000	OWING(18) =	.600
				OWING(19) =	0.000
				OWING(20) =	1.000

ENTER NUCCAP

C-141F WITH NUCLEAR DAMAGE

SUMMARY OF DAMAGE EFFECTS ON PINGUP DRAG

K A C H = .00

BODY DAMAGE PCOE	FLUSSEAGE	WHEEL	WELL	BULLET	NASCELLE	CUTED PYL	INED PYL
ROUGHNESS	-.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
FAC-FACING STEPS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
AFT-FACING STEPS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
POLES	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
WAVINESS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
PROTUBERANCES	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
ELUNTNES	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
TOT.DRAG.EA.BODY =	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

SURFACE DAMAGE MCODE	PING P1	PING P2	H.TAIL	V.TAIL	CUTED PYL	INED PYL
ROUGHNESS	-.00000	0.00000	0.00000	0.00000	0.00000	0.00000
FAC-FACING STEPS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
AFT-FACING STEPS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
POLES	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
WAVINESS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
PROTUBERANCES	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
MISSING PARTS	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
TOT.DRAG.EA.SURFACE =	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

NET DRAG FROM EACH PCOE OF DAMAGE	BODIES	SURFACES	BODIES + SURFACES
ROUGHNESS	-.00000	0.00000	0.00000
FAC-FACING STEPS	0.00000	0.00000	0.00000
AFT-FACING STEPS	0.00000	0.00000	0.00000
POLES	0.00000	0.00000	0.00000
WAVINESS	0.00000	0.00000	0.00000
PROTUBERANCES	0.00000	0.00000	0.00000
ELUNTNES	0.00000	0.00000	0.00000
MISSING PARTS	0.00000	0.00000	0.00000
TOTAL DAMAGE DRAG	0.00000	0.00000	0.00000

C-141A WITH NUCLEAR CAPAGE

FACT = .800

CL	UNCAPACED		U N T R I M E C		P E R C D N A P I C		O A I A		I N C R E M E N T S D U E T O N U C L E A R C A P A G E		D E L T A	
	ALPHA	CP	CC	CD	ALPHA	CP	CD	ALPHA	CP	DELTA	DELTA	DELTA
0.000	-4.39	.6808	.01839	.02826	X	-7.065	.02826	X	-1.93	.01124	-.01127	.0097
.050	-3.68	.6749	.01762	.02821	X	-6.361	.02821	X	-1.62	.01124	-.00006	.01119
.100	-2.96	.6688	.01710	.02923	X	-5.656	.02923	X	-1.31	.01124	.01121	.01125
.150	-2.24	.6625	.01702	.02923	X	-4.952	.02923	X	-1.06	.01124	.01121	.01275
.200	-1.52	.6560	.01725	.03106	X	-4.250	.03106	X	-.72	.01124	.01124	.01277
.250	-.80	.6493	.01785	.03256	X	-3.550	.03256	X	-.50	.01124	.01124	.01288
.300	-.05	.6424	.01852	.03573	X	-2.852	.03573	X	-.23	.01124	.01124	.01511
.350	.63	.6354	.01975	.03947	X	-2.155	.03947	X	.05	.01124	.01124	.01672
.400	1.35	.6282	.02117	.04417	X	-1.460	.04417	X	.33	.01124	.01124	.01876
.450	2.07	.6209	.02263	.04992	X	-.766	.04992	X	.61	.01124	.01124	.02110
.500	2.79	.6134	.02418	.05665	X	.018	.05665	X	.85	.01124	.01124	.02359
.550	3.51	.6058	.02595	.06446	X	.618	.06446	X	1.17	.01124	.01124	.02651
.600	4.22	.5981	.02794	.07334	X	1.208	.07334	X	1.45	.01124	.01124	.02971
.650	4.94	.5906	.02970	.08415	X	1.857	.08415	X	1.72	.01124	.01124	.03326
.700	5.66	.5831	.03145	.09579	X	2.688	.09579	X	2.00	.01124	.01124	.03700
.750	6.38	.5754	.03327	.10946	X	3.374	.10946	X	2.28	.01124	.01124	.04122
.800	7.10	.5677	.03522	.12511	X	4.052	.12511	X	2.56	.01124	.01124	.04579
.850	7.82	.5601	.03731	.15571	X	4.737	.15571	X	2.84	.01124	.01124	.05063
.900	8.53	.5526	.03953	.18034	X	5.416	.18034	X	3.12	.01124	.01124	.05583
.950	9.25	.5450	.04195	.20497	X	6.096	.20497	X	3.40	.01124	.01124	.06147
1.000	9.97	.5375	.04458	.22960	X	6.776	.22960	X	3.68	.01124	.01124	.06755

C-141A WITH NUCLEAR DAMAGE

SUMMARY OF DAMAGE EFFECTS ON MINIZAP DATA

P A C H = 1.20

BODY DAMAGE PCODE	FUSELAGE	WHEEL WELL	WHEEL	NACELLE	OLTED PYL	INFLATED PYL
ROUGHNESS	.00000	.00000	.00000	.00000	.00000	.00000
FWD-FACING STEPS	.00000	.00000	.00000	.00000	.00000	.00000
AFT-FACING STEPS	.00000	.00000	.00000	.00000	.00000	.00000
POLES	.00000	.00000	.00000	.00000	.00000	.00000
RAVINESS	.00000	.00000	.00000	.00000	.00000	.00000
FRCTUBERANCES	.00000	.00000	.00000	.00000	.00000	.00000
ELUNTNESS	.00000	.00000	.00000	.00000	.00000	.00000
TOT. DRAG, EA. BODY =	.00000	.00000	.00000	.00000	.00000	.00000

SURFACE DAMAGE PCODE	WING P1	WING F2	H-TAIL	V-TAIL	OLTED PYL	INFLATED PYL
ROUGHNESS	.00000	.00000	.00000	.00000	.00000	.00000
FWD-FACING STEPS	.00000	.00000	.00000	.00000	.00000	.00000
AFT-FACING STEPS	.00000	.00000	.00000	.00000	.00000	.00000
POLES	.00000	.00000	.00000	.00000	.00000	.00000
RAVINESS	.00000	.00000	.00000	.00000	.00000	.00000
FRCTUBERANCES	.00000	.00000	.00000	.00000	.00000	.00000
MISSING PARTS	.00000	.00000	.00000	.00000	.00000	.00000
TOT. DRAG, EA. SURFACE =	.00000	.00000	.00000	.00000	.00000	.00000

NET DRAG FRCP EACH PCODE OF DAMAGE	BODIES	SURFACES	BODIES + SURFACES
ROUGHNESS	.00000	.00000	.00000
FWD-FACING STEPS	.00000	.00000	.00000
AFT-FACING STEPS	.00000	.00000	.00000
POLES	.00000	.00000	.00000
RAVINESS	.00000	.00000	.00000
FRCTUBERANCES	.00000	.00000	.00000
ELUNTNESS	.00000	.00000	.00000
MISSING PARTS	.00000	.00000	.00000
TOTAL DAMAGE DRAG	.00000	.00000	.00000

C-141A WITH NUCLEAR CHARGE

PAGE = 1.200

CL	UNCAPAGED		UNIT R		I M H E C		A E R C D \ N A P I C		D E L T A		I N C R E M E N T S I E T C N U C L E A R I M P A G E		D E L T A	
	ALPHA	CM	CC	CP	ALPHA	CP	CC	ALPHA	CM	DELTA	DELTA	DELTA	DELTA	DELTA
0.000	-4.01	.0687	.15837	-.3577	-4.86	-.3577	.17627	-.85	-.4653	.0136	.01879	-.00111	.01765	
.050	-3.56	.0671	.15861	-.3691	-4.33	-.3691	.17714	-.77	-.4362	.0124	.01879	-.0024	.01553	
.100	-3.11	.0655	.15961	-.3405	-3.81	-.3405	.17914	-.70	-.4160	.0112	.01879	-.0074	.01553	
.150	-2.67	.0637	.16137	-.3150	-3.29	-.3150	.18111	-.62	-.3758	.0099	.01879	-.0095	.01574	
.200	-2.22	.0620	.16385	-.2835	-2.77	-.2835	.18257	-.54	-.3455	.0087	.01879	-.0122	.01550	
.250	-1.78	.0611	.16717	-.2551	-2.24	-.2551	.18771	-.46	-.3152	.0074	.01879	-.0175	.01521	
.300	-1.33	.0583	.17121	-.2268	-1.72	-.2268	.19274	-.35	-.2850	.0062	.01879	-.0234	.01512	
.350	-.85	.0563	.17600	-.1975	-1.21	-.1975	.19735	-.31	-.2548	.0049	.01879	-.0306	.01515	
.400	-.44	.0543	.18155	-.1702	-.67	-.1702	.20425	-.23	-.2245	.0037	.01879	-.0391	.01517	
.450	.00	.0523	.18786	-.1428	-.15	-.1428	.21154	-.15	-.1942	.0025	.01879	-.0489	.01518	
.500	.45	.0502	.19492	-.1138	.31	-.1138	.21921	-.08	-.1641	.0012	.01879	-.0595	.01522	
.550	.85	.0481	.20274	-.0857	.90	-.0857	.22876	.00	-.1338	-.0000	.01879	-.0722	.01526	
.600	1.24	.0459	.21132	-.0577	1.42	-.0577	.23870	.08	-.1026	-.0012	.01879	-.0869	.01530	
.650	1.79	.0437	.22065	-.0296	1.94	-.0296	.24949	.16	-.0732	-.0025	.01879	-.1024	.01534	
.700	2.23	.0415	.23072	-.0016	2.47	-.0016	.26128	.23	-.0431	-.0038	.01879	-.1191	.01538	
.750	2.68	.0392	.24231	.0263	2.95	.0263	.27454	.31	-.0129	-.0050	.01879	-.1364	.01542	
.800	3.12	.0369	.25547	.0543	3.43	.0543	.28956	.35	.0174	-.0062	.01879	-.1543	.01546	
.850	3.57	.0345	.27024	.0822	4.02	.0822	.30633	.47	.0476	-.0075	.01879	-.1730	.01550	
.900	4.01	.0322	.28663	.1100	4.56	.1100	.32488	.54	.0779	-.0087	.01879	-.1924	.01554	
.950	4.46	.0298	.30462	.1379	5.08	.1379	.34476	.62	.1081	-.0100	.01879	-.2122	.01558	
1.000	4.90	.0274	.32426	.1657	5.60	.1657	.36456	.70	.1384	-.0112	.01879	-.2324	.01562	

## 4. PROGRAM DESCRIPTION

The AAT computer code is a long and complex program that uses many variables and subroutines. For this reason, a programmer who is not familiar with the details of the program operation may have difficulty in making modifications and additions. This program description is presented to aid the programmer in any future development work.

### 4.1 PROGRAM STRUCTURE

Figure 1-1 shows the program overlay structure and the positions of the 59 subprograms and subroutines that comprise the AAT computer code. The deck must be arranged as indicated by this figure; Overlay (0,0) with its program and subroutine are first, followed by Overlay (1,0), Overlay (2,0), Overlay (3,0), Overlay (3.1), Overlay (3,2), Overlay (3,3), and finally, Overlay (4,0).

The overlay structure was devised to allow a logical path through the program while keeping the required computer central memory core to a reasonable amount. Approximate 54,000 bites are required by the program.

### 4.2 SUBROUTINE DESCRIPTIONS

This section contains a description of each subprogram and subroutine listed in alphabetical order. The purpose of each is briefly discussed, followed by the statement required to call the subroutine and definitions for the variables used in the CALL statement.



SUBROUTINE DESCRIPTIONS  
(listed in alphabetical order)

1. Subroutine AALO computes the zero-lift angle of attack, which is dependent upon camber, twist, and incidence. This subroutine is called with the statement

CALL AALO (SPEED)

where speed is the Mach number. The computed value of zero-lift angle is entered into COMMON OUTPUT.

2. Subroutine 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. Other subroutines called are: TLNT, AERA, and LNTP. This subroutine is called with the statement

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

where the 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 the 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

3. Subroutine ADCL computes 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. This subroutine is called with the statement

CALL ADCL(SPEED, CLOPT)

where SPEED is the Mach number. At subsonic speeds, CLOPT is the  $C_L$  for minimum profile drag at supersonic speeds CLOPT is the polar displacement  $C_L$  for  $C_{D_{min}}$ .

4. Subroutine AT62 uses the altitude and equations representing the 1962 U.S. Standard Atmosphere to calculate the unit Reynolds number over Mach number,  $\frac{R_N}{\text{Length} \cdot \text{Mach}}$ . This subroutine is called with the statement

CALL AT 62 (ALT, RNOMFT, IFT)

where ALT = altitude

RNOMFT = Reynolds number divided by Mach x Length

IFT = Indicator to specify English or metric units.

If IFT = 0 units are in feet

If IFT = 1 units are in meters

5. Subroutine ADJUST 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. These factors are functions of either Mach number or  $C_L$ . Subroutine LNTP is also called. This subroutine is called with the statement

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  $y_2$

6. Subroutine AERA calculates the angle of attack for a given untrimmed  $C_L$  condition. For supersonic Mach numbers

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

For subsonic conditions the angle of attack is calculated by one of the three different methods depending on whether the wing is a high aspect ratio, low aspect ratio or a cranked planform. For low-aspect-ratio and cranked planform the effect of vortex lift is accounted for in the angle of attack calculations. Other subroutines called are: DLNT and LNTP. This subroutine is called with the statement

```
CALL AERA (SPEED, CL, ALPHA)
```

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

7. Program AEROA controls the sequence of calculations required to compute minimum drag for a given Mach-altitude or Mach-Reynolds number condition which are contained in COMMON OUTPUT.

The sequence of the calling of subroutines for aerodynamic calculations is controlled by the parameter JPASS which is defined by program SURVEY. The parameter JPASS is used to prevent calling certain subroutines on repeat passes through AEROA if the values they calculate remains fixed. AEROA calls subroutines AT62, DMIN, FDRG, LNTP and WDRG, and AEROA is called from program SURVEY with the statement

```
CALL OVERLAY(4 HOVLY, 3,3)
```

8. Program AEROB controls the sequence of calculations required to compute the trimmed lift, drag, moment, and angle of attack for a given set of conditions. The results of the aerodynamic calculations are contained in COMMON OUTPUT.

The subroutines directly called from program AEROB are ADJUST, AERA, AFTCD, CDDR, CDL1, CDL2, CDRG, CLBRK, CLWBT, CMOW, TDRG, and WBAC. The parameter JPASS is used to prevent calling certain subroutines on repeat passes through AEROB if the value they calculate remains fixed. Program AEROB is called from program SURVEY with the statement

```
CALL OVERLAY(4 HOVLY, 3,4)
```

9. Subroutine AER2 computes the lift-curve slope for a wing surface defined by the data in COMMON CALC. 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. This subroutine is called with the statement

```
CALL AER2 (SPEED, CLA)
```

where SPEED = Mach number

CLA = Computed lift-curve slope ( $\Delta C_L$  per degree).

10. Subroutine AFTCD computes the drag increment due to the fuselage aft-end upsweep as a function of angle of attack. Subroutine DLNT is also called. This subroutine is called with the statement

```
CALL AFTCD (ALPHA, CDAFT)
```

where ALPHA = Angle of attack of the wing

CDAFT = Drag increment due to aft-end upsweep

11. Subroutine BDRG computes the base drag of an arbitrary body. An empirical equation is used to determine base drag. This subroutine is called with the statement

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

where SPEED = Mach number (I)  
AB = Base area (I)  
SREF = Reference area (I)  
CBD = Base drag coefficient (O)

12. BLOCK DATA is used to enter in tabular form the data from various tables and charts that are needed by other subroutines in the program.

13. Subroutine CDDR calculates the drag rise, 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 CDL1 to define the polar shape in the transonic region. CDDR has two entry points, on the first pass, the constants in the drag rise equation, the two limit Mach numbers, and their associated lift-curves are computed. Only the drag rise is computed in subsequent passes. Other subroutines called are: CLWBT, FDRG, LNTP, and WDRG. This subroutine is called with the statement

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

where CL = Lift coefficient  
XMACH = Mach number  
RNOFT = Reynolds number per unit length  
CDR = Drag rise

14. Subroutine CDL1 calculates the constants which are used by subroutine CDL2 to determine the drag polar. Other subroutines called are: ADCL, DLNT, KGIN, and LNTP. This subroutine is called with the statement

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

where SPEED = Mach number (I)  
RNOFT = Reynolds No. per unit length (I)  
FK = Polar shape factor below polar break (O)  
DELCL = Polar lift displacement (O)  
PRIMEK = Additional drag factor for drag polar above polar break (O)  
AKD = Theoretical drag-due-to-lift factor (O)  
AKB = Separation drag factor used to calculate drag polar above separation lift coefficient (O)

15. Subroutine CDL2 uses the polar shape factors determined by subroutine CDL1 and the polar break and separation lift coefficients determined by subroutine CLBRK to compute the drag due to lift. This subroutine is called with the statement

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 (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)

16. Subroutine CDRG 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. Subroutine CDRG is called with the statement

CALL CDRG (SPEED, AEROK, DELCL, CDC)

where SPEED = Mach number (I)  
AEROK = Polar shape factor (I)  
DELCL = Polar lift displacement (I)  
CDC = Camber drag (0)

17. Subroutine CDWN calculates the nose wave drag of body and nacelle components. Subroutine DLNT is also called. This subroutine is called with the statement

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

where AMAX = Maximum cross-sectional area (I)  
XLNOS = Length of nose (I)  
RIN = Radius of inlet area (I)  
BETA =  $\sqrt{M^2 - 1}$   
CDW = Wave drag of component based on maximum cross-sectional area (0)

18. Subroutine CDWT calculates the boattail wave drag of body and nacelle components. Subroutine LNTP is also used. Subroutine CDWT is called with the statement

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

where AMAX = Maximum cross-sectional area (I)  
XLAFT = Length of boattail (I)  
REX = Exit or base area (I)  
BETA =  $\sqrt{M^2 - 1}$   
CDW = Wave drag of component based on maximum cross-sectional area (O)

19. Subroutine CDWW calculates the wave drag for component represented as surfaces. It is called with the statement

CALL CDWW (CDOSR)

where CDOSR = wing wave drag based on the configuration reference area

20. Subroutine CFEQ 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. This subroutine is called with the statement

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

where RNOFT = Reynolds number per unit length  
ZMACH = Mach number  
CBAR = Length  
XTR = Distance along CBAR where transition occurs  
CF = Skin friction coefficient

21. Subroutine CFK3 calculates the equivalent, flat-plate friction drag coefficient for a component with nuclear damage. The surface condition can be described by defining the area and the corresponding roughness height for as many as three different regions on each component. It is called with the statement

CALL CFK3 (LENGTH, RNOFT, XOL1, XOL2, K1, K2, K3, XMACH, CF)

where LENGTH = Component length  
 RNOFT = Reynolds number per unit length  
 XOL1 = Fraction of length where regions 1 and 2 merge  
 XOL2 = Fraction of length where regions 2 and 3 merge  
 K1 = Roughness height of region 1  
 K2 = Roughness height of region 2  
 K3 = Roughness height of region 3  
 XMACH = Mach number  
 CF = Equivalent friction coefficient

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

CALL CLBRK (SPEED, RE, RNOFT)

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

The output, which is entered in COMMON CALC, is defined by

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

23. Subroutine 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. Other subroutines that are called are AALO, ADJUST, AER2, and TAIL. Output is entered into COMMON CALC. This subroutine is called with the statement

CALL CLWBT (SPEED)

where SPEED = Mach number



24. Subroutine CMOW computes the moment at zero-lift for the wing-body configuration. Subroutine TLNT is also used. CMOW is called with the statement

CALL CMOW (SPEED, CMO)

where SPEED = Mach number (I)  
CMO = Wing-body  $C_{m_0}$

25. Subroutine CPUOV 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 CALC, and were defined in subroutine CPZT. This subroutine is called by the statement

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,  $x/c = \frac{1}{2}(1 - \cos(\frac{IV}{32}))$  (I)  
CPI = Incompressible  $C_p$  (0)  
CP = Compressible value of  $C_p$  (0)  
XM = Mach number (I)

26. Subroutine CPZT 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. The output, which consists of a series of critical Mach number and lift combinations, are entered into COMMON CALC. Subroutines CPUOV, LNTP, and SECT are also used. CPZT is called with the statement

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

where ID = Type identification of the airfoil (I)  
XMACH = Mach number for compressible Cp 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)

27. Subroutine DINPT is called only if damage calculations are to be made. DINPT reads the damage input and prints out the values as a check that the data are correctly entered in the problem input. This subroutine is called with the statement

CALL OVERLAY (4HOVLY, 1, 2)

28. Function DLNT is a two-dimensional, nth-order Lagrangian interpolation procedure. Function DLNT also uses Subroutine LNTP. It is called with the statement

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 interpola-  
tions: 1 for step, 2 for linear, 3 for  
parabolic, 4 for cubic, etc. (I)

29. Subroutine DMIN controls the sequence of calculations necessary to compute minimum drag. It uses subroutines ADJUST, BDRG, FDRG, and WDRG. It is called with the statement

CALL DMIN (SPEED, RNOFT, CDMIN)

where SPEED = Mach number (I)  
RNOFT = Reynolds number per unit length (I)  
CDMIN = Minimum drag (O)

30. Subroutine ELEVON computes the lift, moment, and drag effectiveness of a wing trailing-edge flap control surface (elevon). The elevon is used as a trim surface if no horizontal tail or canard is specified in the input geometry. The computed parameters for the elevon are stored in COMMON OUTPUT and latter used in subroutine TDRG for trim calculations. Subroutine ELEVON uses subroutine DLNT and is called with the statement

CALL ELEVON(SPEED), where SPEED is the Mach number

31. Subroutine FDRG calculates friction, form and interference drag for all the components on the airplane. All output is entered into COMMON OUTPUT. Other subroutines called are CFEQ, FFACT, and IFACT. It is called with the statement

CALL FDRG (SPEED, RNOFT)

where SPEED = Mach number  
RNOFT = Reynolds number per unit length

32. Subroutine FFACT computes the form factors for each component. It is called with the statement

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

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

33. Subroutine EFFQ calculates the ratio of local dynamic pressure to freestream dynamic pressure. It is called with the statement

CALL EFFQ (RNOFT, XI, XMACH, HEIGHT, QEFF)

where RNOFT = Reynolds number per unit length  
XI = Distance from nose of component to station at which local dynamic pressure is to be calculated.  
  
XMACH = Mach number  
HEIGHT = Distance above surface at which local dynamic pressure is to be calculated  
QEFF = Local to freestream dynamic pressure,  $q/q$

34. Program GEOM uses input to compute additional geometric parameters that are required by other subroutines. The values computed for individual bodies and surfaces are entered into COMMON INPUT. Computed parameters which apply to the total airplane are inserted into COMMON CALC. This program is called with the statement

CALL OVERLAY (4HOVLY, 2, 0)

35. Subroutine IFACT computes the interference factors for each component. It also uses subroutine DLNT and is called with the statement

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

where ID = Identification of 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 (0)

36. Program INPT reads the geometry of the undamaged configuration and the aerodynamic conditions at which the problem is to be run. It is called with the statement

CALL OVERLAY (4HOVLY, 1, 1)

37. Subroutine KGIN computes the polar shape factor and polar displacement for a drag polar with drag rise. 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  $M_{L2}$ . It uses subroutines CDDR and LSPCF. It is called with the statement

CALL KGIN (CLDB, AKIN, DECLIN, SPEED, AKOUT, DCLOUT)

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)

38. Subroutine LNTP is a one-dimensional, nth-order Lagrangian interpolation procedure. It uses no other subroutines and is called with the statement

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)

39. Subroutine LSPCF is a least squares, polynomial curve fit subroutine. It is called by the statement

CALL LSPCF (X, Y, N, MP, SA, RE, SIGMA, IT)

where X The independent variable array  
Y The dependent variable array  
N The number of points  
MP The degree of polynomial fit  
SA The array containing the coefficients

IT        An error indicator  
          =0 if the coefficients were found correctly  
          ≠0 if the coefficients were not found correctly

SIGMA    An unbiased estimate of the standard deviation  
RE        The array containing the residuals

40.        Program MCRIT constructs a table of critical Mach number versus  $C_L$  from either an input table or by using an empirical method. It uses subroutines ADJUST, CPZT, and LNTP. MCRIT is called with the statement

CALL OVERLAY (4HOVLY, 3, 2)

41.        Program MRIT (MAIN PROGRAM) controls the logic of the calling of the four primary overlays. The lengths of the principal COMMON blocks used in all four overlays are also specified in this program. The primary overlays are called with the following statements.

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

42.        Subroutine NTRIM computes the lift and drag increments required to trim the damaged aircraft pitching moments. It also computes the rolling moments that result from wing or tail damage and makes a rough estimate of the drag that results from trimming these moments. NTRIM is called from subroutine NUCDAM by the statement

CALL NTRIM(J)

where J = Number of the survey condition at which data are being evaluated

43.        Program NUCDAM computes the effects of damage on minimum drag, and equations are included to account for any of the fourteen different modes. Data describing the damage are stored under the variable names DBOD(I,J), DSUR(I,J), and DWING(I). These variables are included in COMMON INPUT. Minimum drag increments

computed by this subroutine are stored in the variable DAMCD (I,J), which is included in COMMON NUCOUT. It calls subroutines WRITE, NUCDAM 2, NTRIM, and WRITE 2. It is called with the statement

```
CALL OVERLAY (4HOVLY, 4, 0)
```

44. Subroutine NUCDAM2 computes the effects of damage on the lift-curve slope, polar shape factor, and pitching moment curve. It then computes, for the untrimmed configuration, the lift, drag, and pitching moment. It is called by the statement

```
CALL NUCDAM2(J)
```

where J = Number of the survey condition at which data are being evaluated.

45. Subroutine RINPT is used to make minor changes to the basic problem input after the initial problem has been run. This provision allows additional problems to be run with ease if changes are minor. It is called with the statement

```
CALL OVERLAY (4HOVLY, 1, 3)
```

46. Subroutine SECT calculates the thickness and camber airfoil ordinates which are used in subroutine CPZT to calculate pressure distributions. SECT can calculate the section data for the standard NACA 6-series and 4-digit airfoils and the boconvex 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. It uses subroutine LNTP and is called by the statement

```
CALL SECT (ID, TOC, CLD)
```

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

47. Subroutine SETUP places initial values of 0.0 in the input and geometry common blocks. This is done to prevent random data from being used in program calculations if input data does not specify each value. It is called with the statement

```
CALL SETUP
```

48. Program 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 programs AEROA and AEROB are called to obtain the trimmed lift, moment and drag, SURVEY then prints out the aerodynamic results. It is called with the statement

```
CALL OVERLAY (4HOVLY, 3, 0)
```

The following secondary overlays are called from SURVEY:

```
OVERLAY(4HOVLY, 3, 1)      (VGEOM)
OVERLAY(4HOVLY, 3, 2)      (MCRIT)
OVERLAY(4HOVLY, 3, 3)      (AEROA)
OVERLAY(4HOVLY, 3, 4)      (AEROB)
```

49. 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. Subroutines AER2 and LNTP are also used. The output is entered in COMMON CALC. TAIL is called with the statement

```
CALL TAIL (SPEED)
```

where SPEED = Mach number (I)  
CLAT = Lift-curve slope contribution of the tail (0)  
A = Trim drag factor (0)  
B = Trim drag factor (0)  
AOH = Angle of zero lift of the tail (0)  
CLDH = Change in lift due to tail deflection factor (0)  
DEDA = Change in downwash per change in angle of attack (0)



50. Subroutine TDRG calculates moment using the wing-body  $C_{m_0}$  and aerodynamic center and the tail lift and moment arm. The moment is calculated at zero horizontal tail setting. The tail deflection required to trim and the resulting lift and drag increments are also computed. TDRG is called with the statement

CALL TDRG (DCLT, DCDT)

Where DCLT = Increment in lift due to trim  
DCDT Increment in drag due to trim

51. Subroutine TLNT is a triple-linear interpolation procedure. It uses subroutines DLNT and LNTP. It is called with the statement

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)

52. 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. It is called with the statement

CALL OVERLAY (4HOVLY, 3, 1)

53. Subroutine WBAC computes the wing-body aerodynamic center location. It first 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. Subroutines ACCR, DLNT and LNTP are used. It is called with the statement

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

54. Subroutine WDRG calculates the wave drag for all the components on the airplane and enters the results into COMMON OUTPUT. The following subroutines are used: CDWH, CDWT, and CDWW. It is called with the statement

CALL WDRG (FMACH)

where FMACH = Mach number

55. Subroutine WRITE prints the minimum drag increments calculated by subroutine NUCDAM. The total drag increment on each component is shown and the total increment due to each mode of damage is also shown. It is called from subroutine NUCDAM with the statement

CALL WRITE (J)

Where J = Number of the survey condition at which data are being evaluated.

56. Subroutine WRITE2 prints a summary of the aerodynamic data that are calculated by the AAT computer code. Lift and drag data are shown for both the undamaged and damaged configurations. It is called from subroutine NUCDAM by the statement

CALL WRITE2 (J)

where J = Number of the survey condition at which data are being evaluated.

57. Subroutine XINPT reads the card in the input deck that directs the program to evaluate a damaged configuration, change parameters in the basic input, or end the problem. XINPT interprets the message on the card and calls the appropriate subroutines required to comply with the directive. It is called with the statement

CALL OVERLAY (4HOVLY, 1, 0)

### 4.3 COMMON BLOCK DESCRIPTIONS

Common blocks in the AAT procedure have been utilized in a manner that helps the programmer keep track of variables that are used. Most of the data used internally are contained in one of the following common blocks:

1. Common block INPUT contains all input data.
2. Common block CALC contains geometric parameters that are computed by the program.
3. Common block OUTPUT contains the aerodynamic parameters that are computed by the program.
4. Common block NUCOUT contains the aerodynamic data that are computed for the damaged configuration.

Table 4-1 lists the name of each variable in the common block INPUT as it is used in the programming statements. The position of each variable within the common block is indicated, and a brief description of each variable is given. Similar information for the OUTPUT, CALC, and NUCOUT common blocks is given in Tables 4-2 through 4-4.

TABLE 1

## Variables in Common Block Input

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Input Data Block #1 (General Information)		
(1)	NBODYS	Number of Bodies
(2)	NNAC	Number of Nacelles
(3)	NPNLS	Number of Panels on Main Wing
(4)	NHT	Number of other lifting surfaces
(5)	NVT	Number of Non-Lifting Surfaces
(6)	ISWP	Indicator for Sweep
(7)	IREF	Indicator for Reference Alpha
(8)	IWNG	Indicator for Wing Planform Definition
(9)	NAFO	Number of airfoil ordinates
(10)	METER	Indicator for metric units
(11)	SREF	Reference Area
(12)	CMAC**	Reference Aerodynamic chord
(13)	XMAC**	:-Station of leading edge of CMAC
(14)	ZCG	Z-Station of CG
(15)	TWIST	Wing twist
(16)	ROUGHK	Surface Roughness Height (inches)
(17)	FMISC	Percent friction drag for misc. items
(18)	NPODS	NBODYS + NNAC
(19)	SPLAN	Planform Area
(20)	TAPR	Tip chord/root chord
(21)	SWP	Wing leading-edge sweep
(22)	ITRIM	Trim Indicator
(23)	RLE	Leading-Edge radius
(24)	TOCR	Reference t/c for input airfoil (NAFO > 0)
(25)	CLDR	Reference CLD for input airfoil (NAFO > 0)
(26)	TECH	Technology Factor
(27)	CONCL	Conical Camber
(28)	UPFUS	Aft Fuselage Upsweep Angle
(29)	AOB	Aft Fuselage width-to-height ratio

(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Input Data Block #2 (Body Geometry)		
(29+I)	BNAME (I)	Name of Body I, I=1, 7
(36+I)	BOD (I,1)	Length
(43+I)	(I,2)	Width
(50+I)	(I,3)	Height
(57+I)	(I,4)**	Wetted area, Total for Type I Bodies
(64+I)	(I,5)	Interference Factor
(71+I)	(I,6)	Number of bodies of Type I
(78+I)	(I,7)**	Maximum cross-sectional area
(85+I)	(I,8)	Base streamtube area
(92+I)	(I,9)	Nose length
(99+I)	(I,10)	Boattail length
(106+I)	(I,11)	Base drag area
(113+I)	I,12)	Inlet area
(120+I)	(I,13)	Vacant
(127+I)	(I,14)*	Fineness ratio
(134+I)	(I,15)	Vacant (Reserved for future expansion)
(169+I)	(I,20)	
(Input Data Block #3 (Surface Geometry)		
(176+I)	SNAME(I)	Name of surface I, I=1,7
(183+I)	SUR (I,1)	Airfoil section type
(190+I)	(I,2)	Airfoil 2-D camber
(197+I)	(I,3)	Thickness ratio, t/c
(204+I)	(I,4)	Leading-edge sweep
(211+I)	(I,5)*	x/c for maximum t/c
(218+I)	(I,6)	Vacant
(225+I)	(I,7)**	Wetted Area (Total)
(232+I)	(I,8)	Exposed root chord
(239+I)	(I,9)	Tip chord
(246+I)	(I,10)	Exposed Semi-span
(253+I)	(I,11)	X-station at leading edge of root chord
(260+I)	(I,12)	Y-station at leading edge of root chord
(267+I)	(I,13)	Z-station at leading edge of root chord
(274+I)	(I,14)	Incidence
(281+I)	(I,15)	Vacant
(288+I)	(I,16)	Vacant

(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(Input Data Block #3 (Surface Geometry), Continued)		
(295+I)	SUR (I,17)*	Exposed taper ratio
(302+I)	(I,18)*	Exposed planform area (Total)
(309+I)	(I,19)*	Aspect ratio
(316+I)	(I,20)*	Characteristic length
(323+I)	(I,21)*	Sweep of quarter-chord
(330+I)	(I,22)*	Sweep of mid-chord
(337+I)	(I,23)*	Sweep of trailing edge
(344+I)	(I,24)*	Sweep of maximum thickness
(351+I)	(I,25)	Vacant, Reserved for future expansion
⋮	⋮	⋮
(386+I)	(I,30)	

## Input Data Block #4 (Arbitrary Airfoil)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(394-423)	AFX(I)	$\pi/c$ for airfoil input, I = 1, 30
(424-453)	AFC(I)	Camber distribution
(455-483)	AFT(I)	Thickness distribution

## Input Data Block #5 (Variable Sweep)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(484)	XPIVOT	X-station of wing pivot
(485)	YPIVOT	Y-station of wing pivot
(486)		Vacant
(487)	AFTSW	Maximum aft sweep of leading edge
(488)	AFTCB	MAC of movable panel in aft position
(489)	AFTOC	t/c of movable panel in aft position
(490)	AFTAW	$A_{wet}$ of movable panel in aft position

## Input Data Block #6 (Survey Conditions)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(491)	NSURV	Number of surveys
(492)	NCLAS	Number of evenly spaced $C_L$ 's
(492+I)	FMSURV(I)	Mach numbers for surveys
(512+I)	ALT(I)	Altitude for surveys
(532+I)	CG(I)	X-station of cg for surveys, fraction of MAC
(552+I)	SWPV(I)	Leading-edge sweep for surveys
(572+I)	CLLO(I)	Low $C_L$ for surveys
(592+I)	CLHI(I)	High $C_L$ for surveys
(612+I)		Vacant

I = 1, 20

(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
Input Data Block #7 (Adjustment Factors)		
(632+I)	IVAL(I)	Indicator for parameters to be adjusted, I=1,20
(653)	NXVAR	Number of Mach numbers in table of Mach function adjustment factors
(654)	NADJ	Number of parameters to be adjusted as a function of Mach number
(655)	NXCL	Number of $C_L$ values in the table of lift function adjustment factors
(656)	NADJ2	Indicator for $M_{CR}$
(656+I)	X(I)	Mach numbers for the table of Mach adjustment factors I=1,15
(672-806)	YM(I,J)	Multiplier factors, I=15, J=9
(807-941)	YA(I,J)	Adder factors, I=15, J=9
(941+I)	XCL(I)	$C_L$ values for the table of lift function adjustment factors, I=1,15
Input Data Block #8		
<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(957)		Reserved for future expansion
⋮		
(1641)		
Input Data Block #9 (Damage Mode Indicators)		
<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(1641+I)	IDAM(I)	Indicator to select damage modes, I=1,17
Input Data Block #10 (Body Damage Parameters)		
<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(1661+I)	DBOD(I,1)	Roughness factor before damage on Body I, I=1,17
(1668+I)	(I,2)	Roughness factor of damaged area on Body I, I=1,17
(1675+I)	(I,3)	x/l where damage starts
(1682+I)	(I,4)	x/l where damage ends
(1689+I)	(I,5)	Fraction of area affected by damage
(1697-1731)	(I,6)-(I,10)	Vacant
(1731+I)	(I,11)	Number of forward-facing steps on Body I, I=1,7
(1738+I)	(I,12)	Width of each step
(1745+I)	(I,13)	Height of each step
(1752+I)	(I,14)	x/l at first step
(1759+I)	(I,15)	x/l at last step
(1767-1801)	√ (I,16)-(I,20)	Vacant



(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
	(Input Data Block #10 (Body Damage Parameters), Continued)	
(1801+I)	DBOD(I,21)	Number of aft-facing steps on Body I, I=1,7
(1808+I)	(I,22)	Width of each step
(1815+I)	(I,23)	Height of each step
(1822+I)	(I,24)	x/1 at first step
(1829+I)	(I,25)	x/1 at last step
(1837-1871)	(I,26)-(I,30)	Vacant
(1871+I)	(I,31)	Total number of holes in Body I, I=1,7
(1878+I)	(I,32)	Number of these holes over wing
(1885+I)	(I,33)	Length of each hole
(1892+I)	(I,34)	Width of each hole
(1899+I)	(I,35)	Depth of each hole
(1906+I)	(I,36)	x/1 where first hole starts
(1913+I)	(I,37)	x/1 where last hole starts
(1921 + I)	(I,38)	Type hole
(1928-1941)	(I,39)-(I,40)	Vacant
(1941+I)	(I,41)	Total number of waves in Body I, I=1,7
(1948+I)	(I,42)	Number of these waves over wing
(1955+I)	(I,43)	Length of each wave
(1962+I)	(I,44)	Width of each wave
(1969+I)	(I,45)	Amplitude of each wave
(1976+I)	(I,46)	x/1 where first wave starts
(1983+I)	(I,47)	x/1 where last wave starts
(1990 -2001)	(I,48)-(I,50)	Vacant
(2001+I)	(I,51)	Number of protuberances on Body I, I=1,7
(2018+I)	(I,52)	Height of each protuberance
(2025+I)	(I,53)	Parasite area of each protuberance, $\Delta f$
(2032+I)	(I,54)	x/1 of first protuberance
(2039+I)	(I,55)	x/1 of last protuberance.
(2047-2081)	(I,56) - (I,60)	Vacant

(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(Input Data Block #10 (Body Damage Parameters), Continued)		
(2081+I)	DBOD(I,61)	Frontal area of blunted body at point of damage
(2088+I)	(I,62)	l/d at point of damage
(2096-2151)	(I,63)-(I,70)	Vacant
(2152-2361)	∇ (I,71)-(I,100)	Reserved for additional modes of body damage

## Input Data Block #11 (Surface Damage Parameters)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(2361+I)	DSUR(I,1)	Roughness factor before damage on surface I, I=1,7
(2368+I)	(I,2)	Roughness factor after damage on surface I, I=1,7
(2375+I)	(I,3)	x/c where damage starts
(2382+I)	(I,4)	x/c where damage ends
(2389+I)	(I,5)	Fraction of area affected by damage
(2396+I)	(I,6)	Lower surface (0) or upper surface (1)
(2404-2431)	(I,7)-(I,10)	Vacant
(2431+I)	(I,11)	Number of forward-facing steps on surface I, lower surface
(2438+I)	(I,12)	Number of forward facing steps on surface I, upper surface
(2445+I)	(I,13)	Width of each step
(2452+I)	(I,14)	Height of each step
(2459+I)	(I,15)	x/c at first step
(2466+I)	(I,16)	x/c at last step
(2474-2571)	(I,17)-(I,20)	Vacant
(2501+I)	(I,21)	No. of aft-facing steps on surface I, lower surface
(2508+I)	(I,22)	No. of aft facing steps on surface I, upper surface
(2515+I)	(I,23)	Width of steps
(2522+I)	(I,24)	Height of steps
(2529+I)	(I,25)	x/c at first step
(2536+I)	(I,26)	x/c at last step
(2544-2571)	∇ (I,27)-(I,30)	Vacant

(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(Input Data Block #11 (Surface Damage Parameters), Continued)		
(2571+I)	DSUR(I,31)	No. of holes on surface I, lower surface
(2578+I)	(I,32)	No. of holes on surface I, upper surface
(2585+I)	(I,33)	Length of each hole
(2592+I)	(I,34)	Width of each hole
(2599+I)	(I,35)	Depth of each hole
(2606+I)	(I,36)	$x/c$ where first hole starts
(2613+I)	(I,37)	$x/c$ where last hole starts
(2620+I)	(I,38)	Type hole
(2627+I)	(I,39)	Porosity factor
(2634+I)	(I,40)	Vacant
(2641+I)	(I,41)	No. of waves on surface I, lower surface
(2648+I)	(I,42)	No. of waves on surface I, upper surface
(2655+I)	(I,43)	Length of each wave
(2662+I)	(I,44)	Width of each wave
(2669+I)	(I,45)	Amplitude of each wave
(2676+I)	(I,46)	$x/c$ where first wave starts
(2683+I)	(I,47)	$x/c$ where last wave starts
(2691-2711)	(I,48) - (I,50)	Vacant
(2711+I)	(I,51)	No. of protuberances on surface I, I=1,7
(2718+I)	(I,52)	Height of each protuberance
(2725+I)	(I,53)	Parasite area of each protuberance, $\Delta f$
(2732+I)	(I,54)	$x/c$ of first protuberance
(2739+I)	(I,55)	$x/c$ of last protuberance
(2747-2781)	(I,56) - (I,60)	Vacant
(2785-3061)	Y (I,61) - (I,100)	Reserved for additional modes of surface damage
(3062)	DWING (1)	$\Delta c/c \sim$ chord of missing leading edge
(3063)	(2)	$r_1 \sim$ inboard edge of missing leading edge
(3064)	(3)	$\Delta r \sim$ span of missing leading edge
(3065)	(4)	$z/t \sim$ Ratio of leading-edge to maximum thickness
(3066)	(5)	corner sharpness factor for leading edge
(3067)	(6)	left, right, or both sides, leading edge
(3068)	Y (7)	$\Delta c/c \sim$ chord of missing trailing edge

(Table 1 Continued)

<u>A( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(Input Data Block #11 (Surface Damage Parameters), Continued)		
(3069)	DWING(I,8)	$\eta_1$ ~ inboard edge of missing trailing edge
(3070)	(9)	$\Delta\eta$ ~ span of missing trailing edge
(3071)	(10)	left, right, or both sides, trailing edge
(3072)	(11)	$\Delta\eta$ ~ span of missing wing tip, left
(3073)	(12)	$\Delta\eta$ ~ span of missing wing tip, right
(3074)	(13)	moment arm of roll trimming device
(3075)	(14)	Fraction of area lost from left H.T.
(3076)	(15)	Fraction of area lost from right H.T.
(3077)	(16)	Fraction of area lost from other surfaces
(3078)	(17)	↓
(3079)	(18)	
(3080)	(19)	
(3081)	(20)	

\*Items marked with an asterisk are computed internally. But, since they logically group with the input variables, they are stored as shown in Table I.

\*\*Optional Input. If zero is input for these parameters, the value to be used will be computed internally.

TABLE 2  
Variables In Common Block OUTPUT

B( )	VARIABLE	DEFINITION
(1-21)	CLTAB(21)	Table of $C_L$
(22-42)	TABMCR(21)	Table of Mach Critical
(43)	CL	$C_L$ Value for Survey
(44)	CD	$C_D$ at $C_L$
(45)	CM	$C_M$ at $C_L$
(46)	ALPHA	$\alpha$ at $C_L$
(47)	CDM	$C_{D_{min}}$
(48)	CDL	Drag Due to Lift
(49)	CDR	Drag Rise at Lift
(50)	CDRO	Drag Rise at Zero Lift
(51)	CLT	Trimmed $C_L$ at $\alpha$ Corresponding to $C_L$
(52)	CDT	Trimmed $C_D$ at $\alpha$ Corresponding to $C_L$
(53)	DH	Trim Deflection (Horizontal)
(54)	FK	Polar Shape Factor
(55)	DELCL	Polar Displacement
(56)	CMO	Wing-Body Moment at Zero Lift
(57)	DCMCL	$(dC_M/dC_L)$ Wing-Body
(58)	XACWB	A.C. Location-Wing-Body
(59)	CLA	Lift Curve Slope
(60)	ALO	Zero-Lift $\alpha$
(61)	R	Leading-Edge Suction
(62)	CMOH	$C_M$ at $DH=0$
(63)	XH	Moment Arm Length From CG to $\bar{c}/4$ Horizontal Tail
(64)	OMEGA	Angle of Line Between CG and HT A.C.
(65)	XCT	Fuselage Station of CG Location
(66)	EO	Span Efficiency of Wing-Body, $R=1$
(67)	BMCRB	Body Mach Critical
(68)	CLDE	$C_{L_{\delta e}}$
(69)	AKD	Elevon Drag Factor
(70)	XCP	Elevon Center-of-Pressure
(71)	DE	Elevon Deflection
(72-94)		Vacant
(95)	DEDA	$d\epsilon/d\alpha$
(96)	CLAW	Wing $C_{L_{\alpha}}$
(97)	CLAB	Body $C_{L_{\alpha}}$
(98)	CLAT	Tail $C_{L_{\alpha}}$
(99)	AH	Tail Drag Factor

(TABLE 2, Continued)

<u>B( )</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
(100)	BH	Tail Drag Factor
(101)	AOH	Tail Incidence for Zero Lift
(102)	CH	Tail Drag Factor
(103)	ABREAK	Tail Drag Factor
(104)	CLDH	$C_{L\delta}$
(105)	CLPB	Polar Break Lift Coefficient
(106)	CLDB	Drag Break (Polar)
(107)	CLMAX	$C_{LMAX}$
(108)	ABRK	$\alpha$ at CLPB
(109)	AMAX	$\alpha$ at $C_{LMAX}$
(110)	DAMAX	$\Delta \alpha$ max
(111)	DEL	Lift Curve Parameter
(112)	CLS	Lift Curve Parameter
(113)	ARLO	Aspect Ratio Boundary
(114)	FMCRO	Mach Critical at Zero Lift
(115)	FML1	Mach Limit 1
(116)	FML2	Mach Limit 2
(117)	CDC	Camber Drag
(118)	CLAMCR	$C_{L\alpha}$ at $M_{CRO}$
(119)	CLAML2	$C_{L\alpha}$ at $M_{L2}$
(120)	CDMCR	Wing Drag at $M_{CRO}$
(121)	CDML2	Wing Drag at $M_{L2}$
(122)	CDATM	Wing Drag at M
(123-129)		Vacant
(130)	XACS	Aerodynamic Center at Stall
(131)	CDO	$C_D$ at Zero Lift
(132)	A2	Drag Rise Parameter
(133)	A3	Drag Rise Parameter
(134)	PL	Drag Rise Parameter
(135)	CDMISC	Miscellaneous Drag
(136)		Vacant
(137)	CDAFT	Aft End Upsweep Drag
(138-142)		Vacant
(143)	ALT	Survey Condition
(144)	SPEED	Survey Condition
(145)	SWEET	Survey Condition

(TABLE 2, Continued)

<u>B( )</u>	<u>VARIABLE</u>	<u>DEFINITION</u>
(146)	JPASS	Program Control Parameter
(147)		Vacant
(148)	RNOFT	RN/FT
(149-153)	TOTCD(5)	Drag Table - Total
(154-188)	CDSUR(I,J)	Drag Table - Surfaces
(189-223)	CDBOD(I,J)	Drag Table - Bodies

TABLE 3  
Variables in Common Block CALC

C( )	VARIABLE NAME	DESCRIPTION
(1)	BO2	Wing Semi-Span (b/2)
(2)	DOB	Body Dia./Wing Span (d/b)
(3)	CR	Wing Root Chord at Centerline
(4)	SPLAN	Theoretical Wing Planform Area
(5)	XCRTE	X-Sta. of Theor. Centerline Root Chord (CR) Trailing Edge
(6)	SWPQC	Equivalent Wing Quarter-Chord Sweep
(7)	SWPMC	Equivalent Wing Mid-Chord Sweep
(8)	SWPTE	Equivalent Wing Trailing-Edge Sweep
(9)	SWMT	Equivalent Wing Max. Thickness Sweep
(10)	SEXW	Equivalent Wing Exposed Area
(11)	CBAR	Equivalent Wing Exposed MAC
(12)	TOCW	Equivalent Wing Thickness Ratio
(13)	CLD	Equivalent Wing 2-D:Design $C_L$
(14)	ARXR	Equivalent Wing Exposed Aspect Ratio
(15)	CRX	Chord Length at Wing Root (Exposed)
(16)	CTX	Chord Length at Wing Tip
(17)	XRX	X-Station at LE of Wing Root
(18)	XTX	X-Station at LE of Wing Tip
(19)	YRX	Y-Station at Wing Root
(20)	YTX	Y-Station at Wing Tip
(21)	XB	X-Station of LE of Wing MAC
(22)	YB	Y-Station of LE of Wing MAC
(23)	SWET	Outboard Panel $A_{wet}$
(24)		Vacant
(25)	SWPLE	Equivalent-Wing LE Sweep
(26)	FOC	Max. Camber Ordinate
(27)	SWPR	Variable-Sweep Angle (Outboard Panel)
(28)	KPASS	Program Control Indicators
(29)	AR	$b^2/S_{plan}$
(30)	TR	$\lambda$ (Equivalent Taper Ratio)
(31)	YIX	Span of Inboard Wing
(32)	SIX	Exposed Area of Inboard Wing
(33)	ARI	Aspect Ratio of Inboard Wing
(34)	CBXP	Root Chord of Equivalent Outboard Wing
(35)	SOXP	Exposed Area of Equivalent Outboard Wing
(36)	AROP	Aspect Ratio of Equivalent Outboard Wing
(37)	DXQC	Equivalent Wing Parameter



(TABLE 3, Continued)

<u>C( )</u>	<u>VARIABLE NAME</u>	<u>DESCRIPTION</u>
(38)	TWIST	Wing Twist
(39)	WINC	Winc Incidence
(40)	XHT	X-Station for HT $\frac{c}{4}$
(41)	TOCS	t/c for Swept Panel
(42)	CBAR2	Exposed CBAR for Swept Panel
(43)	CLDS	CLD for Swept Panel
(44)	YOX	Span of Outboard Swept Panel
(45)	SWPMCS	Mid Chord Sweep of Outboard Swept Panel
(46)	DA1	Correlation Factor for Variable Sweep
(47)	DAC1	Correlation Factor for Variable Sweep
(48)	DA2	Correlation Factor for Variable Sweep
(49)	DAC2	Correlation Factor for Variable Sweep
(50)	DTOC	Correlation Factor for Variable Sweep

TABLE 4  
Variables in Common Block NUCOUT

D( )	VARIABLE	DESCRIPTION
(1-119)	DAMCD(I,K)	$\Delta C_{D_{min}}$ on Each Component Due to Damage I=Body or Surface No. I=1,7 K=Mode No. K=1,17
(120-238)	SYM(I,K)	Indicates Symmetrical Damage if SYM (I,J)=1.0 I=Body or Surface No. I=1,7 K=Mode No. K=1,17
(239)	DCDMIN	Total $\Delta C_{D_{min}}$ Due to Damage
(239+I)	ALP(I)	Undamaged Aircraft $\alpha$ I=1,21
(260+I)	CL(I)	Undamaged Aircraft $C_L$
(281+I)	CD(I)	Undamaged Aircraft $C_D$
(302+I)	CM(I)	Undamaged Aircraft $C_M$
(323+I)	CLT(I)	Undamaged Aircraft Trimmed $C_L$
(344+I)	CDT(I)	Undamaged Aircraft Trimmed $C_D$
(365+I)	CLP(I)	Damaged Aircraft $C_L$
(386+I)	CDP(I)	Damaged Aircraft $C_D$
(407+I)	CMP(I)	Damaged Aircraft $C_M$
(428+I)	CLTP(I)	Damaged Aircraft Trimmed $C_L$
(449+I)	CDTP(I)	Damaged Aircraft Trimmed $C_D$
(470+I)	RCLA(I)	Increment in $C_L$ That Produces Rolling Moment, Caused by: I=1 Missing LE I=2 Missing TE I=3 Missing Left Wing Tip I=4 Missing Right Wing Tip I=5 Missing Left H.T. I=6 Missing Right H.T.
(477)	DXAC	Shift in Wing-Body A.C. ( $\Delta X/C$ )
(477+I)	CROLL(I)	Rolling Moment Coefficient Caused by Asymmetric Wing or Tail Damage. Corresponds to ALP(I), I=1,21.
(498+I)	CDRT(I)	Drag Due to Trimming Out the Rolling Moment, I=1,21.

#### 4.4 PROGRAM LISTING

A complete listing of the program is shown in this section. The subprograms and subroutines appear in the order in which they occur within the program.

```

CVERLAY (CVLY,C,0)
PROGRAM MPTI(INPUT=514,OUTPUT=514,TAPE5=INPUT,TAPE7=CUTPLT,
1 TAPE10=514,TAPE31)
C
C      EMPIRICAL AIRCRAFT AERODYNAMIC PREDICTION PROCEDURE
C
COMMON /INPUT/  A(2001)
COMMON /CUTPLT/ R(223)
COMMON /CALC/   C(C)
COMMON /PLKDAT1/ F(1411)
COMMON /PLKDAT2/ F(90F)
COMMON /PLKTIL/ TITLE(4)
COMMON /PLKPRI/ KPRINT(50)
COMMON /PLKQV1/ IJ
C
15 IJ = 0
WRITE (6,2001)
2001 FORMAT(5X,15H ENTER INPUT )
20 CALL CVERLAY(4HCVLY,1,C)
C
IF (IJ.EC.2) GO TO 30
C
WRITE (6,2002)
2002 FORMAT(5X,15H ENTER GEOMETRY)
CALL CVERLAY(4HCVLY,2,0)
C
WRITE (6,2003)
2003 FORMAT(5X,15H ENTER SURVEY )
CALL CVERLAY(4HCVLY,3,0)
C
GO TO 20
C
30 WRITE (6,2004)
2004 FORMAT(5X,15H ENTER NUCCAP )
CALL CVERLAY(4HCVLY,4,C)
C
REWIND 10
PEAC (10)
PEAC (10) B
GO TO 20
C
END

```



```

C      FUNCTION DLNT (XRAR,YRAR,X,Y,F,NX,NY,NYMAX,LCY,LCY)
C      DOUBLE LAGRANGE INTERPOLATION
C      DIMENSION X(1),Y(1),F(NX*Y,1),FT(25)
C
C      FRAR=C.0
C      IF(LCY.GT.25) LRY=25
C      LLX=LCX
C      IF(LCY.GT.NX) LLX=NX
C      IF(NY.GT.1) GO TO 10
C      CALL LNTP (XRAR,FRAR,X,F(1,1),NX,LLX)
C      GO TO 30
C
C 10  LLY=LCY
C      IF(LCY.GT.NY) LLY=NY
C      DO 20 J=1,NY
C      CALL LNTP (XRAR,FT(J),X,F(1,J),NX,LLX)
C 20  CONTINUE
C      CALL LNTP (YRAR,FRAR,Y,FT,NY,LLY)
C
C 30  DLN1=FRAR
C
C      RETURN
C      END

```

```

C      SUBROUTINE TLNT(XPAF, YPAF, ZPAF, FBAR, X, Y, Z, F, NX, NY, NZ, NXMAX, NYMAX)
C          TRIPLE LINEAR INTERPOLATOR
C      DIMENSION X(1), Y(1), Z(1), F(NXMAX, NYMAX, 1), FT(25)
C      DO 10 I=1, NZ
C 10 FT(I) = CLNT(XBAR, YBAR, X, Y, F(1, 1, I), NX, NY, NXMAX, 2, 2)
C      CALL LNTP(ZBAR, FBAR, Z, FT, NZ, 2)
C      RETURN
C      END

```





```

SA(6)=(ST(2)-2.*ST(4)+5.*ST(6))/XMAX(1)
SA(7)=ST(1)-ST(3)+ST(5)-ST(7)
26 DO 27 I=1,N
SA(1)=64.*ST(1)/XMAX(7)
SA(2)=32.*ST(2)/XMAX(6)
SA(3)=(16.*ST(3)-112.*ST(5))/XMAX(5)
SA(4)=(8.*ST(4)-48.*ST(6))/XMAX(4)
SA(5)=(4.*ST(4)-20.*ST(6)+56.*ST(8))/XMAX(3)
SA(6)=(2.*ST(3)-8.*ST(5)+16.*ST(7))/XMAX(2)
SA(7)=(ST(2)-3.*ST(4)+5.*ST(6)-7.*ST(8))/XMAX(1)
SA(8)=ST(1)-ST(3)+ST(5)-ST(7)
27 DO 29 K=1,N
SIM=SA(I)
DO 28 I=1,M
IIM=I-I
SIM=SIM+SA(IIM)*X(K)**I
28 CONTINUE
RE(K)=SIM-Y(K)
29 CONTINUE
SIG=ABS(RE(1))
DO 30 J=2,N
30 SIG=SIG+ABS(RE(J))
DEAN=FLOAT(N)
SIGMA=(SIG*1.7724538)/(DEAN*1.414214)
RETURN

```

```

C
100 CONTINUE
N = IJ
M = 1
DET = 0.0

```

```

C
C
C
C
C
C

```

```

SOLUTION OF MATRIX A*X = B
N NC. OF ROWS AND COLUMNS IN MATRIX A
M NC. OF COLUMNS IN MATRIX B
IA NC. OF ROWS IN ARRAY A
IP NC. OF ROWS IN ARRAY P
DET SCALED VALUE OF DETERMINANT
IFR = 0 FOR NC FROF, = 1 FOR SINGULAR MATRIX
DO 240 I=1,N
AMAX = A(I,1)
DO 210 J=1,N
IF (ABS(A(I,J)) .GT. ABS(AMAX)) AMAX = A(I,J)
210 CONTINUE
DET = DET * AMAX
IF (AMAX .EQ. 0) GO TO 180
DO 220 J=1,N
A(I,J) = A(I,J) / AMAX
DO 230 L=1,M
P(I,L) = P(I,L) / AMAX
240 CONTINUE
NT = N-1
IF (NT) 50,50,60
50 IFR = 0
GO TO 16
60 DO 140 I=1,NT
K6 = I + 1
K = I
AMAX = A(J,I)
DO 70 J=K6,N
IF (ABS(A(J,I)) .LT. ABS(AMAX)) GO TO 70
AMAX = A(J,I)
K=J
70 CONTINUE
DET = DET * AMAX
IF (K .EQ. J) GO TO 200
DET = -DET
DO 80 J=I,N
SAV = A(I,J)
A(I,J) = A(K,J)
A(K,J) = SAV
80 CONTINUE
DO 90 L=1,M
SAV = P(K,L)
P(K,L) = P(I,L)
P(I,L) = SAV
90 CONTINUE
200 DO 130 K=K6,N

```

```

IF (A(I,I) .EQ. 0.) GO TO 180
IF (A(K,I) .EQ. 0.) GO TO 130
ERAS = -A(K,I) / A(I,I)
DC 110 J=K6,N
110 A(K,J) = ERAS * A(I,J) + A(K,J)
DC 120 L=1,M
120 R(K,L) = ERAS * B(I,L) + B(K,L)
130 CONTINUE
140 CONTINUE
DET = DET * A(N,N)
IF (A(N,N) .EQ. 0.) GO TO 180
NX = N
DC 160 L=1,M
150 B(NX,L) = B(NX,L) / A(NX,NX)
160 IF (NX .LE. 1) GO TO 50
NT = NX-1
DC 170 J=1,NT
DC 170 L=1,M
170 B(J,L) = B(J,L) - B(NY,L) * A(J,NX)
NX = NX - 1
GO TO 150
180 IER = 1
GO TO 16
END

```

```

C      SUBROUTINE TDRG(DCLT,DCDT)
C      COMPUTES EFFECT OF TRIM
C      COMMON /INPUT/ A(2081)
C      COMMON /CALC/ C(50)
C      COMMON /OUTPUT/ CLTAP(21), TAPMCR(21),
1      CL, CD, CM, ALPHA, CDM, CCL, CDP, CDFD, CLT, CDT,
2      CF, FH, FELCL, CMC, FCFCL, XACWP, CLD, ALD,
3      R, CMFH, XH, CMEGA, XCG, EC, FCFCE, CLDE, AYD, XCF, DE,
4      Bi(24),
5      CIAK, CLAG, CLAT, AH, AF, HSTAP, CH, AFFFAK, CLFH,
6      CLPE, CLDP, CLMAX, C2(7), FML1, FML2, CDC,
7      C2(12), XACS, C4(A), CDAPT, C5(5),
      ALT, SPEC, SWEEP, JPASS, P(77)
C
C      EQUIVALENCE (XLF1,A(254)), (CMAC,A(12)), (CFX,C(15)),
1      (ITRIM,A(22)), (NHT,A(4))
C
C      DCLT = 0.0
C      DCDT = 0.0
C
C      CGCCR = (XCG - XLF1)/CRX
C      YAC = XACWP
C
C      A.C. SHIFTS TO POSITION FOR STALLED WING ABOVE CLDE
C
C      IF (CL.GT.CLDP) YAC = XACWP + (XACS - XACWE) * ((CL-CLDP)/
1      (CLMAX-CLDP))**2
C      IF (CL.GT.CLMAX) XAC = XACS
C      IF (SPEED.GE.1.0) YAC = XACWP
C      DCMCL = (CGCCR - XACWB) * CFX/CMAC
C      DH = 0.0
C
C      CLTAIL = CLAT * (ALPHA - HSTAP) + CLDI * DH
C      CLWB = CL - CLTAIL
C      YLT = XH * CDS(CMEGA - ALPHA/57.3) / CMAC
C
C      CM = CM0 + (CGCCR - YAC) * CRX/CMAC * CLDE - CLTAIL * YLT
C      CMQH = CM
C
C      IF (CLFH.EQ.0.0.DF.XLT.EQ.0.0) GO TO 60
C      IF (ITRIM.EQ.1) GO TO 60
C      DH = CM/(CLDH + XLT)
C      CM = 0.0
C
C      60 CONTINUE
C      DE = 0.0
C      IF (NHT.GT.0) GO TO 70
C      IF (ITRIM.EQ.1) GO TO 70
C      XCLDE = (CGCCR * CFX/CMAC - XCP) * CLDE
C      IF (XCLDE.EQ.(.0)) GO TO 70
C      DE = CM/YCLDE
C      CM = 0.0
C
C      70 CONTINUE
C      IF (DH.EQ.0.0) GO TO 100
C      DCLT = DH*CLDH
C
C      DCDT = AH*DH**2 + BH*DH*(ALPHA - HSTAP)
C      IF (ALPHA.GT.APFEAK) DCDT = DCDT + CH*DH*(ALPHA-APFEAK)
C
C      100 IF (DE.EQ.0.0) GO TO 200
C      DCLT = DCLT + CLDE * DE
C      DCDT = DCDT + AKD * DE**2
C
C      200 RETURN
C      END

```

```

C      SUBROUTINE AER2(SPEED, CLA)
C      LIFT CURVE SLOPE
C      COMMON /BLKCLA/ SPLAN, TCC, TAPER, ARWS, DMSTR, CLC, DM1, DGE,
1      COMMON /BLKPR1/ EFSL, SWPMC, SREF
C      COMMON /BLKPR2/ KPRINT(50)

C      CCSZ = CGS(SWPMC)
C      PI = 3.14159
C      TWCC = 1.0
C      IF( SPEED.LT.1.0 ) TWCC = 1. + EFSL/SCFT(1.-SPEED**2)

C      CSLBC = (10.0 + 0.91 * ARWS**3)/(10.0 + ARWS**3)
C      ZMSTR = CSURD + (1.0 - CSURD) * (1.0 - CCSZ)**2
C      ZMSTRO = ZMSTR
C      ZMSTR = ZMSTR + DMSTR
C      ZM1 = 1.0 - 2.0 * TCC * (ARWS**3/(4.0 + ARWS**3))
1      * CCSZ**1.5 * (1.0 + 1.5*CLC**1.5)
C      ZM10 = ZM1
C      IF( ZM10.GT.ZMSTRO ) ZM10 = ZMSTRO
C      ZM1 = ZM10 + DM1
C      IF( ZM1.GT.ZMSTR ) ZMSTR = ZM1
C      ZM2 = ZM1 + TCC + DM1*0.5
C      ZM30 = 1.0 + TCC
C      ZM3 = ZM30
C      IF( ZM2.GT.1.0 ) ZM3 = ZM2 + TCC

C      SIG1=0.0
C      SIG2=0.0
C      IF( ZM2.NE.ZM1 ) SIG1 = 0.5*((SPEED - ZM1)/(ZM2 - ZM1))
C      IF( ZM3.NE.ZM2 ) SIG2 = 0.5*(1.0 + (SPEED - ZM2)/(ZM3 - ZM2))

C      DTCC = 1.0/(4.4*ARWS*CCSZ**1.5 )
C      DTCC = (TCC - DTCC)/CCSZ
C      IF( DTCC.LT.0.0 ) DTCC = 0.0
C      ARDT = ARWS * DTCC
C      IF( DTCC.GT.0.07 ) DTCC = 0.07
C      IF( ARDT.GT.0.1 ) ARDT = 0.1
C      GAMMA = 9.0 * (DTCC/(1.0 + 0.5 * ARDT) )

C      GAMMAO = GAMMA
C      IF( ZM30.NE.ZM10 ) GAMMA = GAMMAO * (ZM3 - ZM1)/(ZM30 - ZM10)
C      IF( GAMMA.GT.GAMMAO ) GAMMA = GAMMAO

C      XF = (16.0 + 3.0*ARWS**2)/(16.0 + 5.0*ARWS**2)
C      XKP = (1.0 + DGB)*(1.0 - DGB)**XF
C      XKT = 1.0
1      IF(SPEED.GE.ZM1.AND.SPEED.LE.ZM2) XKT = 1.0 - (4.0*SIG1*(1.0
1      - SIG1)**2 * GAMMAO
1      IF(SPEED.GT.ZM2.AND.SPEED.LE.ZM3) XKT = 1.0 - (4.0*SIG2*(1.0
1      - SIG2)**2 * GAMMAO
5      CONTINUE

C      Y1 = 2.0 + 0.66667 * SQRT(TAPER) - TAPER**2
C      Y = (1.0 + PI * ARWS)/(3.0 + PI * ARWS) * Y1
C      TWCS = 1.0
C      IF( ZMSTR.LT.1.0 ) TWCS = 1. + EFSL/SCFT(1.-ZMSTR**2)

C      CLAO = (TWCS + PI * ARWS)/(TWCS + SQRT(TWCS + (1.0 -
1      CCSZ**1.3334) * (ARWS/(2.0*CCSZ))**2) )
1      IF( SPEED.GT.0.0 ) RELAP = (SPEED - ZMSTR) * (1.0 +
1      (ZMSTR/SPEED)**2) **2
C      Z1 = PI * ARWS / CLAO
C      Z1 = 3.0 * Z1 * ( Z1 - 1.0 ) * CCSZ**0.6667
C      Z = ZMSTR * CLAO + ARWS**2/Z1

C      IF(SPEED.GT.ZMSTR) GO TO 10

C      CLAR = (.0546311*TWCC*ARWS)/(TWCC + SQRT(TWCC + (1.0 -
1      CCSZ**1.3334 * (SPEED/ZMSTR)**2.667)*(ARWS/(2.*CCSZ))**2
2      ))
2      GO TO 20

```

```

10 CLAP = 1.0/(57.3 *((ZMSTR/SPEED)**2 /CLAG + RETAP/4.0) )
C
C
C
20 CONTINUE
CLA = CLAP * XKT * YKF * SPLAN/SPEF
C
C
25 IF( KPRINT(11).EQ.0 ) GO TO 30
WRITE(6,1000) SPEED, CLA, CLAP, XKT, YKF, SPLAN, SFEF,
1 ZMSTR, ZM1, ZM2, ZM3, CCSZ, CSURB, APDT,
2 ICC, TAPER, FRKS, DMSTR, CLC, F*1, ECR, TWCD, SWPMC
1000 FORMAT(10X+AER2 DUMP, 6F15.5 (19X,6F15.5) )
C
30 RETURN
END

```

BLOCK DATA

C

```

COMMON /BLKDAT1/XSWPL(11), YPMIN(11),
1 AEA(22), PFE(22), CCC(22), RDD(22), XTT(22),
2 RCT(12), ATSW(6), TPCT(4), FXAC1(216), FXAC2(216),
3 XFD4(5), YRD4(5), ZFD4(3), FBD4(5,5,3),
4 XFY0(6), YXFN(6), ZFY5(6,6),
5 XFDf(4), YFDf(4), ZFD6(6), FBD61(4,4,6),
6 FEFf2(4,4,6), FRFA3(4,4,6),
7 MAP, THANS, CY, AMAP(22), RMAP(11),
8 XIN13(10), YIN13(7), ZUT13(10,7), ZGUT14(10,7),
9 NYIN15, XIN15(6), IYIN15, YIN15(5),
10 FCUT15(5,5), FCUT16(5,5),
11 Y1(6), Y1(6), X9(8), YC(8), XAP(5), YDCL(5),
12 XSWPL(7), YTP(4), FEP35(7,4), FEP7(7,4),
13 XCLDP(7), YAKB(7)

```

C

```

COMMON /BLKDAT2/XTP(4), YC1(6), YC2(4), XSWPL(4), YA(4),
1 YP(4), YCY(8), XM(4), CTAP(8,4), CTAP(8,4),
2 XCLM(13), YDY(6), FCIM(13,6), XYC2(C),
3 YMAP(5), FDC(X(5,5), XCY1(C), YMT(4),
4 ZC1Y(6,4), XCY2(6), YFC(6), ZDC1M(6,6),
5 Z2DC1M(5,6),
6 XSP(5), YBYA(6), FFA(8,6), YAR(4), YCN(8),
7 FRVRFY(6,8), YANG(10), YFIC(7), FRA(10,7),
8 XC2(4), YAST(C), FDCM(6,C), XCT(6), YK(5), FDCM2(6,5),
9 XXCLM(5), FCLMXY(9,6),
10 XY(12), XY(7), XF(12,7)

```

C

```

COMMON /BLKDAT3/XPACH3(17), COMK3(17),
1 XLCM2(6), HX3(5), CCC2(6,5),
2 XFC3(6), CLO3(4), XF3(6,4),
3 YPACH3(12), CPF3(13),
4 ZPACH3(13), CPA3(13),
5 XLOH4(12), HXY4(7), CDO4(12,7),
6 RATTG3(8), LMY(10), FFA(3(8,10),
7 XLEP(6), VMACH(6), FRAG4(6,6),
8 DETA1(6), AR1(5), CLA1(6,5),
9 DETA2(6), SWCC2(5), CLA2(6,5),
10 ETA13(6), CLA3(6),
11 DETA4(6), AR4(5), CLA4(6,5),
12 DETA5(6), SWCC5(5), CLA5(6,5),
13 FTA16(6), CLA6(6),
14 DEGR(6), DCOC(6), CLA(6,6)

```

C

```

COMMON /BLKDAT4/DETA11(6), AR11(5), YAC1(6,5),
1 DETA12(6), SWCC12(5), XAC2(6,5),
2 DETA113(6), YAC3(6),
3 DETA14(6), AR14(5), YAC4(6,5),
4 DETA15(6), SWCC15(5), YAC5(6,5),
5 FTA116(6), XAC6(6),
6 DCOC7(5), XAC7(5),
7 DCOC8(5), XAC8(5),
8 ,FLP1(6), CLH1(6),
9 FLP2(6), FFH2(6),
10 ZAR(10), YCLA(10)

```

C

```

COMMON /BLKDAT5/SPAN(6), YTPP(4), FKE(6,4), FKD(6,4),
1 SPAN2(11), YTPR2(5), FKS(11,5),
2 SPAN3(6), YTPP3(4), FKM(6,4),
3 SPAN4(5), YFP1(5), FKA(5,5),
4 SPAN5(5), YFP12(3), FKF(5,3)

```

C

C

```

DATA XSWPL / 0.0, .17453, .34906, .5236, .69813, .87266, 1.04719,
1 1.13446, 1.22173, 1.34300, 1.57074 /,
2 YPMIN / 0.53, 0.53, 0.52, 0.515, 0.5(5), 0.49, 0.45,
3 0.39, 0.30, 0.12, 0.10 /

```

C

```

DATA ROT / 0.0, .4, .8, 1.2, 1.6, 1.8, 2., 2.4, 2.8, 3.2, 3.6, 4. /,
1 ATSW / 1., 2., 3., 4., 5., 6. /,
2 TPPT / 0., 0.2, 0.25, 0.33, 0.5, 1.0 /

```

C

```

FIG. 4.1.4.2-22 DATCON *****
DATA FXAC1/.25, .24, .22, .22, .20, .16, .17, .25, .37, .37, .40, .42,
1 7*0.335, .39, .44, .40, .495, .5,

```



4 .8,3\*.72,.8,.77,.78,.792,.8,.795,.792,.75,.8,3\*.7F,  
 5 .98,.84,.63,.82,.9F,3\*.02,.06,.925,2\*.915,.9F,3\*.9F5 /,  
 6 .26,.35,.365,.38,.2F,.365,.42,.45,.26,.40,2\*.45F,.26,3\*.475,  
 7 .5,.35,.35,.75,.78,.795,.81,.78,2\*.6,.75,3\*.82,  
 8 .97,.905,.895,.8F,.98,3\*.6,.55,.95,2\*.645,3\*.66,  
 9 1.17,1.06,1.03,1.,1.17,1.1,1.08,1.065,1.17,1.09,2\*1.065,1.17,  
 0 3\*1.06 /

C

1 DATA XIN13/ 0.0,1., 2., 3., 4., 5., 6., 7., 8., 20. /,  
 2 YIN13/ 0.0, 20.0, 35.0, 50.0, 65.0, 72.5, 90.C /,  
 3 ZCUT13/ .00973,.00973,.0093,.009,.0087,.0087,4\*.0086,  
 4 .00973,.00973,.0093,.009,.0087,.0087,4\*.0086,  
 5 .00973,.00973,.0093,.009,.0087,.0087,4\*.0086,  
 6 .00943,.00943,.0092,.009,.0087,.0087,4\*.0086,  
 7 .0093,.0093,.0088,.0087,.0086,4\*.0085,4\*.00844,  
 8 .0093,.0093,.0088,.0087,.0086,4\*.0085,4\*.00844 /  
 9 DATA ZCUT14/ 1.28,1.28,1.27,1.252,1.244,1.233,1.227,1.217,2\*1.21,  
 1 1.28,1.28,1.27,1.252,1.244,1.233,1.227,1.217,2\*1.21,  
 2 1.324,1.324,1.311,1.295,1.271,1.266,1.258,1.244,2\*1.23,  
 3 1.363,1.363,1.35,1.33,1.313,1.292,1.281,1.261,2\*1.25,  
 4 1.392,1.392,1.378,1.36,1.343,1.337,1.323,1.306,2\*1.29,  
 5 1.403,1.403,1.392,1.371,1.354,1.325,1.31,1.295,2\*1.28,  
 6 1.403,1.403,1.392,1.371,1.354,1.325,1.31,1.295,2\*1.28 /

C

1 DATA AFAP / 22.0, 21.7, 19.2, 18.35, 22.0, 21.2, 19.2, 27.0, 11.75,  
 2 BMAP / 4 \* 1.75, 3 \* 2.0, 4 \* 0.0 /

C

1 DATA NXIN15/ 6 /,  
 2 XIN15/ 0.0, 0.5, 0.6, 0.8, 0.9, 1.0 /,  
 3 NYIN15/ 5 /,  
 4 YIN15/ 0.0, 5.0, 15.0, 20.0, 23.0 /,  
 5 FCUT15/ 0.865, 0.756, 0.71, 0.6, 0.51, 0.45,  
 6 0.865, 0.755, 0.71, 0.6, 0.51, 0.45,  
 7 0.925, 0.845, 0.958, 0.905, 0.845, 0.78,  
 8 0.978, 1.02, 1.03, 1.02, 0.99, 0.975,  
 9 C.95, 1.04, 1.05, 1.05, 1.045, 1.04 /,  
 0 FCUT16/ 6 \* 1.05, 1.05, 1.04, 1.02, 0.94, 0.88, 0.84,  
 1 1.05, 1.05, 1.04, 1.01, 0.995, 0.985,  
 2 6 \* 1.05, 6 \* 1.05 /  
 3 DATA X1 / 0.7243, 0.8727, 1.0472, 1.2217, 1.3963, 1.5708 /,  
 4 Y1 / 0.85, 0.69, 0.575, 0.51, 0.483, 0.48 /,  
 5 X5 / 0.3765, 0.5236, 0.6081, 0.8727, 1.0472, 1.2217, 1.3963, 1.6 /,  
 6 Y9 / 0.65, 0.625, 0.435, 0.35, 0.375, 0.447, 0.476, 0.48 /,  
 7 XAR / 1.0, 2.3, 3.0, 4.0, 5.0 /,  
 8 YGCL / -0.16, 0.0, 0.05, 0.085, 0.0855 /

C

1 DATA AAA / 8.48243E-4, 8.03417E-4, 3.31222E-6, -1.72323E-3,  
 2 2.02978E-3, 1.36146E-4, 3.79206E-5, 15 \* 0.0 /,  
 3 BBB / -.0296429, -.0340676, -.0142049, 0.0310185,  
 4 -.07305, -4.05767E-3, -1.03545E-3, 15 \* 0.0 /,  
 5 CCC / 1.23335, 1.1599C, 1.03825, 0.585837, 1.57142, .712684,  
 6 .639364, 0.0, 0.0, 5 \* 1.1019, 3 \* .275475,  
 7 3 \* 3.3057, 0.0, 0.0 /,  
 8 DDD / -2.47002, -1.71448, -1.9292, -.559968, -2.99739,  
 9 .018291, .0607728, 15 \* 0.0 /,  
 0 XTT / .35, .375, .41, .45, .37, .39, .42, .371, .5, .2,  
 1 .3, .4, .5, .6, .3, .4, .5, .3, .4, .5, 0.0, 0.0 /  
 2 DATA XSWP / 0.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0 /,  
 3 YTP / 0.0, 0.25, 0.5, 1.0 /,  
 4 FEP35 / 0.94, 0.951, 0.96, 0.968, 0.973, 0.974, 0.971,  
 5 0.992, 0.999, 0.997, 0.999, 0.998, 0.992, 0.984,  
 6 1.0, 1.0, 0.999, 0.997, 0.998, 0.971, 0.944,  
 7 0.956, 0.92, 0.986, 0.975, 0.959, 0.93, 0.863 /,  
 8 FEP7 / 0.86, 0.607, 0.23, 0.936, 0.944, 0.948, 0.942,  
 9 0.90, 0.954, 0.96, 0.968, 0.961, 0.978, 0.956,  
 0 0.906, 0.962, 0.967, 0.968, 0.961, 0.924, 0.96,  
 1 0.986, 0.968, 0.942, 0.91, 0.877, 0.833, 0.79 /  
 2 DATA YCLDB / 0.0, 0.3, 0.4, 0.5, 0.6, 0.7, 2.0 /,  
 3 YAKB / 0.33, 0.352, 0.44, 0.625, 1.0, 1.47, 2.0 /

C

1 DATA XTR / 0.0, 0.1, 0.2, 0.3, 0.5, 1.0 /,  
 2 YC1 / 0.0, 0.225, 0.47, 0.5, 0.32, 0.0 /



2	YC2	/	0.0, 0.22, 0.5, 0.91, 1.05, C.65 /,
3	XSWP1	/	0.0, 3C.C, 5C.0, 60.0 /,
4	YA	/	0.9, 1.04, 1.2, 1.3 /,
5	YB	/	0.0, 0.24, 0.5, 0.71 /,
6	XCY	/	C.0, 2.C, 2.2, 2.5, 3.0, 4.0, 4.5, 10.0 /,
7	YM	/	C.2, C.4, C.6, 1.0 /,
8	DATA CTAR	/	0.0, -0.2, -0.22, -0.355, -0.41, -0.445, -0.45,
9			.12, .12, .36, .43, .47, .6, .72, .72,
1			.58, .58, .15, .24, .27, .62, .92 /,
2	DTAB	/	0.0, -0.05, -0.05, -0.07, -0.07, -0.07,
3			0.0, -0.05, -0.05, -0.07, -0.07, -0.07,
4	XXCLM	/	0.0, .2, .4, .6, .8, 1., 1.2, 1.6, 2.0, 2.4, 2.8,
5			3.2, 4.4 /,
6	YYDY	/	0.0, .25, .5, .75, 1.0, 10.C /,
7	DATA FCLMX	/	0.0, 1.375, 1.57, 1.645, 1.645, 1.575, 1.4, 1.14,
8			1.0, .6, .9, .91, .9, .75, 1.2, 1.4, 1.5,
9			1.5, 1.5, 1.33, 1.11, 1.0, .94, .9, .8, .75,
1			.68, 1.15, 1.4, 1.47, 1.475, 1.4, 1.27, 1.08,
2			.075, .02, .6, .5, .64, .6, 1.1, 1.3, 1.375,
3			1.3, 1.22, 1.23, 1.66, .6, .6, .84, .83,
4			.4, 1.02, 1.2, 1.26, 1.31, 1.27, 1.19, 1.04, .94,
5			.8, .64, .83, .82, .4, 1.02, 1.2, 1.29, 1.31,
6			1.27, 1.16, 1.02, .91, .85, .82, .8, .8,
7	XXC2	/	0.0, 2.0, 4., 6., 8., 10., 12., 14., 20. /,
8	YYMACH	/	0.0, 0.2, 0.4, C.6, 1.0 /,
9	FDCLMX	/	-.11, -.08, C.0, .115, .23, .335, .36, .22, 0.C,
1			-.11, -.08, C.0, .115, .23, .335, .36, .22, 0.C,
2			-.11, -.08, C.0, .1, .2, .3, .32, .18, 0.0,
3			-.11, -.08, C.0, .08, .15, .21, .205, .08, 10*0.0 /

C C C \*\* FIGURE 4.1.1.4-5, DATCOM \*\*\*\*\*

1	DATA XOY1	/	0.0, 1.1, 2.25, 2.5, 3.0, 3.5, 4., 4.5, 6. /,
2	YXMT	/	0.3, C.35, C.4, C.45 /,
3	ZC1MAX	/	0.8, C.8, 1.315, 1.43, 1.51, 1.55, 1.54, 1.47, 1.41,
4			0.8, C.8, 1.315, 1.43, 1.51, 1.52, 1.43, 1.41, 1.36,
5			0.8, C.8, 1.315, 1.39, 1.45, 1.45, 1.43, 1.35, 1.34,
6			0.8, 0.8, 1.3, 1.33, 1.33, 1.36, 1.33, 1.32, 1.32 /,
7	XOY2	/	0.0, 1.0, 2.0, 2.2, 2.4, 3.0, 3.2, 10.0 /,
8	YFCC	/	0.0, 2.0, 4.0, 6.0, 8.0, 10.C /,
9	ZDC1M	/	10*0.0, 0.1, 0.13, 0.15, 0.145, 0.05, 0.0,
1			0.0, C.11, C.275, C.3, 0.2, C.22, C.10, C.C,
2			C.1, C.26, C.44, C.45, C.45, C.30, C.17, C.0,
3			C.2, C.40, C.60, C.60, C.60, C.30, C.25, 0.0,
4			C.2, C.40, C.60, C.60, C.60, C.30, C.25, 0.0 /
5	DATA Z2DC1M	/	0.0, .15, .25, .27, .18, .07, .07,
6			.1, .28, .49, .47, .41, .26, .12, .12,
7			.2, .41, .61, .60, .55, .35, .17, .17,
8			.33, .52, .68, .65, .60, .30, .20, .20,
9			.33, .52, .68, .65, .60, .39, .20, .20 /

1	DATA XSP	/	0.0, 10.0, 20.0, 30.0, 40.0, 50.C, 60.0, 90.0 /,
2	YDYA	/	C.C, 1.2, 2.0, 3.0, 4.0, 50.C /,
3	FDA	/	1.8, 2.2, 3.4, 5.0, 7.4, 10.2, 13.4, 13.4,
4			1.8, 2.2, 3.4, 5.0, 7.4, 10.2, 13.4, 13.4,
5			C.1, 1.1, 2.3, 3.9, 5.9, 6.7, 5.9,
6			1.3, 1.7, 2.4, 3.2, 4.2, 5.3, 6.7, 6.7,
7			2.2, 2.0, 2.1, 2.3, 2.5, 2.8, 3.25, 3.25 /,
8			2.2, 2.0, 2.1, 2.3, 2.5, 2.8, 3.25, 3.25 /

1	DATA XAP	/	0.0, 1.0, 2.0, 3.0, 4.0, 6.0 /,
2	YCG	/	-1., -.75, -.5, -.25, 0.C, .25, .5, .75 /,
3	FKVGF	/	1.57, 1.74, 2.02, 2.30, 2.58, 2.75,
4			1.78, 1.95, 2.18, 2.52, 2.8, 2.90,
5			2.03, 2.21, 2.42, 2.71, 3.0, 3.15,
6			2.48, 2.60, 2.73, 2.93, 3.19, 3.33,
7			3.13, 3.15, 3.20, 3.27, 3.4, 3.60,
8			3.7, 3.8, 3.81, 3.82, 3.88, 3.98,
9			6 * 4.45, 6 * 5.0 /
1	DATA XANG	/	0.0, C.0, 1.396, .2094, .2792, .3490, .4186, .4886,
2			0.5584, C.6980 /,
3	YRIG	/	0.0, 0.03, C.05, C.08, 0.12, C.16, C.5 /,
4	FRA	/	0.4, 0.745, 0.75, 7*0.0,
5			1.0, .62, .40, .255, .19, .148, .11, .079, .05, C.C,
6			1.0, .84, .72, .48, .34, .252, .195, .142, .1, C.C,
7			1.0, .90, .88, .758, .618, .46, .345, .25, .17, C.C,
8			1.0, .617, .585, .514, .472, .42, .35, .23, 0.0,

8 1.0, .925, .895, .835, .76, .665, .565, .46, .35, 0.0,  
9 1.0, .925, .895, .835, .76, .665, .565, .46, .35, 0.0 /

C  
C

DATA XC2 / 0.0, 1.0, 2.0, 3.0, 4.0, 5.0 /,  
1 YAST / 0.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 9.0, 30.0 /,  
2 FDAM / 10, 7, 4, 1, 0, 5, 8, 6, 5, 2, 5, 5, -0.3, 5,  
3 7.5, 4, 1, -1, -1, 5, 5, 5, 1.5, -1.2, -2.5, -1.5,  
4 3, -1.5, -4.1, -3, -1, 5, 2, -4.5, -1, -2, -1.5,  
5 -2.3, -6.5, -5, -3, -1, 5, -4.5, -7, -5, -3, -1.5,  
6 -8, -7, -1, -3, -1, 0.5 /,  
7 XCT / 4.0, 6.0, 8.0, 10.0, 12.0, 14.0 /,  
8 YM / 0.2, 0.4, 0.6, 0.8, 1.0 /,  
9 FDAM2 / 0.1, 0.4, 0.7, 1.1, 1.5, 0, 0.3, 0.6, 0.9, 1.3, 1.5,  
1 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3,  
2 0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0 /

C

DATA XXXCLM / 0.0, .2, .4, .6, .8, 1.0, 1.2, 1.6, 2.0 /,  
1 FCLMXY / 0.7, 1.2, 1.27, 1.43, 1.43, 1.345, 1.25, 1.08, .99,  
2 .65, 1.12, 1.29, 1.355, 1.365, 1.305, 1.21, 1.07, .98,  
3 .6, 1.09, 1.23, 1.29, 1.295, 1.24, 1.17, 1.05, .97,  
4 .55, 1.0, 1.15, 1.23, 1.24, 1.20, 1.15, 1.04, .96,  
5 .5, .81, 1.06, 1.17, 1.20, 1.175, 1.13, 1.02, .935,  
6 .5, .91, 1.09, 1.17, 1.20, 1.165, 1.11, 1.0, .91 /

C

DATA XX / 0.0, .4, .8, 1.0, 1.2, 1.4, 1.6, 2.0, 2.4, 2.8, 3.2, 3.27 /,  
1 XY / 0, .1209, .1745, .2618, .2490, .4263, 1.0 /,  
2 XF / 0, 1.0, .915, .63, .29, 0.0,  
3 0, 1.0, .915, .63, .29, 0.0,  
4 0, 1.0, .915, .755, .52, .245, 0.0,  
5 1.0, .97, .95, .97, 1.0, .95, .87, .69, .552, .38, .185, 0.0,  
6 .92, .90, .90, .95, 1.0, .95, .845, .63, .48, .31, .150, 0.0,  
7 .70, .60, .64, .61, 1.0, .945, .83, .585, .425, .27, .128, 0.0,  
8 .79, .60, .64, .61, 1.0, .945, .83, .585, .425, .27, .128, 0.0 /

C

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\* DATA FOR DAMAGE EVALUATIONS \*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

C

COMPRESSIBILITY FACTOR, FIG 2-15, FZF-1800

C

DATA XMACH3 / 0.0, 0.2, 0.4, 0.6, 0.8, 0.9, 0.98, 1.0, 1.05,  
1 1.1, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4 /,  
2 CCMK3 / 1.00, 1.00, 1.03, 1.05, 1.20, 1.26, 1.95, 1.00, 1.15,  
3 1.38, 1.22, 1.04, 0.90, 0.76, 0.67, 0.60, 0.55 /

C

CDC FOR HOLES, FIGURE 2-16 FZF-1800

C

DATA XICM3 / 1.0, 2.0, 4.0, 6.0, 7.0, 10.0 /,  
1 HICM3 / 0.002, 0.010, 0.030, 0.060, 0.100 /,  
2 CDC3 / 0.035, 0.015, 0.048, 0.101, 0.110, 0.110,  
3 0.009, 0.022, 0.060, 0.175, 0.190, 0.190,  
4 0.015, 0.038, 0.126, 0.275, 0.300, 0.300,  
5 0.025, 0.060, 0.215, 0.470, 0.505, 0.505,  
6 0.040, 0.064, 0.330, 0.775, 0.820, 0.820 /

C

WING ROUGHNESS MAGNIFICATION FACTOR, FIG 2-14, FZF-1800

C

DATA XCC3 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
1 CLC3 / 0.0, 0.2, 0.4, 0.6 /,  
2 XM3 / 1.35, 1.55, 1.50, 1.30, 1.07, (.83,  
3 1.35, 1.55, 1.50, 1.30, 1.07, (.83,  
4 1.05, 1.75, 2.15, 1.83, 1.30, (.83,  
5 1.05, 1.75, 2.15, 1.83, 1.30, (.83 /

C

FORWARD-FACING STEP PRESSURE COEFFICIENTS, FIG 2-13, FZF-1800

C

DATA YMACH2 / 0.0, 0.4, 0.8, 0.9, 0.95, 1.0, 1.05,  
1 1.1, 1.2, 1.6, 2.0, 2.4, 2.8 /,  
2 CPF3 / 0.40, 0.44, 0.52, 0.56, 0.58, 0.61, 0.60, /,  
3 0.57, 0.51, 0.38, 0.27, 0.25, 0.25 /

C

C AFT-FACING STEP PRESSURE COEFFICIENTS; FIG 2-13, FZP-1800  
 C DATA ZMACH3 / 0.0, 0.4, 0.8, 0.9, 0.95, 1.0, 1.05,  
 1 1.1, 1.7, 1.6, 2.0, 2.4, 2.8 /,  
 2 CPA3 / -0.20, -0.20, -0.20, -0.21, -0.40, -0.53, -0.48,  
 3 -0.45, -0.34, -0.19, -0.14, -0.11, -0.10 /

C CAVED-IN PANELS, FIGURE 2-17, FZP-1800  
 C DATA X1CH4 / 10., 13., 15., 20., 25., 30., 40., 50., 60., 80., 100., 120. /,  
 1 HX4 / 0.002, 0.0005, 0.001, 0.002, 0.003, 0.005, 0.010, 0.015 /,  
 2 CCG4 / 0.0250, 0.0350, 0.0410, 0.0060, 0.0040, 0.0028, 0.0028, /,  
 3 0.0016, 0.0018, 0.0007, 0.0004, 0.0003, 0.0002,  
 4 0.0360, 0.0210, 0.0160, 0.0090, 0.0050, 0.0040,  
 5 0.0022, 0.0014, 0.0010, 0.0006, 0.0004, 0.0003,  
 6 0.0050, 0.0020, 0.0020, 0.0010, 0.0007, 0.0005,  
 7 0.0028, 0.0018, 0.0012, 0.0007, 0.0005, 0.0003,  
 8 0.0050, 0.0020, 0.0020, 0.0010, 0.0007, 0.0005,  
 9 0.0040, 0.0026, 0.0018, 0.0010, 0.0006, 0.0004,  
 10 0.0070, 0.0040, 0.0030, 0.0015, 0.0010, 0.0006,  
 11 0.0040, 0.0031, 0.0021, 0.0012, 0.0006, 0.0004,  
 12 0.0070, 0.0050, 0.0040, 0.0020, 0.0015, 0.0010,  
 13 0.0060, 0.0038, 0.0026, 0.0015, 0.0010, 0.0007,  
 14 0.1100, 0.0650, 0.0490, 0.0270, 0.0150, 0.0120,  
 15 0.0070, 0.0045, 0.0031, 0.0017, 0.0011, 0.0008 /

C EFFECT OF SEVERING NOSE, SPHERICAL SHAPE, NACA RM L52014A  
 C DATA DAT103 / 0.0, 0.2, 0.4, 0.6, 0.7, 0.8, 0.9, 1.0 /,  
 1 VMACH / 0.0, 0.2, 0.4, 0.5, 1.0, 1.1, 1.2, 1.6, 2.0, 3.0 /,  
 2 DRAG3 / 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.010, 0.040, /,  
 3 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.010, 0.040, /,  
 4 0.0, 0.0, 0.0, 0.005, 0.012, 0.020, 0.065, 0.150,  
 5 0.0, 0.0, 0.0, 0.010, 0.027, 0.036, 0.090, 0.220,  
 6 0.0, 0.0, 0.0, 0.028, 0.039, 0.050, 0.160, 0.295,  
 7 0.0, -0.010, 0.000, 0.024, 0.033, 0.020, 0.217, 0.330,  
 8 0.0, -0.010, 0.000, 0.054, 0.103, 0.153, 0.284, 0.410,  
 9 0.0, 0.010, 0.000, 0.155, 0.245, 0.358, 0.440,  
 10 0.0, 0.022, 0.010, 0.210, 0.313, 0.425, 0.522, 0.703,  
 11 0.0, 0.022, 0.000, 0.210, 0.313, 0.425, 0.522, 0.703 /

C EFFECT OF FLATTING SPHERICAL NOSE, NACA TN 4201 (PP-38)  
 C DATA XLFC / 0.0, 0.1, 0.2, 0.3, 0.4, 0.5 /,  
 1 VMACH / 0.0, 0.8, 0.8, 1.0, 1.7, 2.8 /,  
 2 DRAG4 / 0.70, 0.535, 0.375, 0.225, 0.098, 0.0,  
 3 0.695, 0.525, 0.365, 0.215, 0.090, 0.0,  
 4 0.545, 0.400, 0.341, 0.202, 0.083, 0.0,  
 5 0.667, 0.460, 0.313, 0.181, 0.075, 0.0,  
 6 0.667, 0.460, 0.313, 0.181, 0.075, 0.0 /

C EFFECT OF MISSING L.E. PANELS ON LIFT CURVE  
 C DATA DETA1 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 AR1 / 0.0, 3.0, 5.0, 7.0, 10.0 /,  
 2 CLA1 / 0.0, -0.0021, -0.0038, -0.0052, -0.0064, -0.0075,  
 3 0.0, -0.0021, -0.0038, -0.0052, -0.0064, -0.0075,  
 4 0.0, -0.0032, -0.0035, -0.0036, -0.0036, -0.0036,  
 5 0.0, -0.0039, -0.0070, -0.0066, -0.0120, -0.0135,  
 6 0.0, -0.0039, -0.0070, -0.0066, -0.0120, -0.0135 /

C DATA DETA2 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 SWPCC2 / -0.0, 0.0, 20.0, 40.0, 80.0 /,  
 2 CLA2 / 0.0, -0.0002, -0.0003, -0.0004, -0.0004, -0.0004,  
 3 0.0, -0.0002, -0.0003, -0.0004, -0.0004, -0.0004,  
 4 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 5 0.0, 0.0005, 0.0009, 0.0012, 0.0015, 0.0018,  
 6 0.0, 0.0005, 0.0009, 0.0012, 0.0015, 0.0018 /

C DATA FLA13 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 CLA3 / 0.0012, 0.0005, 0.0, 0.0006, 0.0024, 0.0042 /  
 C EFFECT OF MISSING T.E. PANELS ON LIFT CURVE

DATA DETA4 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 AR4 / 0.0, 3.0, 5.0, 7.0, 10.0 /,  
 2 CLA4 / 0.0, -0.020, -0.035, -0.050, -0.065, -0.080,  
 3 C.C, -0.020, -0.035, -0.050, -0.065, -0.080,  
 4 C.C, -0.030, -0.045, -0.060, -0.075, -0.090,  
 5 C.C, -0.037, -0.052, -0.067, -0.082, -0.097,  
 6 C.C, -0.037, -0.052, -0.067, -0.082, -0.097 /

DATA DETA5 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 SWFCC5 / -0.0, 0.0, 20.0, 40.0, 60.0 /,  
 2 CLA5 / 0.0, -0.007, -0.003, -0.004, -0.004, -0.005,  
 3 C.C, -0.002, -0.003, -0.004, -0.004, -0.005,  
 4 C.C, C.C, C.C, C.C, C.C, C.C,  
 5 C.C, C.CC07, C.CC12, C.CC15, C.CC17, C.CC19,  
 6 C.C, C.CC07, C.CC12, C.CC15, C.CC17, C.CC19 /

DATA ETA16 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 CLA6 / -0.0017, -0.0010, -0.0002, C.CC08, C.CC21, C.CC36 /

EFFECT OF CUTOUT CHORD LENGTH ON LIFT CURVE  
 FROM BRITISH DATA SHEET, VOL 2A, WINGS CL.01.04

DATA DPCF / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 CCCC / 0.0, 0.2, 0.3, 0.4, 0.5, C.F /,  
 2 CLACLA / 1.0, 1.045, 1.085, 1.112, 1.155, 1.185,  
 3 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,  
 4 1.0, 0.975, 0.950, 0.925, 0.900, 0.875,  
 5 1.0, 0.965, 0.940, 0.915, 0.890, 0.865,  
 6 1.0, C.820, C.850, C.870, C.890, C.910,  
 7 1.0, C.880, C.770, 0.650, 0.600, 0.550 /

EFFECT OF MISSING L.E. PANELS ON AERODYNAMIC CENTER

DATA DETA11 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 AF11 / 0.0, 3.0, 5.0, 7.0, 10.0 /,  
 2 XAC1 / 0.000, C.C38, C.C74, C.113, C.151, C.184,  
 3 0.000, C.C38, C.C74, C.113, C.153, C.184,  
 4 0.000, C.C30, C.C60, C.091, C.123, C.155,  
 5 C.CC0, C.C25, C.C51, C.C77, C.106, C.135,  
 6 0.000, C.C25, C.C51, C.C77, C.106, C.135 /

DATA DETA12 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 SWCC12 / -0.0, 0.0, 20.0, 40.0, 60.0 /,  
 2 XAC2 / 0.000, C.C03, C.C05, C.C08, C.C10, 0.013,  
 3 C.CC0, C.C03, C.C05, C.C08, C.C10, C.C13,  
 4 0.0, 0.0, C.C, C.C, C.C, C.C,  
 5 0.0, -0.003, -0.006, -0.010, -0.014, -0.020,  
 6 0.0, -0.003, -0.006, -0.010, -0.014, -0.020 /

DATA ETA13 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 XAC3 / 0.055, C.C30, C.C06, -0.015, -0.034, -0.050 /

EFFECT OF MISSING T.E. PANELS ON AERODYNAMIC CENTER

DATA DETA14 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 AP14 / 0.0, 3.0, 5.0, 7.0, 10.0 /,  
 2 XAC4 / 0.000, -0.011, -0.025, -0.027, -0.033, -0.037,  
 3 C.CC0, -0.011, -0.020, -0.027, -0.033, -0.037,  
 4 C.CC0, -0.012, -0.027, -0.031, -0.039, -0.045,  
 5 C.CC0, -0.014, -0.026, -0.036, -0.045, -0.052,  
 6 0.000, -0.014, -0.026, -0.036, -0.045, -0.052 /

DATA DETA15 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,  
 1 SWCC15 / -0.0, 0.0, 20.0, 40.0, 60.0 /,  
 2 XAC5 / 0.000, C.C05, C.C09, C.C13, C.C16, C.C19,  
 3 C.CC0, C.C05, C.C09, C.C13, C.C16, C.C19,  
 4 C.C, C.C, C.C, 0.0, C.C, C.C,  
 5 C.C, -0.008, -0.015, -0.021, -0.026, -0.030,  
 6 0.0, -0.008, -0.015, -0.021, -0.026, -0.030 /

DATA ETA16 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /,

```

1      XAC6      / 0.008, 0.004, 0.001, 0.002, 0.006, 0.012 /
C
DATA DCOC7 / 0.0, 0.2, 0.4, 0.6, 0.8 /
1      XAC7      / -0.031, 0.0, 0.025, 0.047, 0.064 /
C
DATA DCOC8 / 0.0, 0.2, 0.4, 0.6, 0.8 /
1      XAC8      / 0.017, 0.0, -0.014, -0.024, -0.030 /
C
EFFECT OF HOLES IN WING ON LIFT-CURVE AND POLAR SHAPE
C
DATA FLO1 / 0.0, 0.02, 0.04, 0.06, 0.08, 0.2 /
1      CLAM1 / 1.0, 0.981, 0.952, 0.910, 0.858, 0.758 /
C
DATA FLC2 / 0.0, 0.02, 0.04, 0.06, 0.08, 0.20 /
1      FPH2 / 1.0, 0.904, 0.759, 0.584, 0.404, 0.404 /
C
VORTEX LIFT INCREMENT FOR MISSING WING TIP (NUCCAM2)
C
DATA ZAR / 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 10.0 /
1      XCIA / 0.0285, 0.0250, 0.0175, 0.0104, 0.0040,
2      -0.0020, -0.0005, -0.0100, -0.0110, -0.0110 /

```

\*\*\*\*\*

DATA FOR ELEVON CALCULATIONS COMMON ELKDAT5

```

DATA SPAN / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /
1      YTPR / 0.0, 0.25, 0.5, 1.0 /
2      FKR / 0.0, 0.3, 0.555, 0.77, 0.925, 1.0,
3      0.0, 0.28, 0.53, 0.75, 0.915, 1.0,
4      0.0, 0.265, 0.51, 0.73, 0.9, 1.0,
5      0.0, 0.25, 0.48, 0.70, 0.885, 1.0 /
6      FKO / 0.0, 0.38, 0.64, 0.84, 0.965, 1.0,
7      0.0, 0.295, 0.54, 0.74, 0.9, 1.0,
8      0.0, 0.24, 0.475, 0.64, 0.855, 1.0,
9      0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /
C
DATA SPAN2 / 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 /
1      YTPR2 / 0.0, 0.2, 0.333, 0.5, 1.0 /
2      FKR2 / 0.0, 0.0245, 0.0435, 0.0400, 0.0480, 0.0417,
3      0.0320, 0.0212, 0.0112, 0.0035, 0.0, 0.0480, 0.0417,
4      0.0400, 0.0310, 0.0460, 0.0345, 0.0365, 0.0530,
5      0.0, 0.0350, 0.0240, 0.0120, 0.0,
6      0.0510, 0.0410, 0.0460, 0.0560, 0.0500, 0.0570,
7      0.0, 0.0270, 0.0450, 0.0560, 0.0410, 0.0600,
8      0.0345, 0.0450, 0.0230, 0.0110, 0.0,
9      0.0, 0.0225, 0.0400, 0.0375, 0.0600, 0.0620,
10     0.0600, 0.0520, 0.0400, 0.0220, 0.0 /
C
DATA SPAN3 / 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 /
1      YTPR3 / 0.0, 0.25, 0.5, 1.0 /
2      FKR3 / 0.0, 0.645, 1.045, 1.240, 1.220, 1.330,
3      0.0, 0.440, 0.740, 0.955, 1.070, 1.120,
4      0.0, 0.310, 0.575, 0.775, 0.930, 1.030,
5      0.0, 0.200, 0.400, 0.600, 0.800, 1.000 /
C
DATA SPAN4 / 0.0, 0.4, 0.8, 1.6, 2.4 /
1      YRFF1 / 0.0, 0.05, 0.1, 0.15, 0.2 /
2      FKA / 0.0, 0.38, 0.68, 1.00, 1.13,
3      0.0, 0.40, 0.74, 1.12, 1.32,
4      0.0, 0.40, 0.72, 1.38, 1.75,
5      0.0, 0.57, 1.05, 1.81, 2.43,
6      0.0, 0.70, 1.27, 2.24, 3.27 /
C
DATA SPAN5 / 0.2, 0.4, 0.6, 0.8, 1.0 /
1      YRFF2 / 0.0, 0.1, 0.2 /
2      FKF / 4.00, 1.60, 0.60, 0.17, 0.0,
3      3.40, 1.55, 0.65, 0.26, 0.12,
4      2.90, 1.46, 0.72, 0.40, 0.25 /

```

END

```

C      CVERLAY(1,0)
      PROGRAM XINPT
C
      COMMON/INPUT/  A(2001)
      COMMON /OUTPL1/ R(223)
      COMMON /CALC/   C(50)
      COMMON /BLKDAT1/ F(1411)
      COMMON /BLKDAT2/ F(906)
      COMMON /BLKTII/  TITLE(6)
      COMMON /BLKPE1/ KPRINT(50)
      COMMON /BLKCV1/ IJ
C
      DIMENSION V(4)
      DATA V /4HNEW, 4HDAMA, 4HCHAN, 4HEND /
C
      IF ( IJ.GE.1 ) GO TO 70
C
      20 CONTINUE
C
      READ(5,1000) (TITLE(I), I = 1,6 )
      IF (EOF(5).NE.C) GO TO 75
      WRITE(6,2000) (TITLE(I), I = 1,6 )
C
      READ(5,1001) (KPRINT(I), I=1,50)
C
      DO 50 I = 1, 50
      IF ( KPRINT(I).GT.C ) WRITE(6,2001) I, KPRINT(I)
C
      50 CONTINUE
      CALL SETUP
      PROGRAM INPT CALLED
      CALL CVERLAY(4HCVLY,1,1)
C
      IJ = 1
      GO TO 60
C
      70 READ(5,1002) WORD
      IF ( EOF(5).NE.C ) CALL EXIT
      IF ( WORD.EC.W(3) ) CALL CVERLAY(4HCVLY,1,3)
      IF ( WORD.FC.W(3) ) IJ = 1
      IF ( WORD.EC.W(2) ) CALL CVERLAY(4HCVLY,1,2)
      IF ( WORD.EC.W(2) ) IJ = 2
      IF ( WORD.EC.W(2).CP.WORD.EC.W(3) ) GO TO 80
      IF ( WORD.EC.W(4) ) GO TO 20
C
      WRITE(6,2002) WORD
      CALL EXIT
C
      75 IF (IJ.GE.1.AND.KPRINT(2).EC.1) CALL XCOPY4(2)
      CALL EXIT
C
      80 CONTINUE
C
      1000 FORMAT( 6A10 )
      1001 FORMAT( 50I1 )
      1002 FORMAT( A4 )
      2000 FORMAT(1H1,/////, 27X,*EMPIRICALLY BASED COMPUTER PROGRAM* //
      1 //
      2 //
      3 //
      4 //
      5 //
      27X,*TO PREDICT THE AERODYNAMIC CHARACTERISTICS*
      31X,*OF AIRCRAFT* //
      30X,*INCLUDING* //
      29X,*ROUGHNESS, PROTRUSIONS, AND* //
      34X,*DAMAGE CALCULATIONS* (R(1)), 27X, 6A10)
      2001 FORMAT(5X,*KPRINT(*,I2,*) = *,I2)
      2002 FORMAT(5X,*UNRECOGNIZED CONTROL WORD *,A4,* READ*)
C
      END

```

```
SUBROUTINE SETUP
C
C   ZEROS OUT INPUT COMMON BLOCKS
C
COMMON /INPUT/  A(2081)
COMMON /OUTPUT/ R(223)
COMMON /CALC/   C(50)
COMMON /BLKDAT1/ D(1411)
COMMON /BLKDAT2/ E(908)
C
DO 10 I= 1, 2081
A(I) = 0.C
10 CONTINUE
C
RETURN
END
```

OVERLAY(1,1)  
PROGRAM INPT

PROBLEM DATA INPUT

COMMON /INPUT/ NPCDS, NPALS, NHT, NVT, ISLP, IREF, IWAG, NAFD,  
METEF, SREF, CMAC, YCC, ZCC, TWIST,  
RCUGHK, FMISC, NPCDS, SPLAN, TAPP, SWP, ITRIM, RLF,  
TOCK, CLCP, TECH, CCNCL, LPPFUS, ACR,  
FMAME(7), PCF(7,10), CFCC, EI, EG, A(4),  
SMAME(7), SUR(7,30),  
AFY(30), AFC(30), AFT(30),  
XPJVT, YPIVNT, XAPFX, AFTSW, AFTCR, AFTCC, AFTAW,  
NSLOU, NCLES, FMSUR(20), ALT(20), CG(20),  
SWPV(20), CLIP(20), CLW(20), CHSV(20),  
IVAL(20), NYVAR, MAGJ, MXCL, MADJ2, X(15),  
YM(15,0), YA(15,0), XCL(15),  
MFLSPT, IFUSC, NCPCC, NPCPPT, XFLS, YFLS, ZFLS,  
PUSY(100), FUSC(100), FLSA(100),  
PGFY(6), PCY(6), FELZ(6), DELX(6,30), DELA(6,30),  
IPAM(20), DECF(7,100), CSLF(7,100), DVIING(20)

DIMENSION TIT(20), W(1), ASUR(7)

DATA TIT / 3H62-, 3H64-, 3H65-, 3H66-, 3H62A, 3H64A, 3H65A, 3H66A,  
3H-62, 3H-63, 3H-64, 3H-65, 3H-66, 3H-63, 3H-64,  
3H-65, 3H-63, 3H-64, 3H-95 /  
DATA W / 4HADJU /

CONFIGURATION DEFINITION

OPTIONS CARDS

100 READ(5,1001) NPCDS, NPALS, NHT, NVT, ISLP,  
IREF, IWAG, NAFD, METEF, ITRIM  
1 READ(5,1002) SREF, CMAC, YCC, ZCC, LPPFUS, ACR  
READ(5,1002) RCUGHK, FMISC, TWIST, CCNCL

WRITE(6,2001) NPCDS, SREF, RCUGHK  
WRITE(6,2002) NPALS, CMAC, FMISC  
WRITE(6,2003) NHT, YCC, TWIST  
WRITE(6,2004) NVT, ZCC, CCNCL  
WRITE(6,2005) ISLP, LPPFUS  
WRITE(6,2006) IREF, ACR  
WRITE(6,2007) IWAG, NAFD, METER, ITRIM  
IF (METER.EC.0) WRITE(6,4002)  
IF (METER.FO.1) WRITE(6,4003)  
IF (METER.FO.2) WRITE(6,4004)  
IF (IWAG.GT.0) READ(5,1002) SPLAN, TAPP, SWP  
IF (IWAG.GT.0) WRITE(6,4001) SPLAN, TAPP, SWP

IF (NHT.EO.0.AND.ITRIM.EC.0) READ(5,1002) CFCC, EI, ED  
FLSELAGF CFCEFTPY

IF (NPCDS.FO.0) GO TO 104  
IF (NPCDS.GT.7) CALL FYIT  
READ(5,1004) (NAME(I), I = 1, NPCDS)  
DO 101 J = 1, 12  
READ(5,1002) (BCD(I,J), J = 1, NPCDS)

101 CONTINUE  
105 WRITE(6,2009) (NAME(I), I = 1, NPCDS)  
WRITE(6,2010) (BCD(1,1), I = 1, NPCDS)  
WRITE(6,2011) (BCD(1,2), I = 1, NPCDS)  
WRITE(6,2012) (BCD(1,3), I = 1, NPCDS)  
WRITE(6,2013) (BCD(1,4), I = 1, NPCDS)  
WRITE(6,2014) (BCD(1,5), I = 1, NPCDS)  
WRITE(6,2015) (BCD(1,6), I = 1, NPCDS)  
WRITE(6,2016) (BCD(1,7), I = 1, NPCDS)  
WRITE(6,2017) (BCD(1,8), I = 1, NPCDS)  
WRITE(6,2018) (BCD(1,9), I = 1, NPCDS)  
WRITE(6,2019) (BCD(1,10), I = 1, NPCDS)  
106 WRITE(6,2020) (BCD(1,11), I = 1, NPCDS)  
WRITE(6,2021) (BCD(1,12), I = 1, NPCDS)

NO 110 I = 1, NPCDS



```

IF (RCD(I,12).GT.C.0) N = N + 1
110 CONTINUE
NNAC = N
NPODYS = NPODS - NNAC
C
C
C SURFACE GEOMETRY
104 NSUR = NPMLS + NHT + NVT
IF (NSUR.GT.7) CALL EXIT
PFAD(5,1004) (SNAME(I), I = 1, NSUR)
PFAD(5,1005) (ASUF(I), I = 1, NSUR)
GO TO 107
IF (J.EC.5.CR.J.EC.6) GO TO 107
READ (5,1002) (SLR(I,J), I = 1, NSUR)
107 CONTINUE
WRITE(6,2023) (SNAME(I), I = 1, NSUR)
WRITE(6,2024) (ASUF(I), I = 1, NSUR)
WRITE(6,2025) (SUR(I, 2), I = 1, NSUR)
WRITE(6,2026) (SUR(I, 3), I = 1, NSUR)
WRITE(6,2027) (SUR(I, 4), I = 1, NSUR)
WRITE(6,2028) (SUR(I, 7), I = 1, NSUR)
WRITE(6,2029) (SUR(I, 8), I = 1, NSUR)
WRITE(6,2030) (SUR(I, 9), I = 1, NSUR)
WRITE(6,2031) (SUR(I, 10), I = 1, NSUR)
WRITE(6,2032) (SUR(I, 11), I = 1, NSUR)
WRITE(6,2033) (SUR(I, 12), I = 1, NSUR)
WRITE(6,2034) (SUR(I, 13), I = 1, NSUR)
WRITE(6,2035) (SUR(I, 14), I = 1, NSUR)
C
C
C IF (I.WG.EC.C) SWP = SUR(NPMLS,4)
C
C
C AIPFGJL AND SWEEP OPTIONS
GO TO 109
109 IF (ASUR(I).EC.TTT(8)) GO TO 108.
GO TO 112
C
108 READ(5,1002) RLL, TGCR, CLDR, TECH
READ(5,1002) (AFY(I), I = 1, NAFC)
READ(5,1002) (AFC(I), I = 1, NAFC)
READ(5,1002) (AFT(I), I = 1, NAFC)
C
WRITE(6,2040) PLE, TGCR, CLDR, TECH
GO TO 111
111 WRITE(6,2041) AFX(I), AFC(I), AFT(I)
CONTINUE
C
C
C VARIABLE SWEEP DATA
112 IF (ISWP.NE.1) GO TO 113
READ(5,1002) XPIVOT, YPIVOT, AFTSW, AFTCB, AFTDC, AFTAW
C
C
C SURVEY CONTROL
113 READ (5,1001) NSURV, NCLAS
GO TO 114
114 I = 1, NSURV
READ(5,1002) (MSURV(I), ALT(I), CG(I), SWPV(I), CLLC(I),
1 CLHI(I))
IF (CLHI(I).EC.C.C) CLHI(I) = 1.0
IF (ISWF.EC.1) GO TO 114
IF (SWPV(I).FC.O.C.AND.SWPV(I).NE.SWP) SWPV(I) = SWP
114 CONTINUE
WRITE(6,2042) NSURV
GO TO 115
115 I = 1, NSURV
WRITE(6,2043) (MSURV(I), ALT(I), CG(I), SWPV(I), CLLC(I),
1 CLHI(I))
CONTINUE
C
C
C ADJUSTMENT FACTORS
116 READ(5,1006) WORD, WPD
IF (WORD.EC.W(1)) GO TO 117
GO TO 122
117 READ(5,1007) (IVAL(I), I = 1, 20), NXVAR, NACJ, NXCL, NAGJ2
IJ = 0
IF (NXVAR.FC.C) GO TO 110
READ(5,1002) (X(I), I = 1, NXVAR)

```



```

C
2001 FORMAT(1H1, 10X, * PROBLEM INPUT PARAMETERS *, //,
      2 5X, *NFCD5 =*, I8, 20X, *SREF =*, F10.2, 20X, *ROUGH=*,
      3 F10.5)
2002 FORMAT(5X, *NPALS =*, I8, 20X, *CMAC =*, F10.3, 20X, *FMISC =*, F10.2)
2003 FORMAT(5X, *NHT =*, I8, 20X, *XMAC =*, F10.3, 20X, *TWTST =*, F10.3)
2004 FORMAT(5X, *NVT =*, I8, 20X, *7CG =*, F10.3, 20X, *CCNCL =*, F10.3)
2005 FORMAT(5X, *ISWP =*, I8, 20X, *UPFLS =*, F10.3)
2006 FORMAT(5X, *ISEF =*, I8, 20X, *ACF =*, F10.3)
2007 FORMAT(5X, *IKAC =*, I8, //,
      1 5X, *MAFC =*, I8, //,
      2 5X, *METER =*, I8, //,
      3 5X, *ITFIN =*, I8, //)
4002 FORMAT(//, 10X, *DIMENSIONS ARE IN FEET *)
4003 FORMAT(//, 10X, *DIMENSIONS ARE IN METERS*)
4004 FORMAT(//, 10X, *DIMENSIONS ARE IN INCHES EXCEPT FOR *, //,
      1 10X, *DIFFERENCE, PLANE COIN, AND VETTER AREAS*, //,
      2 10X, *AND ALSO ALTITUDE AND PITCH ANGLES ALPHAS*)
4001 FORMAT(////, 2X, *DIFFERENCE VINC FOR THE PANEL CASE *, //,
      1 5X, *SPLAN =*, F0.3, 5X, *TAPP =*, F6.4, 5X, *SWP =*, F6.2)

```

```

C
2000 FORMAT(////, 20X, 7F10)
2010 FORMAT(//, 2X, *LENGTH =*, 7F10.3)
2011 FORMAT(//, 2X, *WIDTH =*, 7F10.3)
2012 FORMAT(//, 2X, *HEIGHT =*, 7F10.3)
2013 FORMAT(//, 2X, *WETTED AREA =*, 7F10.3)
2014 FORMAT(//, 2X, *INTER. FZICOR =*, 7F10.3)
2015 FORMAT(//, 2X, *NO. OF THIS TYPE =*, 7F10.3)
2016 FORMAT(//, 2X, *MAXIMUM AREA =*, 7F10.3)
2017 FORMAT(//, 2X, *BASE AREA =*, 7F10.3)
2018 FORMAT(//, 2X, *CASE LENGTH =*, 7F10.3)
2019 FORMAT(//, 2X, *CATTAIL LENGTH =*, 7F10.3)
2020 FORMAT(//, 2X, *BASE DRAG AREA =*, 7F10.3)
2021 FORMAT(//, 2X, *INLET AREA =*, 7F10.3)

```

```

C
2022 FORMAT(1H1, ///, 20X, 7A10)
2024 FORMAT(//, 2X, *SECTION =*, 7(7X, A3))
2025 FORMAT(//, 2X, *CAMBER (2-D) =*, 7F10.3)
2026 FORMAT(//, 2X, *THICKNESS RATIO =*, 7F10.3)
2027 FORMAT(//, 2X, *L.F. SWEEP =*, 7F10.3)
2030 FORMAT(//, 2X, *WETTED AREA =*, 7F10.3)
2031 FORMAT(//, 2X, *PCOT CHORD =*, 7F10.3)
2032 FORMAT(//, 2X, *TIP CHORD =*, 7F10.3)
2033 FORMAT(//, 2X, *SPAN OF PANEL =*, 7F10.3)
2034 FORMAT(//, 2X, *Y- LE RECT CHORD =*, 7F10.3)
2035 FORMAT(//, 2X, *Y- LE RECT CHORD =*, 7F10.3)
2036 FORMAT(//, 2X, *Z- LE RECT CHORD =*, 7F10.3)
2037 FORMAT(//, 2X, *JUNCTION =*, 7F10.3)

```

```

C
2040 FORMAT(///, 10X, * COORDINATES FOR INPUT AIRFOIL *, //,
      1 * GLE =*, F7.4, 5X, *TCCF =*, F7.4, 5X, *CLPC =*, F7.4, 5X,
      2 *TECH =*, F6.3, //,
      3 2X, * XDIST/CHORD CAMBER THICKNESS *)
2041 FORMAT(1X, 2F14.4)
2042 FORMAT(1H1, //, 10X, J5, * CRAC PLARS TO BE GENERATED AT THE*,
      1 * FOILCANG CONDITIONS *, //, 110, *VACH NC *,
      1 T20, *ALTITUDE *, T34, *C.G. *, T40, *SWEEP *, T50, *FFOM CL *,
      1 T60, *TC CL *, //)
2043 FORMAT(5X, F10.2, F13.2, F9.2, F7.1, F11.2, F11.2, 112)
2044 FORMAT(//, 4X, A4, A10, 10X, 5H*****, /1+1)
2045 FORMAT(//, 2X, *AIRFOIL SECTION SUR(*, I1, *, 1) = *, A3,
      1 * NOT FOUND ON LIST* ///)

```

```

C
END

```

CVEPLAY(1,2)  
PROGRAM DIRPT

DAMAGE PARAMETERS ARE READ IN

C  
C  
C  
C

```
COMMON /INPUT/ NPGDYS, NNAC, NPMLS, NHT, NVT, ISWP, IREF, JCDW, NAFC,  
1 METE, SPFE, CMAC, XCG, ZCG, EIA,  
2 PCLCLK, FMISC, IDANG, AP, TAPR, SWP, XOCMT, FLE,  
3 TDCR, CLFR, TECH, CLM(2),  
4 PRAME(7), PCC(7,20), SNAME(7), SLF(7,20),  
5 AFX(30), AFC(30), AFI(30),  
6 XPIVOT, YPIVOT, XAPFX, AFTSW, AFTCE, AFTDC, AFTAW,  
7 NSLSPV, NCLAS, FMSLSPV(20), AL1(20), CC(20),  
8 SWPV(20), CLIC(20), CLPI(20), DHSV(20),  
9 IVAL(20), NYVIP, KACJ, EXCL, MADJ2, X(15),  
0 Y(15), YA(15,9), XCL(15),  
1 NFUSPI, IFUSC, NPPCR, NCFPI, XFUS, YFLS, ZFUS,  
2 FUSY(100), FUSC(100), FLSA(100),  
3 PDY(6), PDY(6), PDZ(6), CLX(6,30), DELA(6,30),  
4 IVAL(20), CRCD(7,100), CSUF(7,100), CWING(20)
```

C  
C

COMMON /BLKTIL/ TITLE(6)

DIMENSION WORD1(16), WORD2(16)

```
DATA WORD1 /10HPCDY, FQUC, 10HPCDY, FWC-F, 10HPCDY, AFT-F,  
1 10HPCDY, HOLE, 10HPCDY, WAWI, 10HPCDY, PACT, 10HPCDY, FLUN,  
2 1H, 1H, 1H, 10HSUPF, PCUC, 10HSUPF, FVD-F, 10HSUPF, AFT-F,  
3 10HSUPF, HOLF, 10HSUPF, WAWI, 10HSUPF, PACT /  
DATA WORD2 /10HNESS, 10HACING STEP, 10HACING STEP,  
1 10HS, 10HNESS, 10HOPERATES, 10HNESS,  
2 1H, 1H, 1H, 10HNESS, 10HACING STEP, 10HACING STEP,  
3 10HS, 10HNESS, 10HOPERATES /
```

C

NPGDS = NBCDYS + NNAC  
NSLR = NPMLS + NHT + NVT

C  
C  
C

\*\*\*\*\* DAMAGE PARAMETERS \*\*\*\*\*

READ TITLE FOR DAMAGE CASE  
READ(5,1004) (TITLE(I), I = 1,6)

C

WRITE(6,5010) (TITLE(I), I = 1,6)  
READ INDICATORS FOR DAMAGE INPUT

C  
C

```
DO 100 I = 1, 20  
IDAM(I) = 0  
100 CONTINUE  
READ(5,1008) (IDAM(I), I = 1,17)  
WRITE(6,3001)  
DO 210 I = 1, 7  
WRITE(6,3002) I, IDAM(I), I+10, IDAM(I+10)  
210 CONTINUE
```

C  
C

FUSELAGE DAMAGE PARAMETERS

NPCASE = IDAM(1) + IDAM(2) + IDAM(3) + IDAM(4) + IDAM(5) + IDAM(6) + IDAM(7)

```
IF (NPCASE.EQ.0) GO TO 235  
WRITE(6,3003)  
READ(5,4002) DUMMY  
DO 230 K = 1, 7  
IF (IDAM(K).NE.1) GO TO 230  
IF (K.EQ.1) NVAFI = 5  
IF (K.EQ.2) NVAFI = 5  
IF (K.EQ.3) NVAFI = 5  
IF (K.EQ.4) NVAFI = 8  
IF (K.EQ.5) NVAFI = 7  
IF (K.EQ.6) NVAFI = 5  
IF (K.EQ.7) NVAFI = 2  
K1 = (K-1)*10 + 1  
K2 = (K-1)*10 + NVAFI  
WRITE(6,4003) K, WORD1(K), WORD2(K)  
DO 220 J = K1, K2  
READ(5,1002) (CRCD(I,J), I = 1, NPGDS)
```

```

C
220 WRITE(6,3004) J,(DPCD(I,J), I = 1, NPDS)
230 CONTINUE
C
C
C SURFACE DAMAGE PARAMETERS
235 WRITE(6,3010) (TITLE(I), I = 1,6)
WRITE(6,3005)
NWCASE=IDAM(11)+IDAM(12)+IDAM(13)+IDAM(14)+IDAM(15)+IDAM(16)
IF (NWCASE.EQ.0) GO TO 255
READ(5,4002) DUMMY
DO 250 K = 1, 16
IF (IDAM(K).NE.1) GO TO 250
IF (K.EQ.11) NVARI = 6
IF (K.EQ.12) NVARI = 6
IF (K.EQ.13) NVARI = 6
IF (K.EQ.14) NVARI = 9
IF (K.EQ.15) NVARI = 7
IF (K.EQ.16) NVARI = 5
K1 = (K - 11) * 10 + 1
K2 = (K - 11) * 10 + NVARI
WRITE(6,4003) K, VGRD1(K), VGRD2(K)
DO 240 J = K1, K2
READ(5,1002) (CSUP(I,J), I = 1, NS(I))
C
240 WRITE(6,3004) J,(DSUR(I,J), I = 1, NSUR)
250 CONTINUE
C
C
C MISSING PARTS OF SURFACES
255 DWING(13) = 0.000099
IF (IDAM(17).NE.1) GO TO 260
S-AG(5,1002) (DWING(I), I = 1, 6)
READ(5,1002) (DWING(I), I = 7, 10)
READ(5,1002) (DWING(I), I = 11, 13)
READ(5,1002) (DWING(I), I = 14, 20)
WRITE(6,3006) DWING(1), DWING(7), DWING(12)
WRITE(6,3007) DWING(2), DWING(8), DWING(14)
WRITE(6,3008) DWING(3), DWING(9), DWING(15)
WRITE(6,3009) DWING(4), DWING(10), DWING(16)
WRITE(6,3010) DWING(5), DWING(11), DWING(17)
WRITE(6,3011) DWING(6), DWING(12), DWING(18)
WRITE(6,3012) DWING(19)
WRITE(6,3013) DWING(20)
IF (DWING(13).EQ.C.C) WRITE(6,3020)
IF (DWING(13).EQ.C.C) DWING(13) = C.5
260 CONTINUE
C
C CONVERT ROUGHNESS HEIGHT FROM INCHES TO FT OR METERS
AA = J./12.
IF (METER.EQ.1) AA = 0.0254
DO 300 I = 1, 7
DPCD(I,1) = DPCD(I,1) * AA
DPCD(I,2) = DPCD(I,2) * AA
DSUR(I,2) = DSUR(I,2) * AA
DSUR(I,1) = DSUR(I,1) * AA
300 CONTINUE
C
C
C INPUT FORMAT STATEMENT
1001 FORMAT(14I5)
1002 FORMAT(7F10.0)
1004 FORMAT(7A10)
1005 FORMAT(A3,7X,A3,7X,A3,7X,A3,7X,A3,7X,A3)
1006 FORMAT(A4,A10)
1007 FORMAT(20I1, 4I5)
1008 FORMAT(20I1)
C
C
C FORMAT STATEMENTS FOR DAMAGE PARAMETERS
3001 FORMAT( //, 30X, *DAMAGE PARAMETERS *, / )
3002 FORMAT (10X, * IDAM(*,I2,*) =*, I2, 10X,
1 * IDAM(*,I2,*) =*, I2)

```

```

C
3C03 FORMAT(//,25X,* RCDDY DAMAGE PARAMETERS, DPCD(I,J) *,//,
1 13X,*I = 1*,5X,*I = 2*,5X,*I = 3*,5X,*I = 4*,5X,
2 *I = 5*,5X,*I = 6*,5X,*I = 7*,//)
3C04 FORMAT (2), *J = *,13,EX, 7F10.6)
3C05 FORMAT( //, 25X,* SURFACE DAMAGE PARAMETERS, DSUF(I,J)*,//,
1 13X,*I = 1*,5X,*J = 2*,5X,*I = 3*,5X,*J = 4*,5X,
2 *I = 5*,5X,*J = 6*,5X,*I = 7*,//)
3C06 FORMAT ( //, 20X, * SURFACE MISSING PART PARAMETERS, DWING(I)*,
1 //,15X,*DWING(1) ** ,F8.3,5X,*DWING( 7) ** ,F8.3,5X,
2 *DWING(13) ** ,F8.3)
3C07 FORMAT(15X,*DWING(2) ** ,F8.3,5X,*DWING( 8) ** ,F8.3,5X,
1 *DWING(14) ** ,F8.3)
3C08 FORMAT(15X,*DWING(3) ** ,F8.3,5X,*DWING( 9) ** ,F8.3,5X,
1 *DWING(15) ** ,F8.3)
3C09 FORMAT(15X,*DWING(4) ** ,F8.3,5X,*DWING(10) ** ,F8.3,5X,
1 *DWING(16) ** ,F8.3)
3C10 FORMAT(15X,*DWING(5) ** ,F8.3,5X,*DWING(11) ** ,F8.3,5X,
1 *DWING(17) ** ,F8.3)
3C11 FORMAT(15X,*DWING(6) ** ,F8.3,5X,*DWING(12) ** ,F8.3,5X,
1 *DWING(18) ** ,F8.3)
3C12 FORMAT (62X, *DWING(19) ** ,F8.3)
3C13 FORMAT (62X, *DWING(20) ** ,F8.3)
4C02 FORMAT(A2)
4C03 FORMAT ( /, 10X,*MODE *, I2, 2X, 2A10)
5C10 FORMAT (1F1, //, 10), (A10)
5C20 FORMAT ( //,30X, * I M P L U T E R P O P *, //, 20X,
1 *MOMENT ARM OF DEVICE THAT TRIMS ROLLING MOMENT, *,
2 *DWING(13), WAS INPUT AS G.G *, /, 20),
3 *PROGRAM WILL SET DWING(13) = 0.5 AND CONTINUE PROBLEM*)
C
END

```

```

OVERLAY(1,3)
PRCGRAM PINPT
C
C
C   READ IN INPUT VARIABLES TO CHANGE PREVIOUS PROBLEM
COMMON /INPUT/  A(3001)
COMMON /BLKTIL/ TITLE(6)
C
DIMENSION NO(30),AA(30),MO(14),IA(14), J1(10),I5(24)
EQUIVALENCE (I1(1),A(1)), (I2,A(15)), (I3,A(491)), (I4,A(492)),
1          (I5(1),A(633)), (I6,A(22))
C
READ(5,1000) (TITLE(I), I=1,6)
READ(5,1001)  M, N
WRITE(6,2000) (TITLE(I), I=1,6)
C
IF (M.EQ.0) GO TO 10
READ(5,1001) ( (MO(I), IA(I)), I=1,M)
10 IF (N.EQ.0) GO TO 20
READ(5,1002) ( (NO(J), AA(I)), I=1,N)
C
20 IF (N.EQ.0) GO TO 110
DO 100 I =1,N
J = NO(I)
A(J) = AA(I)
100 CONTINUE
C
110 IF (M.EQ.0) GO TO 210
DO 200 I =1,M
J = MO(I)
IF (J.LE.10) I1(J) = IA(I)
IF (J.EC.15) I2 = IA(I)
IF (J.EC.22) I6 = IA(I)
IF (J.EC.491) I3 = IA(I)
IF (J.EC.492) I4 = IA(I)
IF (J.GE.633) I5(J) = IA(I)
200 CONTINUE
C
210 CONTINUE
C
1000 FORMAT( 6A10 )
1001 FORMAT(14I5)
1002 FORMAT( 3(I5,5X,F10.0) )
C
2000 FORMAT(1H1,///,27X, 6A10)
END

```





```

SUP(I,1F) = ((SUP(I,8) + SUR(I,9)) / 2.) * SUR(I,10)
IF (I.LE.M2) SUR(I,18) = 2. * SUR(I,16)
C
C ASPECT RATIO = SUR(I,10)
IF (I.LE.M2) SUP(I,19) = (4.*SUP(I,10)**2) / SUR(I,1F)
IF (I.GE.N1) SUR(I,10) = (4.*SUP(I,10)**2) / (2.*SUR(I,1F))
C
C SYMMETRY CONDITION IF VERTICAL HAS Y DISPLACEMENT
IF (I.GE.N1.AND.SUR(I,12).GT.0.0) SUP(I,18) = 2.*SUR(I,1F)
C
C WETTED AREA = SUR(I,7)
IF (SUR(I,7).EQ.C.C) SUR(I,7) = SUR(I,18) * (2.
1 + 0.1843 * SUP(I,2) + 1.5268 * SUR(I,2)**2
2 - 0.8298 * SUP(I,2)**3)
C
C CHARACTERISTIC LENGTH = SUP(I,20)
SUR(I,20) = 0.6687 * SUR(I,8) * (1. + SUR(I,17)**2
1 / (1. + SUR(I,17)))
C *** TRCS, CLCS, AND CBAR2 ARE VARIABLE SWEEP PARAMETERS ***
C TRCS = SUR(NPMLS,3)
C CLCS = SUR(NPMLS,2)
C CBAR2 = SUR(NPMLS,20)
C SWET = SUR(NPMLS,7)
C
C SWEEP ANGLES SUR(I,21) AT C/4
C SUR(I,22) AT C/2
C SUR(I,23) AT TE
C SUR(I,24) AT MAX THICKNESS
C
C TSWP = TAN(SUR(I,4))
C CO = 4. / SUR(I,19) * (1.-SUR(I,17)) / (1. + SUR(I,17))
C SUR(I,21) = ATAN(TSWP - CO * 0.25)
C SUR(I,22) = ATAN(TSWP - CO * 0.50)
C SUR(I,23) = ATAN(TSWP - CO * 1.00)
C SUR(I,24) = ATAN(TSWP - CO * SUR(I,5))
C SWPCCS = SUR(NPMLS,22)
C
C 200 CONTINUE
C
C *****GEOMETRY FOR SINGLE PANEL WINGS *****
C *****TOTAL AIRPLANE PARAMETERS *****
C
C 9001 EQMAT (1H1, FF10.3)
C TWIST = A(15)
C WJTC = A(275)
C
C IF (NPMLS.FC.2) GO TO 120
C RG2 = SUR(I,10) + SUR(I,12)
C DCF = SUR(I,17) / RG2
C CR = SUR(I,9) + (SUR(I,8) - SUR(I,9)) * RG2 / SUR(I,10)
C SPLAN = (CR + SUR(I,8)) * RG2
C ARPLAN = 4.*RG2**2/SPLAN
C TR = SUR(I,9)/CR
C TAPR = TR
C YCRTE = SUR(I,11) - SUR(I,12) * TAN(SUR(I,4)) + CR
C SWF = SUR(I,4)
C SWPP = SWP
C SWPLE = SWP
C SWPCC = SUR(I,21)
C SWPFC = SUR(I,22)
C SWPTE = SUR(I,23)
C SWMT = SUR(I,24)
C SEXV = SUR(I,1F)
C TRAF = EXP(SFD WING AERODYNAMIC CHCD)
C CR/F = SUR(I,20)
C TCCW = SUR(I,3)
C CLD = SUR(I,2)
C ARXR = 4. * SUR(I,10)**2 / SUR(I,1F)
C CCX = SUR(I,8)
C CTX = SUR(I,9)
C XPX = SUR(I,11)
C YTX = SUR(I,11) + SUR(I,10) * TAN(SUR(I,4))
C YFY = SUR(I,12)
C YTX = TR2
C LXCC = SUR(I,10) * TAN(SWPCC)
C
C
C

```

```

C      MOMENT GEOMETRY,
C      CF = MAC DEFINED TO CENTERLINE
C      CR = 0.6667 * CF * (1. + TAPR**2 / (1. + TAPR))
C      IF (CMAC.EC.G.O) CMAC = CB
C      YR = BO2 / 3. * (1. + 2. * TAPR) / (1. + TAPR)
C      YP = SUR(1,11) + (YP - SLR(1,12)) * TAN(SWP)
C
C      IF (NPALS.EC.1) GO TO 125
C
C      ***** GEOMETRY FOR MAIN WING WITH TWC PANELS *****
C      ***** TOTAL AIRPLANE PARAMETERS *****
120  PC2 = SUR(2,12) + SUP(2,10)
C      CPB = SLR(1,12) / PC2
C      CR = SLR(1,9) + (SUR(1,8) - SLR(1,9)) * (SUR(1,12) + SUP(1,10)) /
1      SUP(1,10)
C      SPLAN = WPLAN
C      TP = TAPP
C      IF (IKNG.CT.O) GO TO 122
C
C      SPLAN = SLR(2,18) + (CF + SUR(1,11)) * (SUR(1,12) + SUP(1,10))
122  YR = FC2 * SUR(2,9) / (SPLAN - FC2 * SLR(2,9))
C      ARPLAN = 4. * EC2 ** 2 / SPLAN
C      SWPR = SUP(2,4)
C      IF (IKNG.CT.C) SWPR = SWP
C
C      SPLAN IS INPUT WHEN IKNG=1
C      TAPP IS INPUT FOR TWC-PANEL WING
C      SWP IS INPLT FOR TWC-PANEL WING
C
C      SEXW = SUR(1,18) + SUR(2,18)
C      SWPLE = ATAN((TAN(SUR(1,4)) * SLR(1,18) + TAN(SUR(2,4)) * SLR(2,18))
1      / SEXW)
C      SWPQC = ACOS((COS(SUR(1,21)) * SUR(1,18) + COS(SUR(2,21)) * SUR(2,18))
1      / SEXW)
C      SWPMC = ACOS((COS(SUP(1,22)) * SUR(1,18) + COS(SUR(2,22)) * SUR(2,18))
1      / SEXW)
C      SWPTF = ATAN((TAN(SUR(1,23)) * SUR(1,18) + TAN(SUR(2,23)) * SUR(2,18))
1      / SEXW)
C      SWMT = ACOS((COS(SUR(1,24)) * SUR(1,18) + COS(SUR(2,24)) * SUR(2,18))
1      / SEXW)
C
C      XCPRTE IS DEFINED AS THE Y-STA. OF T.E. OF EQUIV. WING ROOT CHORD
C      XCPRTE = SUR(2,11) + SUP(2,10) * TAN(SUR(2,4))
1      - PC2 * TAN(SWPLE) + CR
C
C      CBAR = EXPOSED WING AERODYNAMIC CHORD, HANDBOOK VII.P.0-7
C      CBAR = (SLR(1,20) * SUR(1,18) + SUR(2,20) * SUR(2,18)) /
1      (SLR(1,18) + SUR(2,18))
C
C      TOCW AND CLD FOR A WING WITH 2 PANELS ARE AREA WEIGHTED
C
C      TOCW = SCRT((SLR(1,18) * SUP(1,3) ** 2 + SUR(2,18) * SUP(2,3) ** 2)
1      / ((SLR(1,18) + SUR(2,18)) * SUP(1,3) ** 2 + SUR(2,18) * SUP(2,3) ** 2))
C      CLD = SCRT((SLR(1,18) * SUP(1,2) ** 2 + SUR(2,18) * SUP(2,2) ** 2)
1      / ((SLR(1,18) + SUR(2,18)) * SUP(1,2) ** 2 + SUR(2,18) * SUP(2,2) ** 2))
C      ARXR = 4. * ((SUR(1,10) + SUR(2,10)) ** 2) / (SUP(1,18) + SUP(2,18))
C
C      PARAMETERS BELOW ARE FOR EXPOSED WING
C      CRX = SUR(1,8)
C      CRX = SLR(1,9)
C      CTX = SUR(2,9)
C      XTX = SUR(2,11) + SUR(2,10) * TAN(SUR(2,4))
C      YRX = XTX - (BO2 - SUR(1,12)) * TAN(SWPLE)
C      YTX = SLR(1,12)
C      YTX = PC2
C      DXCC = SLR(1,10) * TAN(SUR(1,21)) + SUR(2,10) * TAN(SUR(2,21))
C
C      MOMENT GEOMETRY
C
C      CR = MAC DEFINED TO CENTERLINE
C      CP = 0.6667 * CR * (1. + TAPR**2 / (1. + TAPR))
C      YR = FC2 / 3. * (1. + 2. * TAPR) / (1. + TAPR)
C      YP = XTX - (BO2 - YR) * TAN(SWPLE)

```

```

C
C
C ***** CALCULATIONS FOR MOMENT PARAMETERS *****
125 SIX = SUP(1,18)
    YIX = SUP(1,10)
    ARI = SUP(1,10)
    IF (NPMS.EC.1) GO TO 126
C
    YCX = SUR(2,10) + G.E * YIX
    CRXP = (YCX * SUP(2,8) - 0.E * YIX + SUP(2,9))/SUP(2,10)
    SCXP = (CRXP * SUP(2,C)) * YCX
    APCP = 4.*YCX**2/SCXP
C
    HORIZONTAL TAIL PARAMETERS
126 XH = 0.0
    CMGA = 0.0
    IF (CMAC.EC.0.G) CMAC = CB
    IF (XCG.EC.0.C) XCG = XP
    IF (NHT.EC.0) GO TO 130
C
    LOCATE F.T. MAC/4 (XHT)
    XHT = SUR(M1,10)/3. * (1. + 2.* SUP(M1,17)) / (1. + SUP(M1,17))
    * TAN(SUP(M1,4)) + SUR(M1,20) * G.25 + SUP(M1,11)
C
    TAIL LENGTH (XH)
    XH = SQRT((SUP(M1,13)-ZCG)**2 + (XHT-XCG)**2)
    CMGA = ATAN2((SUR(M1,13)-ZCG), (XHT-XCG))
C
C
C
130 WRITE(6,2002) (PNAME(I), I = 1, N2)
    WRITE(6,2003) (POD(I,14), I = 1, N2)
    IF (METER.LT.2) WRITE(6,2004) (BCC(1,4), I = 1, N2)
    IF (METER.LT.2) GO TO 132
    DO 131 I = 1, 7
    W(I) = SUR(I,4) / 144.
    X(I) = SUP(I,7) / 144.
    Y(I) = SUP(I,18) / 144.
131 CONTINUE
132 CONTINUE
    IF (METER.EC.2) WRITE(6,2004) (W(I), I = 1, N2)
    WRITE(6,2005) (PNAME(I), I = 1, N2)
    WRITE(6,2006) (SNAME(I), I = 1, N2)
    IF (METER.LT.2) WRITE(6,2007) (SUR(I,7), I = 1, N2)
    IF (METER.EC.2) WRITE(6,2007) (X(I), I = 1, N2)
    WRITE(6,2008) (SUR(I,17), I = 1, N2)
    IF (METER.LT.2) WRITE(6,2009) (SUP(I,18), I = 1, N2)
    IF (METER.EC.2) WRITE(6,2009) (Y(I), I = 1, N2)
    WRITE(6,2010) (SUR(I,10), I = 1, N2)
    WRITE(6,2011) (SUR(I,20), I = 1, N2)
C
    DO 140 I = 1, N2
    W(I) = SUR(I,21) * 57.2958
    Y(I) = SUP(I,22) * 57.2958
    Y(I) = SUR(I,23) * 57.2958
    Z(I) = SUR(I,24) * 57.2958
140 CONTINUE
C
    WRITE(6,2012) (W(I), I = 1, N2)
    WRITE(6,2013) (Y(I), I = 1, N2)
    WRITE(6,2014) (Y(I), I = 1, N2)
    WRITE(6,2015) (Z(I), I = 1, N2)
    WRITE(6,2016) (SUR(I,5), I = 1, N2)
C
    MAX. CAMBER ORIGINATE (FCC)
    FCC = 0.06651 * CLD
    IF (SUP(1,1).NE.8) GO TO 170
    FCCM = AFC(1)
    DO 160 I = 1, NAFD
    FCCM = AMAX1(FCCM, AFC(I))
160 CONTINUE
    FCC = FCCM * CLD/CLDR
C

```

170 CONTINUE

```

WRITE(6,2016)
WRITE(6,2017) RC2
WRITE(6,2018) DCR
WRITE(6,2019) CP
IF (METER.LT.2) WRITE(6,2020) SPLAN
S = SPLAN/144.
IF (METER.EC.2) WRITE(6,2020) S
WRITE(6,2021) XCPT
WRITE(6,2022) ARPLAN
WRITE(6,2023) TR

```

```

C
S1 = SWPLE * 57.2958
S2 = SWPCC * 57.2958
S3 = SWPMC * 57.2958
S4 = SWPTF * 57.2958
S5 = SWMT * 57.2958

```

```

C
WRITE(6,2025) S1
WRITE(6,2026) S2
WRITE(6,2027) S3
WRITE(6,2028) S4
WRITE(6,2029) S5

```

```

C
IF (METER.LT.2) WRITE(6,2029) SEXW
S = SEXW/144.
IF (METER.EC.2) WRITE(6,2029) S
WRITE(6,2030) CPAF
WRITE(6,2031) TCCW
WRITE(6,2032) CLO
WRITE(6,2033) ARXR
WRITE(6,2034) CFY
WRITE(6,2035) CTX
WRITE(6,2036) XFX
WRITE(6,2037) YTY
WRITE(6,2038) YPX
WRITE(6,2039) YTX
WRITE(6,2040) XP
WRITE(6,2041) YP
WRITE(6,2042) XH

```

```

C
S1 = OMEGA * 57.2958
WRITE(6,2046) S1
2002 FORMAT (1H1,10X, *CALCULATED BODY PARAMETERS*,//,20X,7A10)
2003 FORMAT (//,2X, *FINNNESS RATIO = *, 7F10.3)
2004 FORMAT (/ ,2X, *WETTED AREA = *, 7F10.3)
2005 FORMAT (/ ,2X, *MAXIMUM AREA = *, 7F10.3)
2006 FORMAT (//////,2X, *CALCULATED SURFACE PARAMETERS*,//,20X,7A10)
2007 FORMAT (/ ,2X, *WETTED AREA = *, 7F10.3)
2008 FORMAT (/ ,2X, *TAPER RATIO = *, 7F10.3)
2009 FORMAT (/ ,2X, *PLANFORM AREA = *, 7F10.3)
2010 FORMAT (/ ,2X, *ASPECT RATIO = *, 7F10.3)
2011 FORMAT (/ ,2X, *CHORD LENGTH = *, 7F10.3)
2012 FORMAT (/ ,2X, *SWP C/4 = *, 7F10.3)
2013 FORMAT (/ ,2X, *SWP C/2 = *, 7F10.3)
2014 FORMAT (/ ,2X, *SWP TE = *, 7F10.3)
2015 FORMAT (/ ,2X, *SWP MAX T/C = *, 7F10.3)
2016 FORMAT (/ ,2X, *X/C AT MAX T/C = *, 7F10.3)

```

```

C
2016 FORMAT (1H1,20X, *TOTAL AIRPLANE PARAMETERS*,//;
2017 FORMAT ( ,2X, * WING SEMI-SPAN ,RC2 **,
1 F10.3)
20181 FORMAT ( ,2X, * BODY DIA/ WING SPAN ,DCB **,
1 F10.3)
20191 FORMAT ( ,2X, * WING ROOT CHORD AT CENTERLINE ,CR **,
1 F10.3)
20201 FORMAT ( ,2X, * THEO. WING PLANFORM AREA ,SPLAN **,
1 F10.3)
20211 FORMAT ( ,2X, * X-STA OF ROOT CHORD (CR) T.F. ,XCPT **,
1 F10.3)
20221 FORMAT ( ,2X, * EQUIV. WING ASPECT RATIO ,AR **,
1 F10.3)
20231 FORMAT ( ,2X, * EQUIV. WING TAPER RATIO ,TAPP **,
1 F10.3)
C
20251 FORMAT ( ,2X, * EQUIV. WING LEADING EDGE SWEEP ,SWPLE **,
1 F10.3)

```

```

2026 1 FORMAT ( 2X, * EQUIV. WING QUARTER-CHORD SWEEP ,SWPCC =*,
      1 F10.3)
2027 1 FORMAT ( 2X, * EQUIV. WING MID-CHORD SWEEP ,SWPMC =*,
      1 F10.3)
4001 1 FORMAT ( 2X, * EQUIV. WING TRAILING-EDGE SWEEP ,SWPTE =*,
      1 F10.3)
2028 1 FORMAT ( 2X, * EQUIV. WING MAX.THICKNESS SWEEP ,SWMT =*,
      1 F10.3)
2029 1 FORMAT ( 2X, * EQUIV. WING EXPOSED AREA ,SEXV =*,
      1 F10.3)
2030 1 FORMAT ( 2X, * EQUIV. WING EXPOSED MAC ,CPAF =*,
      1 F10.3)
2031 1 FORMAT ( 2X, * EQUIV. WING THICKNESS RATIO ,TCCH =*,
      1 F10.3)
2032 1 FORMAT ( 2X, * EQUIV. WING 2-D DESIGN CL ,CLD =*,
      1 F10.3)
2033 1 FORMAT ( 2X, * EQUIV. WING EXPOSED ASPECT RATIO ,ARXR =*,
      1 F10.3)
2034 1 FORMAT ( 2X, * CHORD LENGTH AT WING ROOT ,CRX =*,
      1 F10.3)
2035 1 FORMAT ( 2X, * CHORD LENGTH AT WING BREAK ,CBX =*,
      1 F10.3)
2036 1 FORMAT ( 2X, * CHORD LENGTH AT WING TIP ,CTX =*,
      1 F10.3)
2037 1 FORMAT ( 2X, * X-STA. AT L.E. OF WING ROOT ,XRY =*,
      1 F10.3)
2038 1 FORMAT ( 2X, * X-STA. AT L.E. OF WING BREAK ,XBX =*,
      1 F10.3)
2039 1 FORMAT ( 2X, * X-STA. AT L.E. OF WING TIP ,XTX =*,
      1 F10.3)
2040 1 FORMAT ( 2X, * Y-STA. AT L.E. OF WING ROOT ,YRX =*,
      1 F10.3)
2041 1 FORMAT ( 2X, * Y-STA. AT L.E. OF WING BREAK ,YBX =*,
      1 F10.3)
2042 1 FORMAT ( 2X, * Y-STA. AT L.E. OF WING TIP ,YTX =*,
      1 F10.3)
2043 1 FORMAT ( 2X, * X-STA. AT L.E. OF WING MAC ,XP =*,
      1 F10.3)
2044 1 FORMAT ( 2X, * Y-STA. AT L.E. OF WING MAC ,YP =*,
      1 F10.3)
2045 1 FORMAT ( 2X, * DELTA-Y C/4 WING TO C/4 TAIL ,XI =*,
      1 F10.3)
2046 1 FORMAT ( 2X, * INCLINATION LINE BET. WING AND TAIL ,OMEGA =*,
      1 F10.3)
      END

```

```

OVERLAY(3,0)
PROGRAM SURVEY
C
C
AERGDYNAMIC SURVEY
COMMON /INPUT/ A(30F1)
COMMON /OUTPUT/ R(222)
COMMON /CALC/ C(5C)
COMMON /R/KDAT1/ C(1411)
COMMON /R/KDAT2/ F(5G5)
COMMON /ELKTIL/ TITLE(6)
C
DIMENSION FMSLRV(2G), ALT(2G), SWPV(2G), CLLC(2G), CLMI(2G),
1 CG(2G), NAME(7), SNAME(7), SLF(7,2G)
DIMENSION TCL(21), TCC(21), TCF(21), T(21), TCLT(21), TCOI(21)
DIMENSION TICD(5), COSUR(7,5), CDEOR(7,5), TCTE(5), TCTS(5)
DIMENSION TERM(5)
C
EQUIVALENCE (A(6), ISWP), (A(7), IFEF), (A(4G), NSURV),
1 (A(492), NCLAS), (A(493), FMSURV(1)), (A(512), ALT(1)),
2 (A(523), SWPV(1)), (A(573), CLLC(1)), (A(582), CLM(1)),
3 (A(523), CG(1)), (A(275), MAC), (A(3G), SNAME(1)),
4 (A(177), SNAME(1)), (A(1), NPGDYS), (A(2), NMAC),
5 (A(3), NPALS), (A(4), NHT), (A(5), NVT), (A(13), XMAC),
6 (A(12), CMAC), (A(14), ZCG), (A(14), SUR(1,1)),
7 (A(22), ITRIM), (A(4), NHT), (A(1G), METEF)
C
EQUIVALENCE (R(43), CL), (R(44), CD), (R(45), CM), (R(46), ALPHA),
1 (R(142), H), (R(144), FM), (R(145), SWP ),
2 (R(146), JPASS), (R(60), ALC), (R(71), DE),
3 (R(137), CDAFT), (R(48), CFL), (R(47), CCM),
4 (R(49), CDR), (R(5C), CLA1), (R(1G6), CLDB),
5 (R(107), CLMAX), (R(51), CLT), (R(56), CMC),
6 (R(57), CMCCL), (R(51), DEL(L), (R(53), CM),
7 (R(54), FK), (R(52), CDT), (R(154), COSUR(1,1)),
8 (R(15C), CDEOR(1,1)), (R(5C), CCF), (R(117), CFC),
9 (R(135), CDMISC), (R(62), CMCH), (R(63), XH),
1 (R(64), CMFGA), (C(4C), XHT), (R(45), XCG)
C
DATA TERM /ICHOPTION =, IOHFORM =, ICHINTERF =,
1 IOHWAVE =, IOHBASE = /
C
IF (NSURV.LE.C) GO TO #00
SLP2 = SWPV(1)
NPGDS = NPGDYS + NMAC
NSURS = NPALS + NHT + NVT
M1 = NPALS + 1
C
REWIND 10
C
JPASS = 0
DO 50G L = 1, NSURV
FM = FMSLRV(L)
H = ALT(L)
SWEEP = SWPV(L)
CLLRW = CLLC(L)
XH = 0.0
CMEGA = 0.0
XCG = XMAC + CMAC * CG(L)
IF (NHT.EC.C) GO TO 20C
C
XHTL = XHT
IF (FM.GT.1.G) XHTL = XHT + SLF(M1,2G)*C.ZF
YH = SQRT((SUR(M1,13)-ZCG)**2 + (XHTL-YCG)**2)
CMEGA = ATAN2((SUR(M1,13)-ZCG), (XHTL-YCG))
C
200 CONTINUE
C
XNCL = NCLAS - 1
DCL = 0.0
IF (XNCL.GT.0.0) DCL = (CLMI(1) - CLLC(L))/XNCL
C
SLP = SWEEP/57.2C577C6
IF ( ISWP.GT.0.AND.SWEEP.NF.SWP2 ) JPASS = 0
SWP2 = SWEEP
C

```



```

WRITE (10) TCL, TCD, TCM, TA, TCLI, TCFI
WRITE (10) B
C
500 CONTINUE
C
600 CONTINUE
C
1000 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* FT. *, 5X,
1 *SWEEP ANGLE =*F6.2,* DEG. *, 5X,*TAIL DEFL. (DH) =*F6.2,
2 * DEG. * //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
3 T70,*CDL * , T82,*CDP*, T97,*COAFT*,
4 T113,*CD(L+R+AFT)*, / )
1014 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* METERS*, 5X
1 *SWEEP ANGLE =*F6.2,* DEG. *, 5X,*TAIL DEFL. (DH) =*F6.2,
2 * DEG. * //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
3 T70,*CDL * , T82,*CDP*, T97,*COAFT*,
4 T113,*CD(L+R+AFT)*, / )
1001 FORMAT(5X,F8.3,2F15.5, F15.2, 4F15.5)
1002 FORMAT(/10H CLA =,F8.5,1X,*PER DEG. *,10H ALC =,F8.5,1X,
1 *DEG. *, 29X,10H CLR =,F8.5 / 10H DELCL =,F8.5,
2 10H CK =,F8.5, 10H CLMAX =,F8.5 /
3 10H CMJ =,F8.5,1H,8X, 10H CM/CL =,F8.5,1H*,
4 10Y, 12H(*TAIL CFF) , // )
1003 FORMAT(* DRAG PEEKCCLN ----- *,5X,7(A10,5X))
1004 FORMAT(5X,A10,F7.5,7F15.5)
1005 FORMAT(1H1, /1X, 6A10 )
1006 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/FT =*1PF15.4,5X,
1 *L.E. SWEEP ANGLE =*GF6.2,* DEG. *,5X,*TAIL DEFL. (DH) =*
2 F6.2,* DEG. *
3 //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
4 T70,*CDL * , T82,*CDP*, T97,*COAFT*,
5 T113,*CD(L+R+AFT)*, / )
1015 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/METER =*1PF15.4,5X,
1 *L.E. SWEEP ANGLE =*GF6.2,* DEG. *,5X,*TAIL DEFL. (DH) =*
2 F6.2,* DEG. *
3 //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
4 T70,*CDL * , T82,*CDP*, T97,*COAFT*,
5 T113,*CD(L+R+AFT)*, / )
1007 FORMAT(/5X, 10H COMIN =,F7.5 //)
1008 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* FT. *, 5X,
1 *L.F. SWEEP ANGLE =*F6.2,* DEG. *,5X,*TRIMMED CONDITION*
2 //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
3 T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =C*,
4 T113,*CD AT DH=0*, / )
1012 FORMAT(1X, *MACH NO. =*F6.3, 5X,*ALTITUDE =*F7.0,* METERS*, 5X,
1 *L.F. SWEEP ANGLE =*F6.2,* DEG. *,5X,*TRIMMED CONDITION*
2 //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
3 T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =C*,
4 T113,*CD AT DH=0*, / )
1009 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/FT =*1PF15.4,5X,
1 *L.E. SWEEP ANGLE =*GF6.2,* DEG. *,5X,*TRIMMED CONDITION*
2 //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
3 T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =C*,
4 T113,*CD AT DH=0*, / )
1013 FORMAT(1X, *MACH NO. =*F6.3, 5X,*RN/METER =*1PF15.4,5X,
1 *L.F. SWEEP ANGLE =*GF6.2,* DEG. *,5X,*TRIMMED CONDITION*
2 //T10,*CL*, T22,*TOTAL CD*, T28,*CM*, T53,*ALPHA*,
3 T68,*CD LIFT*, T82,*CD R+AFT*, T97,*CL AT DH =C*,
4 T113,*CD AT DH=0*, / )
1010 FORMAT(* ,5X,7(A10,5X))
1011 FORMAT(5X,10HCAMPER =,F7.5,
1 /5X,10HCSAG RISE=,F7.5,
2 /5X,10HMISC =,F7.5 )
1016 FORMAT(36X, *ELEVON* )
C
1020 FORMAT ( ///, 5X, *NOTE. WAVE DRAG FOR *,A10,
1 * IS INCLUDED IN WAVE DRAG CF *, A10)
END

```



```

SUBROUTINE ADJUST(ID, ID2, XVAR, YVAR)
C
C
C   COMMON /INPUT/ A(30F1)
C   COMMON /MLKPR1/ KPRINT(50)
C
C   DIMENSION IVAL(20),X(15),YM(15,9),YA(15,9),XCL(15)
C
C   1  EQUIVALENCE (A(632),IVAL(1)), (A(657),X(1)), (A(672),YM(1,1)),
C   2              (A(687),YA(1,1)), (A(642),XCL(1)),
C              (A(653),NXVAR), (A(655),NXCL)
C
C   YVAR1 = YVAR
C   IF( ID.EQ.3 ) GO TO 300
C 50  NV = IVAL(ID)
C
C   IF( NV.EQ.0 ) RETURN
C 100 CALL LNTP(XVAR, VM, Y, YM(1,NV), NXVAR, 2)
C 200 CALL LNTP(XVAR, VA, Y, YA(1,NV), NXVAR, 2)
C   GO TO 500
C
C 200 NV = IVAL(ID2)
C   IF( NV.EQ.0 ) RETURN
C 400 CALL LNTP(XVAR, VM, XCL, YM(1,NV), NXCL, 2)
C   CALL LNTP(XVAR, VA, XCL, YA(1,NV), NXCL, 2)
C
C 500 YVAR = YVAR1 + VM + VA
C
C   IF( KPRINT(24).EQ.0 ) RETURN
C   WRITE(6,1000) ID, YVAR1, YVAR, VM, VA, YVAR
C 1000 FORMAT(5X,'ADJUST DUMP', 5X, *ID =*, I5/1X,5F15.5 )
C   RETURN
C   END

```

```

OVERLAY(3,1)
PRCCFAM VGEOM
C
C
COMPUTES GEOMETRY FOR VARIABLE-SWEEP CONFIGURATIONS
COMMON /CALC/ PD2, DCP, CF, SPLAN, XCRTE,
1 SWPCC, SWPMC, SWPTE, SWMT, SFXW, CBAP,
2 TDC, CLD, AFXR, CFY, CTY, XRY, YTX,
3 YFX, YTX, YF, YR, SWET, XYZ, SWLF, FDC,
4 SWPL, KPASS, AKPLAN, TR, YTX, STY, APT-CEXP,
5 SGXP, AFCD, CXOC, TWIST, WINC, XHT, TDC, S, CBAP2, CLDS,
6 YGX, SWMCS, CA1, PAC1, CA2, PAC2, DTCC
COMMON /BLKPR/ KPRINT(50)
C
COMMON /INPLT/ A(3061)
COMMON /OUTPLT/ B(223)
C
DIMENSION X(6), Y(6), XT(6), YT(6), D(6), P(6)
DIMENSION SUP(7,20)
EQUVALENCE (A(184), SUP(1,1)), (NPMLS, A(3)), (AFTAL, A(490)),
1 (AFTCR, A(488)), (AFTDC, A(486)), (AFTSW, A(487)),
2 (XPIVOT, A(484)), (YPIVOT, A(485)), (SWPF, A(21))
C
EQUVALENCE (SWP, P(145))
C
C
T = APS(SWP-SWPV)
C
IF (KPASS.EQ.0.AND.T.LE.0.005) GO TO 100
C
IF (KPASS.EQ.1) GO TO 25
C
CA1 = 0.0
CAC1 = 0.0
CA2 = 0.0
DAC2 = 0.0
DTCC = 0.0
SWPX = SWP
SWP = SWPR
C
C
25 N = NPMLS
Y(2) = SUP(N,12) + SUP(N,10)
ETA = SUP(N,12)/Y(2)
CR = (SUP(N,9) - ETA * SUP(N,9))/(1.-ETA)
TANSR = TAN(SUP(N,23))
CZ = SUP(N,9) * TAN(A(15)/E7.3)
CPDCR = 0. * (COS(SUP(N,4))/COS(SUP(N,4) - SUP(N,22))
1 + COS(SUP(N,23))/COS(SUP(N,22) - SUP(N,23)) )
X(1) = SUP(N,11) - SUP(N,12) * TAN(SUP(N,4))
Y(4) = X(1) + CZ
Y(1) = 0.0
Y(4) = 0.0
X(2) = X(1) + Y(2) * TAN(SUP(N,4))
X(3) = X(2) + SUP(N,9)
Y(3) = Y(2)
CC 30 I = 1,4
D(I) = SQRT((X(I) - XPIVOT)**2 + (Y(I) - YPIVOT)**2)
P(I) = ATAN((X(I) - XPIVOT) / (Y(I) - YPIVOT))
IF ((Y(I) - YPIVOT).LE.0.0) P(I) = P(I) + 3.141592
30 CONTINUE
C
35 CONTINUE
DSWP = SWP - SWPV
SWPV = SWP
C
DO 40 I = 1,4
XT(I) = YPIVOT + D(I) * SIN(DSWP + P(I))
YT(I) = YPIVOT + D(I) * COS(DSWP + P(I))
40 CONTINUE
C
DXCC = 0.75*XT(2) + 0.25*XT(3) - SUP(1,11) - SUP(1,8)*0.25
SWPTE = ATAN((XT(3) - XT(4))/(YT(3) - YT(4)))
SWPLE = SWP
CC = TAN(SWPTE) / TAN(SWPLE)
SWPCC = ATAN((1.0 - (1.0 - CC)/4.0) * TAN(SWPLE))

```

```

SWPMC = ATAN((1.0 - (1.0 - CC)/2.0)* TAN(SWPLE) )
XFLD = YT(2) - YT(2) * TAN(SWPLE)
YAFT = XT(3) - YT(3) * TAN(SWPLE)
CRCLP = YAFT - XFLD
CPCC = 0.5 * ( COS(SWPLE)/COS(SWPLE - SWPMC) + COS(SWPTF)/
1      COS(SWPMC - SWPTE) )
CLDS = SUR(N,2) * CPCC/CPCCP
TGCS = SUR(N,3) * CPCC/CPCCR
RC2 = YT(2)
DGR = SUP(1,12)/RC2
C
45 CONTINUE
TTIP = 1.0 + YT(2)* TAN(SWPLE) * (CC - 1.0)/CRCLP
TWIST = ATAN(DZ/(CRCLP*TTIP)) * 57.296
WINC = A(275) * (1. - TAN(CSWP)*TANSR) * COS(DSWP)
C
C3 = CPCLP * (1.0 - (YT(3)/YT(2)) * (1.0 - TTIP) )
SPLAN = (CRCLP + C3) * YT(2) + C3 * (YT(2) - YT(3))
ASPLAN = 4. * RC2 ** 2 / SPAN
YMD = YT(3) + (YT(2) - YT(3)) / (1.0 / SCRT(C3 / (TTIP + CRCLP)) + 1.0)
TR = 1. - (YMD / YT(2)) * (1. - TTIP)
YWC = SLP(NPNS,12)
C
YTE2 = XT(4) + YWD * TAN(SWPTF)
CPX2 = CRCLP * (1. - YWD / YT(2)) * (1. - TTIP)
SX2 = SPAN - (CRCLP + CPX2) * YWC
STOTAL = SX2
TCCW = TCCS
CLD = CLDS
SWPMCS = SWPMC
IF (NPNS.EC.1) GO TO 46
C
S1 = (XT(4) - XT(3)) / (YT(4) - YT(3))
S2 = (SUR(1,11) + SUR(1,8) - SUP(2,11) - SLP(2,8)) / SUP(1,10)
IE (S1.EC.S2) GO TO 46
C
C SOLVE FOR THE LINE INTERSECTION YI *****
1 YI = (XT(4) - SLP(1,11) - SUR(1,8) + S1*YT(4) - S2*SUR(1,12)) /
GA (S1 - S2)
SX2 = (XTE2 - SLP(2,11) - SUR(2,8)) * (SUR(2,12) - YI)
STOTAL = SUR(1,18) + SX2
TCCW = SCRT((SLP(1,3)**2 + SLP(1,18) + TCCS**2 * SX2) / STOTAL)
CLD = SCRT((SUP(1,2)**2 + SUP(1,18) + CLDS**2 * SX2) / STOTAL)
C
46 SWET = SX2 * (2. + .1643*TCCS + 1.5268*TCCS**2 - .8395*TCCS**3)
TP = TR + CPCLP / CRX2
CRAR2 = 0.66667 * CRX2 * (1. + (TP**2 / (1. + TP)))
CRAR2 = CRAR2 + DAC1 + (DAC2 - DAC1) * (DSWP / (AFTSW - SWPR))
C
SWMT = SUR(N,24) + DSWP
TCCW = TCCW + DTCC * DSWP / (AFTSW - SWPR)
C
IF (NPNS.EC.1) GO TO 47
C
COSMC = COS(SLP(1,22)) * SUP(1,18) + COS(SWPMC) * SY2
COSCC = COS(SLP(1,21)) * SLP(1,18) + COS(SWPC) * SX2
TANLE = TAN(SUR(1,4)) * SLP(1,18) + COS(SWPLE) * SX2
TANTE = TAN(SUR(1,23)) * SUR(1,18) + COS(SWPTF) * SX2
COSMT = COS(SLP(1,24)) * SLP(1,18) + COS(SWMT) * SX2
C
SWPMC = ACOS(COSMC / STOTAL)
SWPC = ACOS(COSCC / STOTAL)
SWPLE = ATAN(TANLE / STOTAL)
SWPTE = ATAN(TANTE / STOTAL)
SWMT = ACOS(COSMT / STOTAL)
C
47 CONTINUE
C
SFY = STOTAL
CPX = CPX2
CTY = CRCLP + TP
SCY = SX2
YDY = YMD - YWC
IF (NPNS.EC.1) GO TO 48
YW2 = (SUP(1,12) + SUP(2,12)) * 0.5

```

```

CRXP = CRCLP *(1.-Yw2 *(1.-TR)/YMIC)
SOXP = (CRXP +CTA) *(YMIC -Yw2)
ARCP = 4.*(YMIC -Yw2)**2/SCXF
ARXP = 4.*(YMIC -SUR(1,12))**2/SEXt
FCC = 0.06651 * CLD
C
IF (KPASS.FC.1) GO TO 50
C
IF (KPASS.FC.2) GO TO 49
DA1 = SUR(N,7) -SWET
DAC1 = SUR(N,20) -CRAR2
SWP = AFTSW
KPASS = 2
GO TO 35
49 CONTINUE
C
IF( AFTAL.GT.O.O ) DA2 = AFTAL - SWET
IF( AFTCP.GT.O.O ) DAC2 = AFTCB - CRAR2
IF( AFTCC.GT.O.O ) DTCC = AFTCC - TCCS
KPASS = 1
SWP = SWPX
GO TO 35
50 CONTINUE
C
IF (KPRINT(12).FO.1) WRITE(6,1000) KPASS, SWPV, DA1, DAC1,
1 CPRCC, CA2, DAC2, DTCC, CA, SWPCCS, CPRC,
2 TWIST, WJNC, SWPT, CRAR2, CLDS, TCCS, SWET, DCR, TCCV,
3 CLD, SFYK, SWPLF, SWPCC, SWPFC, SWPIF, ARPLAN, TR, SPLAN
1000 FORMAT(5X, *VGEOM DUMP, KPASS =*, I3, 10X, *SWPLE ** F12.4 /
1 (1X, 7F15.5 ) )
C
100 CONTINUE
END

```

```

OVERLAY(3,2)
PROGRAM MCRIT
C
C COMPUTES CRITICAL MACH NUMBER
C
COMMON/INPUT/ A(20F1)
COMMON/CLTACT/ F(223)
COMMON/CALC/ C(50)
COMMON/BLKDAT1/ C(1411)
COMMON/BLKDAT2/ E(40P)
COMMON/BLKMCR/ CLMCR(10), XMCR(10), NMCR, SA(32), SP(32),
1 SC(32), SD(22), SE(32), XL(32), ZT(32), ZS(32)
1 , RGC
C
DIMENSION CLTAB(21), TARMCR(21), PCF(7,20)
EQUIVALENCE (F(1),CLTAB(1)), (P(22),TARMCR(1)),(A(37),BCD(1,1))
1 , (A(3),NPALS), (B(67),FMCRP)
C
ID = A(184)
TCC = C(12)
SWEEP = C(7)
IF (NPALS.FG.1) GO TO 10
C
ETA = A(262) /C(1)
IF (ETA.LE.0.5) SWEEP = A(332)
10 CONTINUE
CLC = C(13)
AP = C(29)
PAMY = BCD(1,7)
PLNS = BCD(1,9)
FMCRB = 0.985
FPM = 10.
IF (PAMY.GT.0.0) FPM = PLNS/SCPT(.7854 *PAMY)
IF (FRN.LE.6.25) FMCRB = .82 * FRN*.1
C
IF (ID.LE.23) CALL CP2T(ID,C.6, TCC,CLC,SWEEP,AP,FMCRP)
C
100 CLTAB(1) = 0.0
DO 200 I = 1, 11
IF ( I.(GT.1) ) CLTAB(I) = CLTAB(I-1) + 0.1
CALL INTP(CLTAB(I), TARMCR(I), CLMCR, XMCR, NMCR, 2)
CALL ADJUST(0,4, CLTAB(I), TARMCR(I))
IF ( TARMCR(I).LT.C.0 ) TARMCR(I) = C.0
IF ( TARMCR(I).(GT.FMCRP) ) FMCRP = TARMCR(I)
C
200 CONTINUE
C
300 WRITE(6,1000) (CLTAB(I), TARMCR(I) , I =1,11)
C
1000 FORMAT(1H1, 5X, * MACH CRITICAL TABLE * /// 5X * CL *, 5X
1 * MACH CRITICAL * //(5X,2F15.4) )
C
END

```

```

C      SUBROUTINE CPZT(ID,XMACH,TGC,CLD,SWP,AF,FMCRP)
C      CALCULATES AIRFOIL PRESSURES - BRITISH METHOD
C      AND COMPUTES MACH CRITICAL USING CREST CRITERIA
C      COMMON /PLKMC/ CIMCR(10), XMCP(10), NMCR, SA(32), SR(32),
1      SC(32), SD(32), SE(32), XU(32), ZI(32), ZS(32)
C      COMMON /RLKPR/ RFC
C      COMMON /RLKPR/ KPRINT(50)
C      DIMENSION COSE(32), SINE(32), YCPI(32), DZDYL(32)
C      PI = 3.1415927
C      SWEEP = SWP * 57.29578
C      COSPHI = ( COS(SWP) )**(AF/(1.4+AR))
C      IF( SWEEP.GT.40.0 ) COSPHI = 0.5 * (COSPHI + .76604**(AF/(1.4+AR)))
C      COSPHI IS THE COSINE OF THE EFFECTIVE ISCPAR SWEEP
C      SWP = ACOS(COSPHI)
C      NCP = 31
C      NMCR = 0
C      DO 100 I = 1,NCP
C      YI = I
C      XN = NCP + 1
C      XN = NCP + 1
C      COSE(I) = COS(XI*PI/XN)
C      SINE(I) = SIN(XI*PI/XN)
C      XU(I) = 0.5*(1.0 + COSE(I))
100 CONTINUE
C      CALL SECT(ID, TGC, CLD)
C
C      ALPHA = 6.0
350 A = ALPHA/57.296
C      NMCR = NMCR + 1
C      XM = XMACH
C      IF( XM.GT.0.9 ) XM = 0.9
C      IF( XM.LE.0.0 ) XM = 0.01
C      XM2 = XM**2
C      IF( KPRINT(1).EQ.1 ) WRITE(6,1001) XM, ALPHA, SWEEP
C      CL = 0.0
C      CL = 0.0
C      DCPM1 = 0.0
C      DCPM1 = 0.0
C      CMI = 0.0
C      CM = 0.0
C      DO 500 IV = 1, NCP
C      SA(IV) = 0.0
C      IF( ID.LT.1000.PP.ID.GT.20 ) SP(IV) = 0.0
C      S3AX = ((-1.0)**IV - 1.0)/(XN*(1.0 + COSE(IV)))
C      SC(IV) = S3AX * SQRT(RFC/2.0)
C      SD(IV) = 0.0
C      SE(IV) = 0.0
C      DO 400 IU = 1, NCP
C      IMV = IU - IV
C      S1 = XM/SINE(IU)
C      S2 = COSE(IU)/(SINE(IU)**2)
C      S3 = XM/SINE(IU)
C      S4 = XM/SINE(IU) - 2.0 * ((-1.0)**IU - 1.0) /
1      ( XM * SINE(IU) + (1.0 - COSE(IU)) )
C      S5 = - S2
C      IF( IU.EQ.IV ) GO TO 150
C      SIGN = ((-1.0)**IMV - 1.0)/XN
C      DEN = (COSE(IU) - COSE(IV))
C      S1 = SIGN * 2.0 * SINE(IU) / (DEN**2)
C      S2 = (-2.0*(-1.0)**IMV*SINE(IU))/(SINE(IV)*DEN)
C      S3 = S1 - SIGN * 2.0 * (1.0/(SINE(IU)*DEN))
C      S4 = 2.0 * SIGN * (1.0 - COSE(IU)*COSE(IV)) /
1      (SINE(IV) * DEN**2)
C      S5 = - 2.0 * ((-1.0)**IU - 1.0) /
2      (XN * SINE(IV) * (1.0 - COSE(IU)) )
3

```

```

S5      = -2.0 + (-1.0)**IPV/DEM
C
150 CONTINUE
C
SA(IV)  = SA(IV) + S1      *ZT(IJ)
IF( ID.LT.10.CC.ID.GT.20 ) SP(IV) = SA(IV) + S2      *ZT(IJ)
SC(IV)  = SC(IV) + S2      *ZT(IJ)
SD(IV)  = SC(IV) + S2      *ZS(IJ)
SE(IV)  = SE(IV) + S5      *ZS(IJ)
400 CONTINUE
C
CALL CPUGV(1.0, A, SWP, IV, CPIL, CPL, XM)
C
CALL CPUCV(-1., A, SWP, IV, CPIL, CPL, XM)
C
DCPI    = CPIL - CPIU
CCP     = CPL - CPU
C
DZDXU  = SP(IV) + SE(IV)
DZDXL  = -SP(IV) + SE(IV)
C
YCPI(IV) = CPIU
DZDXUT(IV) = DZDXL
C
IF( KPRINT(1).EQ.1 ) WRITE(6,1000) XU(IV),CPIL,CPIU,DCPI,DZDXU,
1 DZDXL, CPU, CPL, CCP, XU(IV)
C
DX      = XU(NCP)
XR      = C.5 * (1. + YU(1))
IF( IV.NE.1 ) DY = XU(IV-1) - YU(IV)
IF( IV.NE.1 ) XF = C.5 * (XU(IV-1) + XU(IV))
CLI     = (1 + DY * (DCPI + DCPM1)) * C.5
CL      = CL + FY * (DCP + DCPM1) * C.5
CMI     = CMI + DX * (DCPI + DCPM1) * C.5 * (YP - C.25)
C       = C + CX * (DCP + DCPM1) * C.5 * (XF - C.25)
C
DCPM1  = DCPI
CCPM1  = CCP
C
500 CONTINUE
C
CLI     = CLI + XU(NCP) * DCPI * C.5
CL      = CL + XU(NCP) * DCP * C.5
CMI     = CMI + YU(NCP) * DCPI * C.5 * (0.5 + XU(NCP) - C.25)
CMI     = CMI + XU(NCP) * DCP * C.5 * (C.5 + XU(NCP) - C.25)
CPCRIT = (1.42857**2) * (C.52828 * (1. + C.2 * XM**2) + 3.5 - 1.)
C
IF( KPRINT(1).GT.0 ) WRITE(6,1002) CLI,CMI,CPCRIT,CL,CMI
600 CONTINUE
C
IF( A.LT.DZDXUT(NCP) ) GO TO 610
NCPF   = NCP - 1
ALPHA  = ALPHA - 1.
GO TO 350
C
610 CALL LNTP(A, XCREST, DZDXUT, XU, NCP, 4)
C
620 CALL LNTP(XCREST, CPCR, XU, YCPI, NCP, 4)
C
CPCRST = CPCR/CCSPHI**2
CF12   = CPCRST * CPCRST
Y*DDN  = 1./(1.0223 - .8260*CPCRST - .361*CF12 - .123*(CPCRST**2
1 - .0173*CPCRST**4)
1 IF( ID.EC.F ) Y*DDI = 1./(1.012 - .8551*CPCRST
1 - .4493*CF12 - .2210*CPCRST**2 - .0447*(CPCRST**4)
C
Y*DD   = Y*DDN/CCSPHI
IF( XMDD.LE.0.F2.CR.JC.FO.6 ) GO TO 640
IF( XMDD.LE.1.06 ) XMDD = .2666 - .5720*XMDD + 2.7122*XMDD**2
1 - 1.4182*XMDD**3
1 IF( XMDD.GT.1.06 ) XMDD = C.8545 + .1122*XMDD
C
640 CONTINUE
IF( XMDD.GT.FMCRB ) Y*DD = FMCRB
C
BMDD2  = 1. - (XMDD * CCS(SWP))**2
IF( BMDD2.GT.0.0 ) CLMDD = CLI/SCRT(BMDD2)

```

```

C      IF( XPRINT(1).GT.0 ) WRITE(6,1003) X*DD,CLMCR,XCFEST,CPCF,CGSPHI
      CLMCR(NMCR) = CLMCD
      X*CF(NMCR) = X*DD
C
C      700 IF( CLI.LE.0.0 ) RETURN
      IF( NMCF.GE.10 ) RETURN
      ALPHA = ALPHA - 1.
      GO TO 350
C
C      1000 FORMAT(5X,10F12.4)
C
C      1001 FORMAT(1H1, ///10X, 57HPRESSURE DISTRIBUTION AND SURFACE SLOPES F
100  2-D AIRFOIL , 6X,*MACH **F6.4, 5X,*ALPHA **F6.3 ///
1      10X,*SWEEP ANGLE =*, F7.3 //
2      12X,*X/C*,7X,*CPI(UPPER)*,2X,*CPI(LOWER)*,4X,*DCPI*,
3      5X,*DZ/DX(UPPER)*,2X,*DZ/DX(LOWER)*,1X,*CP(UPPER)*,
4      2X,*CP(LOWER)*,4X,*DCP*, 9X,*X/C* // )
C
C      1002 FORMAT(//10X,*CLI **F7.4, 10X,*CMI **F7.4, 20X,*CP(M=1) = *,
1      F8.4,
2      F7.4 / ) //11X,*CL **F7.4, 6X,*CP(C/4) =*
C
C      1003 FORMAT( /10X,*MACH CRIT **F7.4, 10X,*CL AT MDC **F7.4, 10X,
1      *X-CFEST **F7.4, 10X,*CP-CFEST **F8.4 /
2      50X,*COSINE OF EFFECTIVE ISRAF **F7.4 //)
C
      END

```



C  
C  
C

SUBROUTINE SFCT(IG, TCC, CLR)

CALCULATES THICKNESS AND CAMPER

COMMON /BLKMCB/ CLMCR(10), XMCB(10), NMCB, SA(32), SP(32),  
1 SC(32), SD(32), SE(32), XU(32), ZT(32), ZS(32)  
1 FCC  
COMMON /INPUT/ A(3041)  
COMMON /BLKPRI/ KPRINT(50)  
COMMON /BLKDAT1/ C(1411)

C

DIMENSION XSECT(26), Y164A(26), CAMB(26)  
DIMENSION Y163(26), Y165A(26), Y163(26), Y164(26), Y165(26),  
1 Y166(26), CAMB1(26)  
1 DIMENSION AC(11), A1(11), A2(11), A3(11), C1(11), C2(11), C3(11)  
DIMENSION AA(22), BE(22), CC(22), CD(22), XTT(22)  
DIMENSION XT(30), YC(30), YT(30)  
EQUIVALENCE (C(23), AA(1)), (C(45), PR(1)), (C(67), CC(1)),  
1 (C(89), CD(1)), (C(111), XTT(1)),  
2 (A(23), RLE), (A(9), NXSFCT),  
3 (A(304), YT(1)), (A(424), YC(1)), (A(454), YT(1)),  
4 (A(24), TCC5), (A(25), CLCF), (A(26), TECH)

C

DATA XSECT/ 0.0, 0.005, 0.0075, 0.0125, 0.025, 0.05, 0.075, 0.1,  
1 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6,  
2 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0 /  
DATA Y163 / 0.0, 0.082, 0.104, 0.1275, 0.175, 0.244, 0.295, 0.3362,  
1 0.3804, 0.4445, 0.4753, 0.4935, 0.5, 0.4635, 0.4766, 0.4456,  
2 0.414, 0.3715, 0.3224, 0.2712, 0.216, 0.1618, 0.1088,  
3 0.0604, 0.0214, 0.0 /  
DATA Y164 / 0.0, 0.082, 0.089, 0.125, 0.1701, 0.2343, 0.2826, 0.3221,  
1 0.342, 0.4392, 0.4639, 0.4844, 0.488, 0.4888, 0.4843,  
2 0.4805, 0.4735, 0.462, 0.445, 0.4227, 0.2281, 0.1722,  
3 0.1176, 0.0671, 0.0 /  
DATA Y165 / 0.0, 0.0772, 0.0832, 0.1169, 0.1574, 0.2177, 0.2647, 0.304,  
1 0.3666, 0.4143, 0.4502, 0.476, 0.4874, 0.4866, 0.4863,  
2 0.4812, 0.473, 0.464, 0.452, 0.4367, 0.4156, 0.3875, 0.3597,  
3 0.3321, 0.3051, 0.2706, 0.0 /  
DATA Y166 / 0.0, 0.0759, 0.0813, 0.1141, 0.1516, 0.2087, 0.2536, 0.2917,  
1 0.352, 0.4001, 0.4363, 0.4526, 0.4532, 0.4533, 0.5, 0.471,  
2 0.4665, 0.4665, 0.4502, 0.4367, 0.4176, 0.394, 0.3773,  
3 0.354, 0.3405, 0.0 /  
DATA Y163A / 0.0, 0.0816, 0.0883, 0.125, 0.1737, 0.2412, 0.2917, 0.3324,  
1 0.395, 0.44, 0.4714, 0.4813, 0.4805, 0.4866, 0.4837, 0.4813,  
2 0.4811, 0.4842, 0.4817, 0.4804, 0.474, 0.464, 0.4535, 0.4403,  
3 0.425, 0.4021 /  
DATA Y164A / 0.0, 0.0804, 0.0869, 0.1225, 0.1688, 0.2327,  
1 0.2915, 0.3109, 0.03813, 0.04272, 0.04406, 0.04837,  
2 0.04988, 0.04985, 0.04854, 0.04884, 0.04888, 0.04021,  
3 0.03507, 0.03127, 0.02623, 0.02103, 0.01582, 0.01067,  
4 0.00541, 0.0001 /  
DATA Y165A / 0.0, 0.0765, 0.0828, 0.1183, 0.1622, 0.2182, 0.265, 0.304,  
1 0.3658, 0.4127, 0.4483, 0.4742, 0.4812, 0.4895, 0.4883,  
2 0.4865, 0.4837, 0.4804, 0.4859, 0.4822, 0.4812, 0.48252,  
3 0.4771, 0.4738, 0.4604, 0.0 /  
DATA CAMB / 0.0, 0.082, 0.0836, 0.0863, 0.1055, 0.1103,  
1 0.1242, 0.02081, 0.03553, 0.04011, 0.04257, 0.05742,  
2 0.06120, 0.06204, 0.06571, 0.06851, 0.06631, 0.06508,  
3 0.06274, 0.05913, 0.05401, 0.04673, 0.03607, 0.02452,  
4 0.01226, 0.0 /  
DATA CAMB1 / 0.0, 0.025, 0.035, 0.0535, 0.063, 0.158, 0.212, 0.2585,  
1 0.3365, 0.0305, 0.04475, 0.0485, 0.111, 0.0515, 0.05475, 0.05515,  
2 0.04475, 0.05355, 0.0515, 0.0485, 0.04475, 0.036, 0.03365, 0.02585,  
3 0.158, 0.0 /  
DATA A0 / 5 \* C.2060, 3 \* C.14845, 3 \* 0.514246 /,  
1 A1 / 0.213337, -0.46082, -0.248867, -0.21027, -0.27118, /,  
2 0.412103, 0.103233, 0.083362, -0.40115, 2\*0.0 /,  
3 A2 / -2.931954, -0.4831, 0.175334, 0.3417, 0.1402, /,  
4 -1.67261, -0.581166, -0.18215, 1.1151, 2\*0.0 /,  
5 A3 / 5.22917, 0.559395, -0.264917, -0.32182, -0.082137, /,  
6 1.68869, 0.262205, -0.060, -1.06401, 2\*0.0 /,  
7 C1 / 0.2, 0.234, 0.215, 0.465, 0.7, 0.234, 0.315, /,  
8 0.445, 0.212, 0.315, 0.445 /,  
9 C2 / -0.40625, -0.688571, -0.233325, -0.684, -1.6625, /,  
1 -0.48571, -0.233333, -0.684, -0.68571, -0.2333, -0.684 /,  
2 C3 / -0.070312, -0.052178, -0.032407, 0.292, 1.3125, /,  
3 -0.03878, -0.032407, 0.292, -0.03878, -0.032407, 0.292 /



```

SR(I) = -D1(I40) -2.* D2(I40)*X -3.* F3(I40)*X*X
C
940 ZT(I) = TCC/.2 * ZT(I)
SP(I) = TCC/.2 * SR(I)
CALL LNTP(X, ZS1, XSECT, CAMR1, 26, 4)
ZS(I) = ZS1 * CLD
C
950 CONTINUE
C
960 IF( KPRINT(1).EQ.C ) RETURN
WRITE(6,1G04) TCC, CLD, RCC
CC 970 I = I, NCP
INV = NCP + 1 - I
ZUP = ZT(INV) + ZS(INV)
ZLD = -ZT(INV) + ZS(INV)
WRITE(6,1G05) XU(INV), ZT(INV), ZS(INV), ZUP, ZLD
970 CONTINUE
C
1G00 FORMAT(/5X,*XMU =*,F10.6,3X,*ZML =*,F10.6,3X,*XPL =*,F10.6,
1 3X,*ZML =*F10.6, /
1 3X,*ZPTE =*F10.6,3X,*ZTHIK =*,F10.6 // )
1G01 FORMAT(5X, 7F10.6 / )
1G02 FORMAT( 15.5X,5F10.6 )
1G03 FORMAT( 10F6.6 )
1G04 FORMAT(10X, *T/C =*F7.4, 10X,*CAMREP =*F7.4, 10X,*1.E.*RADIUS =*,
1 F7.5 //T17,*X/C*,T26,*THICKNESS*,T44,*CAMREP*,T6C,
2 *LPPE*,T75,*LCWER* )
1G05 FORMAT(5X, 5F15.5 )
2G00 FORMAT(/ 10X, *PICOMVFX AIRFOIL SECTION* / )
2G01 FORMAT(/10X, *6A SERIES AIRFOIL SECTION* /)
2G02 FORMAT(/ 10X, *SUPERFCITICAL AIRFOIL SECTION* /)
2G03 FORMAT( /10X, *6 SERIES AIRFOIL SECTION* /)
2G04 FORMAT( /10X, *4-DIGIT AIRFOIL SECTION* /)
RETURN
END

```

```

C      SUBROUTINE CPUQV(S, A, SWP, IV, CPI, CP, XM)
C      COMPUTES CP FOR AN INFINITELY SHEARED WING
C      COMMON /PLKPCR/ CLMCR(10), YMCR(10), NMCR, SA(32), SB(32),
1      SC(32), SD(32), SE(32), XU(32), YU(32), ZS(32)
C      , RCC
C
C      10 COSA = COS(A)
C      SINA = SIN(A)
C      COSL = COS(SWP)
C      SINL = SIN(SWP)
C
C      20 F1 = SA(IV)
C      F2 = SB(IV)
C      F3 = SC(IV)
C      F4 = SD(IV)
C      F5 = SE(IV)
C      FX = SQRT((1. - XU(IV))/XU(IV))
C      30 DUL = 1.0/(1. + ((F2 + S* F5)/COSL)**2 )
C
C      40 UQV2 = DUL * ( COSA * (1. + F1 * COSL + S* F4 * COSL)
1      + S* SINA * COSL * (1. + F3/COSL) * FX )**2
2      + DUL * ( COSA * SINL * (F1 + S* F4)
3      + S* SINA * SINL * (1. + F2/COSL) * FX )**2
4      + (SINL * COSA)**2 * (1. - DUL)
C
C      CPI = 1.0 - UQV2
C      CP = CPI
C
C      50 CPIQ = (1. - SINL * SINL + F1 * F1 + 2.0 * F1 * COSL)
1      / (1. + (F2/COSL)**2 ) - SINL * SINL
C      CPIQ = 1.0 - CPIQ
C      IF( CPIQ.GT.0.0 ) CPIQ = 0.0
C
C      YMN = XM * COSL
C      IF( YMN.LE.0.01.CP.YMN.GE.1.0 ) RETURN
C
C      XM2 = XM * XM
C      BETA = SQRT(1. - YMN * YMN)
C
C      60 SINA = SINA/BETA
C      IF( S1 = (COSL * COSL - CPIQ * YMN) * XM2
C      IF( TEST.LT.1.0 ) WRITE(6,1000)
C      IF( TEST.GE.1.0 ) GO TO 200
C
C      70 B = SQRT(1.0 - TEST)
C
C      F1 = F1/B
C      F4 = F4/BETA
C      F3 = F3/B
C      80 DUL = 1.0/(1. + ((F2 + S* F5)/(B * COSL))**2 )
C
C      90 UQV2 = DUL * ( COSA * (1. + F1 * COSL + S* F4 * COSL)
1      + S* SINA * COSL * (1. + F3/COSL) * FX )**2
2      + DUL * ( COSA * SINL * (F1 + S* F4)
3      + S* SINA * SINL * (1. + F2/COSL) * FX )**2
4      + (SINL * COSA)**2 * (1. - DUL)
C
C      100 CP = -1.42857/XM2
C      TEST = 1.0 + 0.2 * XM2 * (1. - UQV2)
C      IF( TEST.LT.0.0 ) GO TO 200
C      CP = -CP * ( TEST**2.5 - 1.0 )
C
C      200 CONTINUE
C      RETURN
1000 FORMAT(10X,'KUCHEMAN-LEBER CORRECTION FACTOR REACHES LIMIT' )
C      END

```



```

100 CONTINUE
C
RNCFT2 = RNCFT * FML2 / SPEED
CALL FDAG(FML2, RNCFT2)
CALL WDRG(FML2)
CAML2 = C.0
DO 200 I = 1, NFNLS
CAML2 = CAML2 + CDSUR(I,1) + CDSUR(I,2) + CCSUP(I,3) + CDSUF(I,4)
200 CONTINUE
C
300 CONTINUE
C
CALL DMIN(SPEED, RNCFT, CDD)
C
END

```

```

SUBROUTINE AT62(HGT,RNCFT,IFT)
COMPUTES REYNOLDS NO. AS FUNCTION OF ALTITUDE

RNCFT EQUALS RN/FT OR METER AT M=1
CONVERTS HGT TO FEET IF IFT = 1
ALT = HGT
IF (IFT.EC.1) ALT = ALT * 3.28084
X = 1.-6.8753E-6 * ALT
RNCFT = (X + 0.38312)/1.38312 * 7100574. * X**3.2559
IF (ALT.LE.36089.) GO TO 10
X = (36089.- ALT) * 4.806118E-5
RNCFT = 2.302229E6 * EXP(X)
CONVERT RNCFT TO RN/METER IF IFT=1
IF (IFT.EC.1) RNCFT = RNCFT*3.28084
CONVERT RNCFT TO RN/INCH, IF IFT = 2
IF (IFT.EC.2) RNCFT = RNCFT/12.
RETURN
END

```

```

SUBROUTINE DMIN(SPEED, RNDFT, CDMIN)
C
C
C
COMMON/INPUT/ A(2001)
COMMON/OUTPUT/ R(223)
COMMON/CALC/ C(50)
COMMON/BLKDAT1/ C(1411)
COMMON/BLKDAT2/ F(906)
C
C
DIMENSION PGC(7,20),TOTCD(5),CDBCD(7,5)
C
EQUIVALENCE (A(11),SREF), (A(1),NBCOYS), (A(2),ANAC),
1 (A(37),RND(1,1)), (B(149),TOTCD(1)),
2 (B(189),CDBCD(1,1)), (A(17),FMISC), (P(125),CDMISC)
C
NRND = NBCOYS + ANAC
DO 10 I = 1, NRND
CDBCD(I,5) = 0.0
C
C
CALL FCRG(SPEED, RNDFT)
C
CDMISC = (TOTCD(1)+TOTCD(2)+TOTCD(3)) * FMISC * 0.01
CALL ADJUST(?,0, SPEED, CDMISC)
C
DO 20 I=1,7
CDB = 0.0
RND = 1.
IF (1.0GT, NRND) GO TO 15
BASE = BCD(I,11)
CALL FCRG(SPEED, BASE, SREF, CDB)
RND = BCD(I,6)
IF (RND, EC, 0.0) RND = 1.
15 CDBCD(I,5) = CDB * RND
C
C
20 CONTINUE
C
TOTCD(5) = 0.0
DO 25 I = 1, NRND
TOTCD(5) = CDBCD(I,5) + TOTCD(5)
25 CONTINUE
C
CALL WDFG(SPEED)
C
CDMIN = TOTCD(1)+TOTCD(2)+TOTCD(3)+TOTCD(4)+TOTCD(5)
CDMIN = CDMIN + CDMISC
C
C
30 RETURN
END

```



```

C      SUBROUTINE FDRG(SPEED, PNOFT)
C
C      THIS SUBROUTINE CALCULATES FRICTION, FORM, AND INTERFERENCE
COMMON /INPUT/ A(20*1)
COMMON /OUTPUT/ R1(14*1), TCTCD(5),
1      CFSUF(7,5), CCRCD(7,5)
COMMON /CALC/ C(50)
COMMON /BLKPRN/KPRJNT(50)
C
C      DIMENSION BGD(7,20), SUR(7,30)
EQUIVALENCE (A(37),RCD(1,1)), (A(1=4),SUF(1,1)), (A(16),RUGHK),
1      (A(1),NOCYS), (A(2),NFAC), (A(3),NPALS), (A(4),NHT),
2      (A(5),NVT), (A(11),SPEF1), (R1(114),CRITM)
EQUIVALENCE (C(23),SWF1), (C(4),TCCS), (C(42),CBPF2),
1      (C(43),CLCS), (C(45),SWPMCS), (ISWP,A(f))
C
DO 10 I=1,3
TCTCD(I) = 0.0
DO 5 J=1,7
CPRCD(J,I) = 0.0
5 10 CFSUF(J,I) = 0.0
CONTINUE
C
J2 = NCCYS
K2 = NCCYS + NMAC
L2 = NPMLS
N2 = NPMLS + NHT + NVT
C
C      BODY CONTRIBUTION TO FRICTION, FORM, AND INTERFERENCE DRAG
CALL IFACT(4,REFUS,CRITM,SPEED,FI)
IF (K2.EC.0) GO TO 30
DO 20 J=1,K2
C
XL = BGD(I,1)
CALL CFEC(RNOFT,SPEED,XL,0.,RUGHK, CDF)
FR = RCD(I,14)
K = 1
IF (J.GT.J2) K = 2
CALL FFACT(K,FR,0,0,SPEED,0.0,FF)
AWET = SCC(1,4)
DRAG = CDF * AWET/SREF
FI = 1.
IF (RCD(I,5).NE.C.0) FI = RCD(I,5)
IF (SPEED.GT.1.0) FI = 1.
IF (I.GT.1) GO TO 15
RFFLS = RNCFT * XL
CALL IFACT(1,RFFLS,CRITM,SPEED,FI)
C
15 CPRCD(I,1) = DRAG
CPRCD(I,2) = DRAG * (FF - 1.)
CPRCD(I,3) = DRAG * FF * (FI - 1.)
20 CONTINUE
C
C      SURFACE CONTRIBUTION
30 IF (N2.EC.0) GO TO 50
CALCULATE DRAG FOR SURFACES
C
DO 40 I = 1, N2
XL = SUR(I,20)
IF (ISWP.EC.1.AND.I.EC.L2) XL = CBPF2
CALL CFEC(RACFT,SPEED,XL,C.,RUGHK, CDF)
C
TCC = SUR(I,3)
IF (ISWP.EC.1.AND.I.EC.L2) TCC = TCCS
TYP = SUF(1,1)
CAM = SUF(1,2)
IF (ISWP.EC.1.AND.I.EC.L2) CAM = CLCS
CALL FFACT(3,TCC,TYP,CAM,SPEED,CRITM,IF)
C
MTSWP = SUR(I,24)
IF (ISWP.EC.1.AND.I.EC.L2) MTSWP = SWPMCS
ID = 3
IF (I.LE.L2) ID = 2
CALL IFACT(ID,MTSWP,CRITM,SPEED,FI)

```

```

C
AWET = SUR(I,7)
IF (ISWF,FC,1,AND,I,FC,L2) AWET = SWET
DRAG = CFF * AWET/SREF
CDSUR(I,1) = DRAG
CCSUR(I,2) = DRAG *(FF -1.)
CDSUR(I,3) = DRAG *FF *(FI -1.)
C
40 CONTINUE
C
50 DO 60 I =1,3
DO 60 J =1,K2
C
60 TOTCD(I) = CDFCD(J,I) + TOTCD(I)
C
DO 70 J =1,N2
70 TOTCD(I) = CDSUP(J,I) + TOTCD(I)
80 CONTINUE
RETURN
END

```

```

C      SUBROUTINE CFEC(RNCFE, ZMACH, CBAR, XTR, ROUGHK, CF)
C      THIS SUBROUTINE CALCULATES THE SKIN FRICTION COEFFICIENT
C      USING THE WHITE-CHRISTOPH TECHNIQUE.
C      COMMON /BLKPRT/ KPRINT(50)
C      FTURB(X) = T**2*0.430/(ALOG10(RNL*X*T**1.67*F))**2.56
C      FLAM(Y) = 1.328*CFCFIL/SQRT(Y*RNCFE)
C      FPRIM(X) = 0.43*T*F**((ALOG10(RNL *X*T**1.67*F))**(-2.56)
1      -1.11178*(ALOG10(RNL *X*T**1.67*F))**(-2.56))
C      IF ( KPRINT(19).GT.0 ) WRITE(6,100) RNCFE,ZMACH,CBAR,XTR,ROUGHK
C      IF ( CBAR.LE.C.C.OR.CBAR.GT.10000. ) GO TO 500
C      ZMACH2 = ZMACH*ZMACH
C      T = 1.0/(1.0 + 0.178*ZMACH2)
C      F = 1.0 + 0.03916*ZMACH2*T
C      DXNP1 = 0.0
C      RNCF = 0.0
C      RNL = RNCFE
C      RN1 = RNL
C      Z = RNL * CBAR * T**1.67 * F
C      IF (Z.LE.1.0) GO TO 500
C      K = 0
C      IF ( ROUGHK.LE.0.0 ) GO TO 5
C      CFR = T/(1.65 + 1.62*ALOG10(CBAR/RULCHK))**2.5
C      CFS = FTURB(CBAR)
C      CFP = FPRIM(CBAR) - CFS
C      D = CFR - CFS
C      IF (K.EC.C.AND.D.LE.0.0) GO TO 5
C      IF (ABS(D).LE.0.0001) GO TO 5
C      RNL = RNL * (1. + D/CFP)
C      Z = RNL * CBAR * T**1.67 * F
C      IF (Z.LE.1.0) RNL = RN1 * 0.1
C      RN1 = RNL
C      K = K + 1
C      IF (K.GT.10) GO TO 4
C      RNCF = RNL * CBAR
C      GO TO 2
4      WRITE(6,1000) RNCFE, RNL, CFR
C      5 CONTINUE
C      CFCFIL = (1.0 - 0.1256*ZMACH2)**(-.12)
C      IF (XTR.LE.0.0) GO TO 100
C      CXN = 0.1 * XTR
C      10 DP = FPRIM(CXN)
C      IF (DP.EC.C) GO TO 200
C      DXNP1 = DXN - (CXN * FTURB(CXN) - XTR * FLAM(XTR))/DP
C      IF ( DXNP1.LE.C.0 ) DXNP1 = 0.5 * DXN
C      DX = ABS(DXNP1 - DXN)
C      DY = 0.0001*CBAR
C      IF (DX.LE.DY) GO TO 20
C      CXN = DXNP1
C      GO TO 10
C      20 XP = CXNP1 + CBAR - XTR
C      CF = (XP/CBAR)*FTURB(XP)
C      GO TO 300
C      100 CF = FTURB(CBAR)
C      GO TO 200
C      200 WRITE (6,1000) DXN, RNL
C      CF = 0.0
C      300 RNL = RNCFE
C      CBAR = FTURB(CBAR)
C      IF ( KPRINT(19).EC.0 ) GO TO 400
C      WRITE(6,1003) CF, CFCFIL, DXNP1, CFCFIL, RNL, RNCF

```

```

400 RETURN
500 CF = C.O
WRITE(6,1002) CRAP
RETURN
C
1000 FORMAT (10X,*SUBROUTINE CFEG WILL NOT CONVERGE*/ 1X,3F15.7 )
1001 FORMAT(10X,*CFEG INPUT*, 1P15.4, 0P4F15.7 )
1002 FORMAT(10X,*CRAP =*, 1P14.5, 2X,*OUT OF RANGE IN CFEG, CF SET EQUAL
1 TC ZEPIC* // )
1003 FORMAT(10X,*CFEG OUTPUT*, 4F15.7, 1P2E15.4 )
END

```

```

SUBROUTINE FFACT(JD,GEOM, TYP,CLD,SPEED,CRIT,FF)
C
C THIS SUBROUTINE CALCULATES FORM FACTORS FOR EACH COMPONENT.
C
C ITYP = TYP
C FF = 1.0
C IF (SPEED.GT.1.0) GO TO 40
C
C GO TO (10,20,30) ID
C
C FUSELAGE OR BOGIES (ID=1)
10 FR = GEOM
C FF = 1.0 + 60./FR**3.0 + 0.0025*FR
C GO TO 40
C
C NACELLE (ID=2)
20 FR = GEOM
C FF = 1.0 + 0.35/FR
C GO TO 40
C
C WING AND SURFACES (ID=3)
30 TCC = GEOM
C TCC2 = TCC*TCC
C IF (ITYP.LE.7) FF = 1.0 + 1.44*TCC + 2.0*TCC2
C IF (ITYP.EQ.9) FF = 1.0 + 1.2*TCC + 1.0*TCC2*TCC2
C IF (ITYP.GE.10.AND.ITYP.LE.20) FF = 1.0 + 1.68*TCC + 3.*TCC2
C IF (ITYP.NE.8) GO TO 40
C DELTAM = SPEED - CRIT
C IF (DELTAM.GE.C.0) ZK = 0.202
C IF (DELTAM.LE.-2) ZK = 0.12
C IF (DELTAM.GT.-2.AND.DELTAM.LT.0.0) ZK = 0.202 + .6972*DELTAM
1 FF = ZK*(CLD/.4+1.44*TCC + 2.0*TCC2 +1.
C
C 40 RETURN
C END

```

```

C      SUBROUTINE IFACT(ID,PARAM,CRITM,SPEED,FI)
C      THIS SUBROUTINE CALCULATES INTERFERENCE FACTORS FOR THE
C      FUSELAGE AND LIFTING SURFACES.
C      DIMENSION XMACH(7),REFUS(9),WRP(63)
C      DATA      XMACH / 0.25,0.4,0.6,0.7,0.8,0.9,0.95,0.97,0.99,1.0 /
1      REFUS / 3.0,10.0,20.0,30.0,40.0,60.0,80.0,100.0,10000.0 /
1      WRP / 1.063,1.02,0.979,0.951,0.926,0.907,0.867,
1      1.076,1.036,0.996,0.977,0.947,0.927,0.884,
1      1.066,1.029,0.989,0.966,0.943,0.91,
1      1.044,1.049,1.037,1.013,0.988,0.968,0.930,
1      0.902,1.018,1.032,1.015,1.003,0.983,0.973,
1      0.940,0.967,1.019,1.014,1.014,1.01,1.007,
1      0.937,0.944,1.011,
1      0.924,0.976,1.011,0.924,0.976,1.015 /
C      FI = 1.0
C      IF (SPEED.GT.1.0) GO TO 30
C      GO TO (10,20,20,40), ID
C      FUSELAGE CORRELATION FACTOR (FWP)
10  PARIG = PARAM/10.0**6
   XM1 = CRITM - .1
   XM = SPEED
   IF (XM.GT.XM1) XM = XM1
C      RWB = FI
C      WRP = FI
C      GO TO 30
C      LIFTING SURFACES
20  A = CCS(PARAM)
   IF (A.GE.C.75) RLS = -.9946 + 5.358*A - 4.5389*A*A + 1.2487*A**3
   IF (A.GT.C.65.AND.A.LT.0.75) RLS = .6966 - .752*(.75-A)
   IF (A.LE.C.65) RLS = 0.9214
   B = SPEED - C.25
   IF (B.LT.C.0) B = 0.0
   RLS = RLS - .0015 + .1818*B - .2756*B*B + 1.0677*B**3
   FI = RLS*RWB
   IF (ID.EC.3) FI = RLS
C      30 RETURN
C      40 RWB = 1.0
   RETURN
   END

```

```

C      SUBROUTINE WDRG(FMACH)
C
C      COMPUTES TOTAL AIRPLANE WAVE DRAG
C
C      COMMON/INPUT/  A(30F1)
C      COMMON /CLTPLT/ P(223)
C      COMMON /CALC/   C(5C)
C      COMMON /ELKDAT1/ D(1411)
C      COMMON /ELKDAT2/ F(90P)
C      COMMON /BLK&PD/ WPD(9)
C
C      DIMENSION CDSUR(7,5), CDPOD(7,5), TOTCD(5), SUR(7,20),POD(7,20)
C      EQUIVALENCE (B(14C),TOTCD(1)), (F(154),CDSLP(1,1)),
1      (P(1PC),CDACC(1,)), (A(37),PCD(1,1)),
2      (A(1P4),SUR(1,1))
C
C      EQUIVALENCE (AP,C(29)), (CLP,C(13)), (DOP,C(2)), (SWPLE,C(25)),
1      (SWPTE,C(8)), (IS,A(1P4)), (NBCDYS,A(1)),
2      (NHT,A(4)), (NNAC,A(2)), (NPALS,A(3)),
3      (NVT,A(5)), (SEFL,C(1C)), (SPLT,C(4)),
4      (SPEF,A(11)), (TDCW,C(12)), (TAPF,C(3C))
C
C      TOTCD(4) = 0.0
C      DC 10 I = 1,7
C      CDSUR(I,4) = 0.0
C      CDSUR(I,4) = C.0
C
C      J2 = NBCDYS
C      K2 = NBCDYS + NNAC
C      L2 = NPALS
C      M1 = NPALS + 1
C      N2 = NPALS + NHT + NVT
C
C      IF( FMACH.LE.1.0 ) RETURN
C      BETA = SQRT( FMACH**2 -1.)
C
C      PCCY CONTRIBUTIONS
C
C      IF( K2.EC.C ) GO TO 50
C
C      DC 20 I = 1,K2
C      PAMX = PCD(I,7)
C      PLNS = PCD(I,9)
C      PIN = C.0
C      PEX = SQRT(BCC(I,8)/3.14159)
C      FLFT = PCD(I,10)
C      PNC = PCD(I,6)
C      IF (RNG.F.C.C.C) BND = 1.0
C      IF (I.GT.J2) KIN = SQRT(BCC(I,12)/3.14159)
C
C      CALL CDWN(PAMX,PLNS,PIN,BETA,CDW1)
C      CALL CDWT(PAMX,FLFT,PEX,BETA,CDW2)
C
C      CDSDG(I,4) = (CDW1 + CDW2) * PAMX/SPEF * PNC
C      TOTCD(4) = TOTCD(4) + CDSDG(I,4)
C      CONTINUE
C
C      50 CONTINUE
C
C      WING AND SURFACE CONTRIBUTION
C
C      200 WPD(1) = AR * (1.-DOP)**2 * SPLAN/SEFL
C      WPD(2) = TAPF/(1.-DOP*(1.-TAPF))
C      WPD(3) = SWPLE
C      WPD(4) = SWPTE
C      WPD(5) = FMACH
C      WPD(6) = SEFL/SPEF
C      WPD(7) = IS
C      WPD(8) = CLC
C      WPD(9) = TDCW
C
C      CALL CDW(CDW1)
C      CDSUR(1,4) = CDW1
C      TOTCD(4) = TOTCD(4) + CDW1

```

```

C 250 IF (N2.EC.L2) GO TO 300
      DO 290 J = M1, N2
C
      WPD(1) = SLR(J,19)
      WPD(2) = SLR(J,17)
      WPD(3) = SLR(J,4)
      WPD(4) = SLR(J,23)
      WPD(6) = SLR(J,18)/SREF
      WPD(7) = SLR(J,1)
      WPD(8) = SLR(J,2)
      WPD(9) = SLR(J,3)
C
      CALL CDW(CDW1)
      COSLR(J,4) = CDW1
      TOTCD(4) = TOTCC(4) + CDW1
C 290 CONTINUE
C 300 CONTINUE
C
      RETRN
      END

```



```

SUBROUTINE CDWN(AMAX, XLNDS, FIN, PETA, CDW)
C
C NOSE WAVE DRAG (BODY COMPONENT)
C
C DIMENSION XLD(7), YBETA(4), FCDW(7,4)
C
C DATA XLD / 1., 2., 3., 4., 6., 8., 12. /
1 YBETA / C.0, (.32, 0.458, 0.6633 /,
2 FCDW / .1666, .0617, .0373, .0287, .0166, .0146, .0063,
3 .2376, .1254, .088, .061, .0364, .0177, .0082,
4 .2052, .1667, .1067, .066, .0311, .0175, .0061,
5 .3816, .2010, .1031, .0624, .0297, .0172, .0075 /
C
C CDW = C.0
C IF( AMAX.LE.0.0 ) GO TO 50
C DMAX=SQRT(AMAX/C.785398 )
C ELCD=XLNDS/DMAX
C IF( FIN.GT.0.0 ) GO TO 40
C IF( ELCD.LE.0.0 ) GO TO 50
C
C IF( PETA.GE.C.6633 ) GO TO 20
C
10 CDW = GLNT(ELCD,PETA, XLD, YBETA, FCDW, 7,4,7, 2,2)
C GO TO 50
C
20 X = PETA/SQRT(1.+ ELCD**2)
C CDW = (1.2 + 1.15 * X)/(1. + 1.9 * X) / (1. + ELCD**2)
C GO TO 50
C
C
C FOR A BODY WITH AN OPEN NOSE, SUCH AS A NACELLE, THE CONTRIBUTION
C OF THE NOSE TO THE TOTAL BODY WAVE DRAG COEFFICIENT IS COMPUTED AS
C INDICATED BELOW. (THE EXPRESSIONS USED FOR OPEN AND CLOSED BODY
C PCATTAIL CONTRIBUTIONS DO NOT REQUIRE THIS DISTINCTION--SEE CDW1.)
C
40 SORTB = SORT(PETA)
C IF( SORTB.LT.0.8144 ) SORTF = 0.8144
C CDW = ((1.-2.*FIN/DMAX)/ELCD)**1.5/SORTF
C
50 RETURN
C END

```

```

C
C
C
C
C
SUBROUTINE CDWT(AMAX, XLAFT, REX, PETA, CCW)
ROATTAIL WAVE (FAG (BODY COMPONENT))

DIMENSION DRCD(5), CDWRT(5)

CDW = 0.0
IF (AMAX.LE.G.0) RETURN
IF (XLAFT.LE.0.COJ1) RETURN
DPMX=SQRT(AMAX)/0.785398 )
ELCD=XLAFT/DPMX
X=PETA/ELCD
Y=X*X
Z=ELCD*ELCD

C
C
C
C
C
ROATTAIL WAVE DRAG IS A FUNCTION OF ROATTAIL FINNESS RATIO, BASE
DIAMETER/MAXIMUM DIAMETER, AND MACH NUMBER. COMPUTE THE ROATTAIL
WAVE DRAG COEFFICIENT AT FOUR DRCD(1) VALUES AND INTERPOLATE TO
C
DRCD(1)=0.0
CDWRT(1) = (1.165 -.5112*X -.5372*Y +.3964*X*Y)/Z
IF ( X.GT.1.0 ) CDWRT(1) = C.513/X/Z

C
DRCD(2)=0.4
CDWRT(2) = (1.067 -1.706*X +1.6632*Y -.686*X*Y)/Z
IF ( X.GT.1.0 ) CDWRT(2) = C.3352/X/Z

C
DRCD(3)=0.6
CDWRT(3) = (0.7246 -1.4616*Y +1.5795*Y -.6542*X*Y)/Z
IF ( X.GT.1.0 ) CDWRT(3) = C.1580/X/Z

C
DRCD(4)=0.8
CDWRT(4) = (0.2555 -.5008*X +.5024*Y -.2077*X*Y)/Z
IF ( X.GT.1.0 ) CDWRT(4) = C.C494/X/Z

C
CDWRT(5)=0.0
DRCD(5)=1.0
DRCD=2.C*PEX/DPMX
CALL LNTP(DRCD, CCW, DRCD, CDWRT, 5, 2)

C
RETURN
END

```

```

C      SUBROUTINE CDW(COSF)
C      PROCEDURE FOR WING PRESSURE DRAG
C
COMMON /BLKWPD/ AP, ZLAM, ZLE, ZTE, ZM, COSF, TYPE, CAM, TOC
COMMON /BLKPRN/ KPRINT(50)
COMMON /BLKCAT1/ 411)
C
C      DIMENSION AA(2), CC(22), DC(22), XT(22)
C      EQUIVALENCE (A(1),D(3)), (B(1),D(4)), (CC(1),D(67)),
1      (DC(1),D(80)), (XT(1),D(111))
C
C      10 N = TYPE
C
C      HCT = CAM * 0.055 / TOC
C      XTCC = XT(N)
C      TOC2 = TOC**2
C      ROOT = (AA(N) + PR(N) * TOC + CC(N) * TOC2 + DC(N) * TOC2 * TOC) / TOC
C      IF (N.EQ.8) ROOT = 0.8F216 * (TOC) ** 1.606 / TOC
C
C      CALCULATION OF QUANTITIES THAT ARE FUNCTIONS OF GEOMETRY.
C
17 ZKW = 1.0
C      TOC2 = TOC**2
C      TOC13 = TOC**0.33333333
C      TOC53 = TOC**1.66666667
C      ARW = AR * TOC13
C      ARW3 = TOC * AR**3
C      DEN1 = 1.0 / ARW + ARW3
C      DEN2 = 2.0 / ARW3 + 1.0
C      ZKT = 1. + 4. * (1.5 - XTCC * (1. + 5 * SQRT(ROOT))) ** 2
1      - 0.25 * SQRT(ROOT) * (1. - XTCC) ** 2
C      ZKC = 1.0 + 2.5 * HCT ** 2
C      ZKB = 1.069
C      IF (ZKW.EC.1.2) ZKB = 1.0
C      TR1 = 0.5 / (1. + ZLAM) ** 2
C      TR2 = 1. / (1. + ZLAM**2) ** 2
C      ZLAM1 = 1. + TR2 * (TAN(ZLE) + TAN(ZTE)) ** 2
C      Z1 = TAN(ZLE) ** 2
C      Z2 = TAN(ZTE) ** 2
C      IF (Z1.GE.Z2) ZKP = (COS(ZLE) + TE) * (Z1 - Z2) / ZLAM1
C      IF (Z2.GT.Z1) ZKP = (COS(ZTE) + TR) * (Z2 - Z1) / ZLAM1
C
C      ZK = (2.0 / DEN1) + 3.33 / DEN2 * ZKP
C      Y = APW * ZKP
C      IF (ARW.GE.1.0) Y = SQRT(7KP / (1.7321 * (1.0 - ZKP) + 1.0 / (ZKP * ARW**2)))
C
C      THIS DC LOOP SOLVES FOR *APE* BY NEWTON'S METHOD.
C
C      DO 100 I=1,10
C      Y3 = Y**3
C      Y4 = X**3
C      F = (2.0 * X) / (X4 + 1.0) + (3.33 * X3) / (X3 + 2.0) - ZK
C      FP = (2.0 - 6.0 * X4) / ((X4 + 1.0) ** 2) + (20.0 * X ** 2) / ((X3 + 2.0) ** 2)
C      FFP = F / FP
C      IF (KPRINT(20).EQ.2 .AND. I.EC.1) WRITE(6,90)
C
C      IF (KPRINT(20).EQ.2) WRITE(6,95) I, Y, F, FP, FFP
C
C      Y = Y - FFP
C      IF (ABS(FFP).LE.0.0001) GO TO 110
100 CONTINUE
110 IF (KPRINT(20).EQ.2) WRITE(6,96)
C
C      IF (KPRINT(20).EQ.2) WRITE(6,45) ZLE, ZTE, TOC2, TOC13, TOC53,
1      ARW, ARW3, DEN1, DEN2, ZKT, ZKC, ZLAM1, ZKP
C
C      BETA2 = ZM**2 - 1.0
C      BETA = SQRT(BETA2)
C      DJV = BETA / TOC2
C      BETAW = BETA / TOC13
C      BETA2W = BETA2 ** 2
C      CD1 = 0.0
C      CD2 = C.C
C      CFA = TOC2 * (2.0 + 3.333 / (ZKW**2.6))

```

```

ZKLIN = CDA/TCC2
IC (ZM.GT.1.0) CC1 = CDA/PETA
TEST1 = 1.0/COS(ZLF) + C.01
IF (ZM.GT. TEST1) CC2 = CDA/SCRT(BETA2-TAN(ZLE)**2)
ARE=X
TEST = ABS(TAN(ZLF))
UFXP = (1.+2.*ZLAM**4*.33333)
FB = (.3 + 0.7*ZXP**LEXP
YM = C.5*(1.+ZLAM**2 + (2.-ZLAM)**2 )
Z = COS(ZLF) + COS(ZTF)
IF( PETA.GT.TEST ) XM = (TEST/PETA)**Z
130 ZML = SCRT( TEST**2+1.C)
ARE2 = ARE**3
ZK1 = ZM**2.E
T1 = 1./APE / (1.+(1.+ZLAM) * FB + FFAL** (1.+ZK1) )
T2 = ARE3 / (1.+C.23333 *ARE3 +FFAL**4)
T3 = 2./ARE3 / (1.+(.66667+ZLAM) *PETA** (1.+ZK1) *FB)
T4 = 1./ (1.+3.* ARE * PETA**4)
ZFEXP = FB**XM
FBX = FB**YM
C1 = ZKF * ZK1 + PETA * FBXM + (T1+T2)
C2 = ZKF * ZK1 + PETA * FBXM + (T3+T4)
CDW = ZKT * ZK1 * ZKC * ZKF * TCC53 * (2./C1 +3.33/C2)
CDWA = CDW/TCC53
CCCSF = CDW*SCSF
IC (ZM.GT.1.0) CDD = CDW*DIV
AREW = ARE
ARE = AREW/TCC13

C
IF (KPRINT(20).EQ.?) WRITE(6,47)
C
IF (KPRINT(20).EQ.2) WRITE(6,46) DIV, PETA2, PETA, PETA2, CD4, CD1, CC2,
1 TEST, YM, FB, ZML, T1, T2, T3, T4, C1, C2, CD4, CDD
C
VLE = ZLE * 57.2956
VTE = ZTE * 57.2956
IF (KPRINT(20).GT.0) WRITE(6,25) AC, ZKF, ZLAM, ZKT, VLE, ZKC, VTE,
1 ZKLIN, TCC, ARE, TYPE, CAM, TCC, TCC12, XTCC, TCC53, PCOT,
2 FB, HQT, ZML, ZKW, ARE, SCSF, AFEW
C
55J IF (KPRINT(20).GT.0) WRITE(6,600)
C
IF (KPRINT(20).EQ.1) WRITE(6,610) ZM, PETA, XM, ZFEXP, CDWA, CCW,
1 CD1, CC2, CCCSP
C
C
90 FORMAT(4H C I, 5X, 14X, 14X, 14X, 14X, 2H FP, 13X, 3H FFP/)
95 FORMAT(1H , 13, 3X, 1P4F15.7)
26 FORMAT(1H C, 2X, 24Z1F, 12X, 24Z2F, 12Y, 4HTCC2, 11X, 5HTCC13, 10X, 5HTCC53/
1 1H , 2X, 24HAF, 12Y, 4HAPW, 11X, 4HDFN1, 11X, 4HDFN2, 11X, 2HZKT/
1 1H , 2X, 24H7KC, 12X, 5H7LAM, 10X, 2H7KF )
45 FORMAT(1H C, 1P5E15.7/(1), 1P5E15.7))
47 FORMAT(1H C, 2X, 3H DIV, 12X, 5HPETA2, 10X, 5HPETA, 10X, 6HPETA2, 5X, 3H CDA/
1 1H , 2X, 3H C1, 12X, 5HC2, 12X, 4HTEST, 11Y, 24YM, 13X, 2HFP/
1 1H , 2X, 24ZML, 12X, 24T1, 13X, 24T2, 13X, 24T3, 13X, 24T4/
1 1H , 2X, 24C1, 13X, 24C2, 12X, 3HCCW, 12X, 6HKLINF C)
48 FORMAT(1H C, 1P5E15.7/(1), 1P5E15.7))
25 FORMAT(//, 16Y, 24H W I T A G P R E S S U R E D R A G //
111X, 17Y INFLUENCE PARAMETERS, 16X, 15H GLTFLT PARAMETERS//
17X, 19H ASPECT RATIO = , F7.4, 10X, 12H KFLANG = , F7.4/
17X, 19H TAPER RATIO = , F7.4, 10X, 12H KTHICK = , F7.4/
17X, 19H LE SLEEP DEG = , F7.4, 10X, 12H KCAMP = , F7.4/
17X, 19H TE SLEEP DEG = , F7.4, 10X, 12H KLINTH = , F7.4/
17X, 19H THICK RATIO = , F7.4, 10X, 12H ACWIG = , F7.4/
17X, 19H AIFGIL TYPE. = , F3.0, 5.3, 5.3, 5X, 12H T/C 1/3 = , F7.4/
17X, 19H XT/C = , F7.4, 10X, 12H T/C 2/3 = , F7.4/
17X, 19H FO/T = , F7.4, 10X, 12H F = , F7.4/
17X, 19H H/T = , F7.4, 10X, 12H M = , F7.4/
17X, 19H KW = , F7.4, 10X, 12H ARE = , F7.4/
17X, 19H SW/SREF = , F7.4, 10X, 12H AREF = , F7.4/
600 FORMAT (//, 31X, 24H RESULTS //, 72H MACH R W I G EXP FEXP CCWIG
1 CD COTH2P COTH3C CD/SREF )
610 FORMAT(1H , 4(F7.2, 2X), 5(F7.5, 2X))
FFL:RN
END

```

```

C      SURPCUTINE ADPG(SPEED, AB, SREF, CDR)
C
C      BASE DRAG
C      COMMON/INPUT/ A(3CF1)
C      COMMON/OUTPUT/ R(223)
C      COMMON/CALC/ C(50)
C      COMMON/BLKDAT1/ D(1411)
C      COMMON/BLKDAT2/ F(50)
C      COMMON/BLKPF1/ KPRINT(50)
C
C      CDR = 0.0
C      IF( AB.LE.0.0 ) RETURN
C      IF(SREF.GE.1.0) GO TO 10
C      CDR = (0.1 + 0.1222 * SPEED**2) * AB/SREF
C      GO TO 20
C
10 CONTINUE
C      CDR = (1.42 / (3.15 + SPEED*SPEED) ) * AB /SREF
C      IF( SPEED.LE.1.0 ) CDR = 0.2222 * AB/SREF
C
20 CONTINUE
C      IF(KPRINT(15).EQ.C) GO TO 30
C      WRITE(6,1000) SPEED, AB, SREF, CDR
1000 FORMAT(1X*MACH **F10.7,5X*FASE A/F **F10.7,5Y*SREF **
1      F12.5,5X*FASE DRAG **F10.7 / )
C
30 RETURN
END

```



```
CDT = CD + DCCT  
CLT = CL + DCLT
```

C  
C

```
JPASS = 2
```

C

```
1000 FORMAT( /// 10X, *FUSLAGE AFT-END UPSWEEP CPAG * //  
1 10X, *WING ANGLE*, 10X, *CLCD* / )  
1001 FORMAT( 1X, 2F15.5 )  
END
```

```

SUBROUTINE CDDR(CL, XMACH, CDR)
C
C
C
COMMON /INPUT/ A(30P1)
COMMON /CALC/ C(50)
COMMON /CLTPLT/ CLTAR(21), TARMCR(21), P(71), FMCRC, FPL1, FPL2,
1 COMMON /BLKPR1/ CDC, CLAMP1, CLAPL2, EX(12), A2, A3, PL, FXY(89)
C
C
C
FMCRR = P(25)
C
C
CDR = 0.0
CDRG = 0.0
CALL LNTP(CL, XMCR, CLTAR, TARMCR, 11, 2)
IF( XMACH.CT.1.C ) RETURN
XM = XMACH - XMCR
YMC = XMACH - FMCRR
IF( XMC.CT.C.C ) CDRG = A2 * XMC**2 + A3 * YMC**2
IF( XM.CT.0.0 ) CDR = PL * XM**2
IF( XM.CT.C.12 ) CDR = 0.0144*PL + 0.24*PL * (XM -.12)
CDR = CDR + CDRG
C
RETURN
1000 FORMAT(5X, +CDR CUM+ / (1X, 7F15.5))
END

```



```

SUBROUTINE CLWBT(SPEED)
C
C
C COMPUTES WING-BODY-TAIL LIFT CONTRIBUTION
COMMON /BKCLA/ CLAI(11)
COMMON /INPUT/ A(20,F1)
COMMON /OUTPUT/ R(5,F1),CLA,ZLG,PX(35),CLAW,CLAR,CLAT,AA,PP,HSTAR,
1 CC,APFEAK,CLDH,BYX(11,C)
COMMON /CALC/ R02,CCR,CP,SPLAN,XCPTE,
1 S1PCC,S1PMC,S1PTE,S1MT,SEY4,CBAP,
2 TCC,CLD,ARXF,CPX,CTX,XPY,ATX,
3 YFX,YTY,XP,YP,XP,CMEGA,S1PLE,FCC,
4 S1PR,KPASS,AF,TP,YIX,SIX,ARI,CPXF,
5 SCXF,ARCP,FXCC,TWJST,WINC,XHT,TCCS,CFAR2,CLDS,
6 C(7)
C
DIMENSION B00(7,2C)
EQUIVALENCE (A(37),B00(1,1)), (NHT,A(4)), (NFCOYS,A(11)),
1 (TECH,A(2C)), (SREF,A(11)), (ITPIM,A(22))
C
C
C CLAW = 0.0
C
CLAI(1) = SPLAN
CLAI(2) = TCC
CLAI(3) = TR
CLAI(4) = AR
CLAI(5) = 0.0
IF (TECH.GT.0.) CLAI(5) = 0.0334
CLAI(6) = CLD
CLAI(7) = 0.0
IF (TECH.GT.0.) CLAI(7) = 0.09
CLAI(8) = DOB
CLAI(9) = 0.0
IF (TECH.GT.0.) CLAI(9) = 1.173762 * TCC
CLAI(10) = S1PMC
CLAI(11) = SREF
C
CALL AEP2(SPEED,.CLA)
CLAW = CLA
C
CLAT = 0.0
AA = C.C
PP = C.C
CC = C.C
APFEAK = 50.0
HSTAR = 0.0
CLDH = 0.0
C
IF( NHT.EQ.0 ) GO TO 30
C
20 CALL TAIL(SPEED)
30 IF (NHT.EQ.0.AND.JTRIM.FC.C) CALL ELEVEN(SPEED)
C
CLAR = 0.0
IF (NFCOYS.EQ.0) GO TO 50
C
WR = R02(1,2)
HB = R02(1,3)
AR = R02(1,7)
C
AB = MAX.CROSS-SECTIONAL AREA OF FUSELAGE
BL = A(254)
PLN = R02(1,9)
PERIM = 3.14159*SQRT((WR**2 + HB**2)/2.)
RHPF = WB/HP+PFCIM/SOFT(AP)
AK1 = C.005042*PHFF -.000379*RHPF**2 +.00001*RHPF**3
CLAB = AK1 + (BL/PLN)**0.3333 + AB/SREF
C
50 CLA = CLAW + CLAT + CLAR
C
CALL AALG(SPEED)
CALL ADJUST(3,C, SPEED, ALC)
C
RETURN
END

```

```

SUBROUTINE TAIL(SPEED)
C
C COMPUTES TAIL FLOW FIELD AND LIFT
C
COMMON/INPUT/  # (3001)
COMMON /OUTPLT/  # (223)
COMMON /CALC/    C (50)
COMMON /BLKDAT1/ F (1411)
COMMON /BLKDAT2/ F (908)
COMMON /BLKCLA/  CLA (11)
COMMON /BLKPR1/  KPRINT (50)
C
DIMENSION SUR (7,30), CDSUR (7,5), XSWPL (11), YPMIN (11)
DIMENSION XZCC (7), YSC (4), ZEL (7,4), XM (11), YFM (11)
C
EQUIVALENCE (SUR (1,1), A (184)), (CDSUR (1,1), P (154)), (AR, C (29)),
1 (CLAW, B (9E)), (NPNS, A (E)), (LINC, C (39)),
2 (TP, C (30)), (SWPCC, C (F)), (FCZ, C (1)),
3 (YP, C (21)), (AF, F (09)), (RH, F (100)),
4 (AGH, P (101)), (CB, A (12)), (SREF, A (11)),
5 (XHT, C (40)), (XCETE, C (5)), (XSWPL (1), F (1)),
6 (YRPM (1), D (12)), (CLIT, P (SE)), (CLCH, F (104)),
7 (DEDA, B (9E)), (CDATM, P (122)), (SPLAN, C (4))
C
DATA XZCC / 0.0, 0.1, 0.2, 0.3, 0.4, 0.6, 0.8 /,
1 YSC / 0.0, 0.2, 0.6, 1.0 /,
2 ZEL / 1.2, 0.2, 0.72, 0.57, 0.46, 0.31, 0.20,
3 1.1, 0.77, 0.64, 0.54, 0.44, 0.30, 0.15,
4 0.05, 0.76, 0.60, 0.48, 0.40, 0.28, 0.10,
5 0.31, 0.46, 0.22, 0.36, 0.27, 0.16, /,
6 XM / 0.0, 0.0, 1.0, 1.1, 1.2, 1.3, 1.4, 1.6, 1.8, 2.0, 2.2 /,
7 YFM / 1.0, 1.0, 1.0, 0.56, 0.4, 0.77, 0.66, 0.53, 0.46, 0.41, 0.4 /
C
C ** DOWNWASH IS CALCULATED ***
C
K = NPNS + 1
C
APHT = SLP (K, 10)
CXHT = SUR (K, 10)
SWLHT = SUR (K, 4)
HTLAM = SUR (K, 17)
CAMHT = SUR (K, 2)
HTCC = SUR (K, 3)
HTLE = SUR (K, 11)
HTY = SUR (K, 12)
HTZ = SUR (K, 13)
SWFCHT = SLP (K, 22)
TINC = SUR (K, 14)
C
10 ZKA = 1. / AR - 1. / (1. + AP ** 1.7)
IF (AR.LT.2.3) ZKA = 0.37 - 0.0567 * AR
ZKTR = (10. - 3.0 * TR) / 7.0
C
DEDA = 0.0
XPAC = XP + 0.25 * CB
XLHT = XHT - XPAC
IF (XLHT.LE.0.0) GO TO 30
C
ZKH = (1. - 0.5 * HTZ / RD2) / (XLHT / RC2) ** 0.33333
C
CGSO = SQRT (CCS (SWPCC))
20 CEDAQ = 4.44 * (ZKA * ZKTR + ZKH + CGSO) ** 1.19
C
IF (KPRINT (21).EQ.0) GO TO 21
WRITE (6, JCCG) CLAW, DEDAC, CDATM, HTZ, XHT, XB,
1 HTLE, CB, ZKA, ZKTR, ZKH, CCSC
21 CONTINUE
CALL AER? (0.1, CLAM)
CEDA = CEDAQ * CLAW / CLAM
C
C ** DYNAMIC PRESSURE AT THE TAIL ***
C
30 CDC = 0.0
PCCG = 0.0
IF (NPNS.EQ.0) GO TO 50

```

```

C
C   CDO = CDATE
C
C   XDC = (XHT - XCOTE)/CB
C   IF( XDC.LE.0.0 ) GO TO 50
C
C   ZWCC = 0.68 * SCRT(CDO * (XCC + 0.15))
C   CCCC = 2.42 * SCRT(CDO)/(XDC + 0.3)
C
C   GAMMA = ATAN(HT7/(XDC*CB))
C   AW = 2.0
C   CL = CLAB + AW
C   EW = 0.51566 * CL/AR
C   ZPC = XDC * TAN(GAMMA + EW - AW/57.3)
C   ZCZW = ZCC/ZWCC
C
C   IF( ZGZW.GE.1.0 ) GO TO 50
C
C   DCCC = CCCC * (CDS(1.570796 + ZCZW))**2
C
C   SC CDO = 1.0 - CCCC
C
C   ** CARRY-OVER FACTORS ZKWB AND ZKBW ARE COMPUTED **
C
C   60 CDBHT = 2.0 * HTY/(2.0 * HTY + SCRT(APHT * SEXHT) )
C   ZKWB = 1.002P + .7116*CDBHT + .42*CDBHT**2 - .1266*CDBHT**3
C   ZKBW = .0004 + 1.2662*CDBHT + .6018*CDBHT**2 + .1263*CDBHT**3
C
C   ** LIFT-CURVE-SLOPE OF THE EXPOSED SURFACE IS COMPUTED **
C
C   CLAI(1) = SEXHT
C   CLAI(2) = HTDC
C   CLAI(3) = HTLAM
C   CLAI(4) = ARHT
C   CLAI(5) = C.C
C   CLAI(6) = CAMHT
C
C   CLAI(7) = 0.0
C   CLAI(8) = 0.0
C   CLAI(9) = 0.0
C   CLAI(10) = SWPCHT
C
C   70 CALL AER2(SPEED, CLA)
C   CLAT = CLA * (ZKWP + ZKBW) + (1. - CEDA) * CDC
C   CLDH = CLA * ZKWP * CDC
C
C   ** INDUCED DRAG FACTOR FOR TAIL IS COMPUTED **
C
C   CALL LNTP(SWLHT, RTSUB, XSWPL, YRMIN, 11, 2)
C   PTAIL = RTSUB
C   IF( SPEED.LE.C.C ) GO TO 80
C   ZNDM = 12.0 * (CDS(SWLHT)+1.6) * (SPEED - 0.9)
C   FNDM = 1.0/(1. + 7*ZNDM + 2*ZNDM*ZNDM)
C   RTAIL = RTSUB * FNDM
C
C   80 HTK1 = (1.-PTAIL) * 0.01745 / (CLA + ZKWP)
C   HTK2 = PTAIL * 0.31631/ARHT * SREF/SEXHT
C   HTK = HTK1 + HTK2
C
C   AH = HTK * CLDH + CLGW
C   PH = 2. * HTK * CLAT + CLDH
C   AQH = (1.-CEDA) * (WINC - TINC)
C
C   IF (XLHT.GE.0.0) GO TO 90
C
C   CANAPD INTERFERENCE FACTOR ON WING (EL) IS COMPUTED
C
C   ZDC = APS(HYZ/CP)
C   SCOSW = SEXHT/SPLAN
C
C   ELS = PLNT(ZDC, SCOSW, XZCC, YSC, ZEL, 7, 4, 7, 4, 2)
C   CALL LNTP(SPEED, FM, XM, YFM, 11, 4)
C
C   EL = ELS + FM
C   IF (EL.GT.1.0) EL = 1.0

```

```

C      CLAW = CLAW - EL * CLAT
C
C 90    CONTINUE
C      IF( KPRINT(21),FC.1 ) WRITE(6,1000) CLAT, SPEED, CLA, ZKWE, COBHT,
1      ZKFW, DECA, COO, CLOH, AH, PH, ACP, DEAD,
2      RTSUR,RTAIL,HTK1,HTK2,PTK
C
C      RETURN
1000  FORMAT(10X,*TAIL LIFT DUMP* /(1X, 7F15.5) )
      END

```

SUBROUTINE FLEVGN(SPEED)

COMPUTES AERODYNAMIC EFFECT OF ELEVONS FOR TAILLESS AIRCRAFT

COMMON /INPUT/ A(3081)  
 COMMON /CALC/ C(50)  
 COMMON /OUTPUT/ P(223)  
 COMMON /FLKPAT/ KPPJNT(50)  
 COMMON /FLKDATE/ SPAN(6), YTPR(4), FKR(6,4), FKC(6,4),  
 SPAN2(11), YTPR2(5), FKS(11,5),  
 SPAN3(6), YTPR3(4), FKM(6,4),  
 SPAN4(5), YBF1(5), FKA(5,5),  
 SPAN5(5), YBF2(3), FKF(5,3)

DIMENSION XCF(5), YTC(7), ZCML(5,7)

EQUIVALENCE (CF, A(170)), (PFI, A(171)), (BFC, A(172)),  
 (AR, C(20)), (SWPMC, C(7)), (CLPIE, C(25)),  
 (TPP, C(30)), (SWPCC, C(6)), (FMCFC, P(114)),  
 (TPC, C(12))  
 EQUIVALENCE (CLDF, R(68)), (AKC, P(69)), (XCP, R(70))

DATA XCF / 0.05, 0.15, 0.25, 0.35, 0.4 /,  
 YTC / 0.02, 0.05, 0.09, 0.12, 0.15, 0.18, 0.21 /,  
 ZCML / -.51, -.39, -.26, -.14, -.22, -.47, -.38, -.28,  
 -.23, -.22, -.43, -.34, -.27, -.23, -.22,  
 -.30, -.32, -.27, -.23, -.22, -.35, -.30, -.26,  
 -.23, -.22, -.32, -.28, -.25, -.22, -.22,  
 -.30, -.27, -.24, -.22, -.22 /

XMC2 = SPEED \* CCS(SWPMC)  
 IF (SPEED.GT.FMCPG) XMC2 = FMCRC \* CCS(SWPMC)  
 BETA = SQRT(1.-XMC2\*\*2)

CF = ACS(1.-2.\*(CF))  
 CID = 2.\*(CF + SIN(CF)) / 57.3

CLDC = AP / (2. + SQRT(4. + (1.-XMC2\*\*2)\*(AR/CCS(SWPMC))\*\*2))

ZKPI = DLNT(BFI, TPR, SPAN, YTPR, FKF, 6, 4, 6, 2, 2)  
 ZKBI = DLNT(BFO, TPR, SPAN, YTPR, FKF, 6, 4, 6, 2, 2)  
 ZKB = ZKBI - ZKPI

ASC = SQRT(1. - (1.-CF)\*\*1.61)  
 ZKC = 1. + 1.375 \* (1.-ASC)\*\*2 / AR

CLDE IS THE FLAP LIFT EFFECTIVENESS PER DEG. ELEVON  
 CLDE = 0.77 \* CID \* CLDC \* ZKC \* ZKB  
 SWPHL = ATAN(TAN(SWPLE) - 4./AR \* (1.-TPP)/(1.+TPP) \* (1.-CF))

ZKDI = DLNT(BFI, TPR, SPAN, YTPR, FKC, 6, 4, 6, 2, 2)  
 ZKCC = DLNT(BFO, TPR, SPAN, YTPR, FKC, 6, 4, 6, 2, 2)  
 ZKD = ZKCC - ZKDI  
 FRCFILE DRAG TERM AKDP

AKDP = 1./BETA \* C.00168 \* CF\*\*2.322 \* CGS(SWPHL) \* ZKD

F = AP/6.283  
 D = BFG

ZKA = DLNT(E, PFI, SPAN4, YBF1, FKA, 5, 5, 5, 4, 2)  
 ZKF = DLNT(D, PFI, SPAN5, YBF2, FKF, 5, 3, 5, 2, 2)

INDUCED DRAG TERM AKDI  
 AKCI = ZKA \* ZKF \* 0.31F31 \* CLDE\*\*2 / AP

AKC = AKDP + AKDI

CCLP = 0.7208 / (CLDC \* ZKB)

ZKSWI = DLNT(BFI, TPR, SPAN2, YTPR2, FKS, 11, 5, 11, 2, 2)  
 ZKSWC = DLNT(BFO, TPR, SPAN2, YTPR2, FKS, 11, 5, 11, 2, 2)  
 ZKSW = ZKSWC - ZKSWI

ZKMI = DLNT(BFO, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)  
 ZKJI = DLNT(BFI, TPR, SPAN3, YTPR3, FKM, 6, 4, 6, 2, 2)  
 ZKM = ZKMI - ZKJI

```

C      DCML = DLNT(CF, TQC, XCF, YTQC, ZDCML, 5, 7, 5, 2, 2)
C      DCMCLP = DCML * ZKM + ZKSW * AF / 1.5 * TAN(SWPCC)
C      XCP = 0.25 - DCMCLP * DCCLP
C      IF (KPRINT(21).GT.G) WRITE(6, 1000) CF, SWPLF, SWPCC, SWPMC, AR,
1      ZKB, ZKC, CLDF, ZKF, AKCP, ZKA, ZKF, AKDI,
2      ZKSW, ZKM, DCML, XCP
1000 FORMAT(10X, *ELEVON DUMP*/(1X, 7F15.5))
C      RETURN
      END

```

```

SUBROUTINE AALO(SPEED)
C
C   CALCULATES ZERO LIFT ANGLE OF ATTACK
C
COMMON /INPUT/  A(2051)
COMMON /OUTPUT/ R(223)
COMMON /CALC/   C(50)
COMMON /BLKDAT1/ D(1411)
COMMON /FLKDAT2/ F(908)
COMMON /PLKPR1/ KPPINT(50)
C
EQUIVALENCE (TWIST,C(29)), (WJNC,C(39)), (PINC,A(275)),
1            (COP,C(2)),      (TDC,C(12)),  (SWPCC,C(6)),
2            (CLD,C(13)),     (TAP,C(30)),  (CIA,R(15)),
3            (ALC,F(60)),     (C1,R(56)),  (C2,R(57)),
4            (C3,R(9P)),     (NPMLS,A(3)), (NHT,A(4))
C
C
ALDC = 0.0
ALCT = 0.0
ALCI = 0.0
CLAB = 0.0
SWPCCB = 90.0
TAU = 0.0
ZKRW = 0.0
ZKWR = 0.0
TINC = A(275 + NPMLS )
C
XMCR = 0.75
XMND = SPEED * CCS(SWPCC)
TCCC = TCC/CCS(SWPCC)
C
IF ( TCCC.LT.0.1 ) XMCR = 0.75 + 1.25 * (0.1 - TCCC)
DALDC = 5.6
IF ( XMND.GT.XMCR ) DALDC = 5.6 - 249.0 * (XMND - XMCR)**2
IF ( DALDC.LT.0.0 ) DALDC = 0.0
C
ALEC = - DALDC + CLD
C
200 IF ( TWIST.EQ.0.0 ) GO TO 300
IF ( SPEED.GT.1.0 ) GO TO 300
TAU = TWIST
PETA = SQRT(1.0 - SPEED**2)
SWPCCB = 90.0
IF ( BETA.GT.0.0 ) SWPCCB = ATAN(TAN(SWPCC)/PETA) + 57.29576
CALC = 0.093 - 0.003571*SWPCCB + 0.5761*TAP - 0.2645*TAP**2
ALDT = - DALC + TAU
C
300 IF ( WJNC.EQ.0.0.AND.TINC.EQ.0.0 ) GO TO 400
ZKRW = .0004 + 1.2662*COR + .6015*DCP**2 + .1263*DCB**3
ZKWR = 1.0026 + .7114*DCB + .42 *DCP**2 - .13664*DCB**3
CLAB = C3 + 7*AL/(ZKRW + ZKWR) * C1
ALCI = (CLAB * WJNC + C2 * (WJNC - TINC))/(CLA
1      + (PINC - TINC))
C
400 ALB = ALEC + ALDC + ALDT + ALCI
IF ( KPRINT(13).GT.0 ) WRITE(6,1000) ALB, ALDC, ALCT,
1      ALCI, SPEED, TDC, CLD, TAU, SWPCCB, PCB,
2      TAP, C1, C2, C3, CLAB, ZKRW, ZKWR, WJNC,
3      TINC, PINC
C
1000 FORMAT(5X,*AALO DUMP*,5X,*ALB =*,F6.3,5X,*ALDC =*,F6.3,
1      5X,*ALCT =*,F6.3,5X,*ALCI =*,F6.3 / (5X,7F15.5) )
RETURN
END

```

SUBROUTINE CLRPK(SPEED, RE, RNOFT)

CALCULATES LIFT BREAK CL AND CLMAX

COMMON /INPUT/ A(2081)

COMMON /CLTPLT/ B(223)

COMMON /CALC/ C(50)

COMMON /PLKDAT/ KPPINT(50)

COMMON /PLKDAT1/ XSWPL(11), YFMIN(11),  
AAA(22), BFR(22), CCC(22), DCC(22), XTT(22),  
RGT(12), ATSW(6), TPR1(6), FXAC1(216), FYAC2(216),  
XPD4(6), YPD4(6), ZPC4(3), FPC4(5,5,3),  
XCY(6), YXM(6), ZPC(6,6),  
XPD4(4), YPD4(4), ZPD(6), FPD61(4,4,6),  
FPC42(4,4,6), FPC63(4,4,6),  
YI(3), YAP(2), RM4F(11),  
XIN13(10), YIN13(7), ZOUT13(10,7), ZOUT14(10,7),  
NXIN15, YIN15(6), NYIN15, YIN15(5),  
FCUT15(6,5), FCUT16(6,5),  
YI(6), YI(6), YC(3), YC(6), YAP(5), YDCL(5),  
XSWP(7), YTP(4), FEP25(7,4), FEP7(7,4),  
XCLDP(7), YAP(7)

COMMON /BLKDAT2/ XTP(6), YC1(6), YC2(6), XSWF1(4), YA(4),  
YP(4), XGY(6), YM(4), TAP(5,4), GTAP(5,4),  
XYCLM(12), YFY(6), FCLMY(12,6), XYC2(6),  
YYWACH(5), FCCLMY(9,5), XCY1(6), YXM(4),  
ZC1MAX(5,4), XQY2(5), YFCC(6), ZC1M(8,6),  
ZC2C1M(5,6),  
XSP(5), YCYA(5), FDA(5,6), YAR(5), YCC(5),  
FKVCFM(5,5), XANG(10), YPTCC(7), FEA(10,7),  
XC2(6), YAST(6), FFCM(5,5), YCT(6), YM(5), FDCM2(6,5),  
XYVCLM(6), FCLMXX(9,6),  
XY(12), XY(7), XF(12,7)

DIMENSION XF1(7), YF1(7), YF2(7)

EQUIVALENCE (ALC,P(60)), (CLA,B(59)), (TCC,C(12)), (CLD,C(13)),  
1 (CLAT,F(98)), (CLAW,P(56)), (CLS,P(112)), (CP,C(11)),  
2 (SWPCC,C(6)), (CLFP,P(105)), (CLDP,P(106)), (DF1,C(111)),  
3 (CLMAX,P(107)), (APRK,P(108)), (APAX,F(109)), (TS,A(184)),  
4 (DAMAX,P(110)), (APLQ,P(112)), (SWPLE,C(25)), (CCNCL,A(27))  
5 (AF,C(20)), (TP,C(20)), (FC,C(26))  
DATA XF1 / 5.0, 6.0, 6.47717, 6.77815, 6.95424, 7.29794, P. /,  
1 YF1 / -.125, -.11, -.07, -.01, 0., 0.03, 0.025 /,  
2 YF2 / -.1163, -.0713, -.055, -.0175, 0., 0.0375, 0.0463 /

NJ = TS  
CALL LNTP(TR, C1, YTP, YC1, 6,4)  
CICCS = ((1+1.) \* CCS(SWPLE))  
ARUP = 4./CICCS  
ARLC = 3./CICCS

IF (SPEED.GE.1.0) GO TO 20  
CYO = ANAP(NT) \* TCC  
IF (DYO.LT.0.5) CYC = 0.8  
IF (DYO.GT.2.4) CYO = 2.4

XMN = SPEED \* CCS(SWPCC)  
IF (XMN.LT.0.2) XMN = 0.2  
XMC = XMN  
CLIN = CLD/CCS(SWPLE)\*\*2  
IF (XMN.GT.0.7) XMC = 0.7

APRKO = DLAT(DYO, XTP, XCYC, YXM, ZPC, 6,6, 6, 4,2)  
APRK = (APRKO + (12.05-4.1\*XMN)\*CLIN) \* CCS(SWPCC)  
APRKC = 2./CCS(SWPLE)  
IF (APRK.LT.APRKC) APRK = APRKC  
CLPRG = CLA \* APRK

DCLPR = C.C  
FPR = 1.54  
IF (XMN.GT.0.5) FPR = 1.54 - 2.9 \* (XMN - 0.5)  
IF (CCNCL.GT.0.0) DCLPR = FPR \* ((CCNCL + .0643)\*CCS(SWPLE))\*\*2

CLPB = CLFBO + DCLPR



```

C DCLDRE = 0.1226 -.00714*SPEED -.12P57*SPFFD**2
C IF( NI.LE.9 ) DY = AMAP(NI)*TCC +RMAP(NI)*CLD
C IF( NI.GT.9 ) DY = AMAP(NI)*TCC +1.75*CLC
C
C TRANS = 0.0
C IF( DY.GT.1.65 ) TRANS = (DY - 1.65)/.4
C IF( DY.GE.2.05 ) TRANS = 1.0
C IF( SWPLE.GE.0.67 ) TRANS = 0.0
1 CLDP = CLPB + TRANS * (-.0376 -.24414*SPEED -.06P5+SPEED**2
+ .4146*SPEED**3 + RE * (CLCPF) )
C
C GO TO 30
C
C 20 CLD = 1.0
C RETAC = 10.0
C IF (SWPLE.NE.0.0) RETAC = SCRT('SPEED**2 -1.)/ABS(TAN(SWPLE))
C IF (RETAC.GT.1.0) GO TO 25
C CLSP1 = 0.85
C CLSP9 = 0.85
C IF( SWPLE.GT.0.7243 ) CALL LNTP(SWPLE, CLSP1, Y1, Y1, 6, 4)
C IF( SWPLE.GT.0.3545 ) CALL LNTP(SWPLE, CLSP9, X0, Y0, 6, 4)
C CALL LNTP(AR, CLCF, XAR, YDCL, 5, 4)
C CLSB0 = CLSB0 + CLCF
C CLSP = CLSP1 + (CLSP9 - CLSP1) *(RETAC - .1) * 1.25
C CLSB = CLSB + 0.5 * CLG
C CLFR = CLSP
C CLDR = CLSB
C 25 CONTINUE
C
C SUPERSONIC MAXIMUM LIFT COEFFICIENT ****
C CNAC4B = CLAW * 14.325 * SCRT(SPEED*SPEED -1.)
C
C CM = 1./SPEED
C CLMAX = 0.7722 + 0.3384 * CM :1.1648 * CM*CM -0.8215 * CM**3
C
C CM1 = 1. - CNAC4B
C IF(CM1.LT.0.0) CM1 = 0.0
C
C CLMAX = CLMAX - 0.048 * CM1
C
C 1 AMAX = 68.5429 -177.2327 * CM +461.9204 * CM**2
-629.4522 * CM**3 +321.4001 * CM**4
C
C AMAX = AMAX +15.8074 -3.0001*CNAC4B -12.8073*CNAC4B**2
C
C IF( CLMAX.GT.1.25 ) CLMAX = 1.25
C IF( AMAX.GT.54.5 ) AMAX = 54.5
C
C GO TO 300
C 30 CONTINUE
C
C CLMAX = 0.0
C DY = AMAP(NI) * TCC
C CALL LNTP(TR, C2, XTR, YC2, 6, 4)
C
C
C
C XMT = XTT(NI)
C IF( XMT.LI.0.3 ) XMT = 0.3
C IF( XMT.GT.0.45 ) XMT = 0.45
C IF( AR.LE.ARLC ) GO TO 200
C
C C1MAXB = DLNT(DY, XMT, XDY1, YXMT, ZC1MAX, 9, 4, 9, 2,2)
C
C FCC = 100.*FCC
C DC1MAX = DLNT(DY, FCC, XDY2, YFCC, ZDC1M, 6, 6, 6, 2,2)
C IF( XMT.GE.0.25 ) DC1MAX = DLNT(DY, FCC, XDY2, YFCC, ZDC1M, 6, 6, 6, 2,2)
C
C RNCB = ALG10(RNDFT * CB)
C CALL LNTP(RNCF, F1, XF1, YF1, 7, 2)
C CALL LNTP(RNCF, F2, XF1, YF2, 7, 2)
C DC1RN = F1 + F2 * DY
C
C C1MAX = C1MAXB + DC1MAX + DC1RN
C
C SWEEP = SWPLE * 57.2956

```

```

100 CALL LNTP(SWEEP, AY, XSWP1, YA, 4,4)
   CALL LNTP(SWEEP, FY, XSWP1, YB, 4,4)
C
DYM14 = DY - 1.4
IF( DYM14.LT.C.C ) DYM14 = 0.0
IF( DY.GT.2.5 ) DYM14 = 1.1
C
CLMGC1 = AY - BY * DYM14
CLMAX = CLMCC1 * CLMAX
C
C3 = DLNT(DY, SPEED, XDY, XM, CTAR, F, 4, F, 2, 2)
D = DLNT(DY, SPEED, XDY, XM, CTAR, F, 4, F, 2, 2)
C
DCLMAX = C3 + (D-C3)* SWEEP/60.
CLMAX = CLMAX + DCLMAX
C
120 DAMAX = DLNT(SWEEP, DY, XSP, YDYA, FDA, E, 6, E, 2, 2)
AMAX = CLMAX/CLAW + ALC + DAMAX
CLS = CLA * (AMAX - 2.*DAMAX - ALC)
GC TO 300
C
C
C
C
200 CONTINUE
C
ABETA = AR/SCRT(1. - SPEED * SPEED)
XCLM = ABETA * CLCDS
IF( XMT.LE.C.35.OR.XCLM.GE.2.0 ) CLMXP = DLNT(XCLM, DY, XXCLM,
1 YDY, FCLMX, 13, 6, 13, 2, 2)
IF( XMT.GT.0.35.AND.XCLM.LT.2.0 ) CLMXP = DLNT(XCLM, DY, XXXCLM,
1 YDY, FCLMXX, 9, 6, 9, 2, 2)
C
C2TAN = (C2 + 1.) * AP * TAN(SWPLE)
DCLMX = DLNT(C2TAN, SPEED, XYC2, YMACH, DCLMX, 9, 5, 9, 2, 2)
C
CLMAX = CLMXP + DCLMX
AST = AP * COS(SWPLE) * (1. + 4.*U + TR*TR)
AMAXB = 35.0
IF( XCLM.GT.0.9 ) AMAXB = 49.8472 - 20.6922*XCLM + 5.6674*YCLM**2
1 - 0.4279*XCLM**4
IF( YCLM.GT.3.6 ) AMAXB = 21.0
IF( C2TAN.LE.4.5 ) DAM = DLNT(C2TAN, AST, YC2, YAST, FDAM, 6, 9, 6, 4, 2)
IF( C2TAN.GT.4.5 ) DAM = DLNT(C2TAN, SPEED, YCT, YM, FDAM2, 6, 5, 6, 4, 2)
C
AMAX = AMAXB + DAM
C
300 CONTINUE
C
TAIL CONTRIBUTION TO CLMAX *****
C
DEL = CLAT * 57.3 * SIN(AMAX/57.3) * COS(AMAX/57.3)**2
CLMAX = CLMAX + DEL
C
IF( KPRINT(16).EQ.1 ) WRITE(6,1000) CLFP, CLCP, APFKC, APFK,
1 XMN, DYC, DCLPS, TRAV, RETAC, DY, C1, C2, AMAX, APUP,
2 ARLC, XMT, CLMAYR, DCIMAX, CLMCC1, DCLMAX, DEL,
3 XCLM, CLMXP, C2TAN, DCLMX, CLMAX, DC1RN, RNCB
1000 FORMAT(10X, *CLBKK DUMP* / (1X, 7F15.5) )
C
RETURN
END

```

```

SUBROUTINE COLL(SPEED, RNOFT, FK, DELCL, PRIMEK, AKD, AKE)
C
C COMPUTES DPAG DUE TO LIFT CONSTANTS
C
COMMON/INPUT/ A(2081)
COMMON/CLTPLT/ R(223)
COMMON/C/LC/ C(50)
COMMON/BLKCAT1/ C(1411)
COMMON/BLKDAT2/ F(908)
COMMON/BLKPR1/ KFPINT(50)
C
DIMENSION CCLLV(13), SCUFV(13), XRN(0), YDRT1(C), YDRT2(C)
DIMENSION XSWPL(11), YRMIN(11), AA(22), FF(22), CC(22), DD(22),
1 XSWP(7), YTP(4), FEP35(7,4), FFF7(7,4), XCLDB(7), YAKP(7)
2 ,SUF(7,20)
C
EQUIVALENCF (D(1),XSWPL(1)),(D(12),YRMIN(1)),(D(23),AA(1)),
1 (D(45),PR(1)),(D(47),CC(1)),(D(49),CF(1)),
2 (D(132),XSWP(1)),(D(133),YTP(1)),
3 (D(137),FFF7(1,1)),(D(138),XCLDB(1)),(D(140),
4 YAKP(1)),(D(1342),FEP35(1,1)),
5 (A(194),SUF(1,1)),(R(FF,A(23)),(TOCR,A(24)),
6 (TOCS,C(41)),(CPAG,C(42)),(COPCL,A(27)),
7 (OCB,C(2)),(ISWP,A(6)),(CLP,C(13)),
8 (SWQC,C(5)),(SKPL,C(25)),(MPALS,A(3)),
9 (P,C(29)),(TK,C(30)),(R,B(61))
0
1 ,(EC,R(66))
C
DATA RCURV / 2., 4., 6., 10., 20., 30., 40., 50., 60.,
1 RCUPV / 0., 100., 130., 200.,
2 RCURV / 0.17, 0.37, 0.47, 0.6, 0.73, 0.782, 0.81, 0.83, 0.84,
3 0.855, 0.865, 0.874, 0.874 /
1 DATA XRN / 0., 5., 7., 10., 20., 40., 100., 300., 600. /
2 YDRT1 / .07, .07, .05, .025, .5*0.C /
YDRT2 / .175, .175, .165, .12, .05, .03, .015, .005, 0.0 /
C
SREF = A(11)
SPLAN = C(4)
CB = C(11)
CLM = B(50) * 57.2057P * SREF/SPLAN
PC2X = C(1) - C(1)*C(2)
FKL1 = 0.0
DCL1 = 0.0
FKL2 = 0.0
DCL2 = 0.0
CLM0 = CLM
CLM2 = R(119) * 57.2957F * SREF/SPLAN
RRAR = 0.0
C
CLDB = R(106)
CALL LNTP(CLDB, AKB, XCLDB, YAKP, 7, 2)
FAKB = 1.24 - .04 * RNOFT * (E * 1.0E-6)
IF(FAKB.LT.1.0) FAKB = 1.0
AKB = FAKB * AKB * SREF/SPLAN
C
PX = -1.41 + 1.442*TP - 1.26*TR**2 + .528*TF**3
CX = 0.7125 - 1.497*TP + 1.476*TP**2 - .6909*TR**3
ED = 1.0 + BX*DCB + CX*DCP**2
C
FMACH = SPEED
FMCRP = B(114)
FML1 = R(115)
FML2 = R(116)
IF(FMACH.GT.FMCRP) FMACH = FMCRP
IF(SPEED.GE.FMCRP.AND.SPEED.LT.FML2) CLMC = B(118) *
1 57.2057F * SREF/SPLAN
IF(SPEED.GT.1.0) CALL ACPL(SPEED, CLCPT)
IF(SPEED.LT.FML2) CALL ACCL(FMACH, CLOFT)
C
DO 100 I = 1, NPALS
C
NI = SUR(I,1)
TC = SUR(I,3)
RLECC = AA(NI) + ER(NI)*TC + CC(NI)*TC*TC + DD(NI)*TC**3
IF(NI.EQ.8) RLECC = RLER *(TC/TOCR)**1.5

```

```

RLE = FLECC * SUP(1,20)
IF( ISWP.FC.CC.1.NE.NPMLS ) GO TO 50
RLECC = AA(NI) + PP(NI)*TCCS + CC(NI)*TGCS**2 + [C(NI)]*TCCS**3
TC = TCCS
IF (NI.EC.8) RLECC = RLECC * (TC/TOCR)**1.5
RLE = FLECC + CRAR2
C
C 90 CONTINUE
RNLER = RLE * PNOFT/10.0**3
CCTANS = 5. - 6.511 * SLP(1,4)
IF (SUR(J,4).GT.0.35) CCTANS = 1./TAN(SUP(1,4))
OMEGA = RNLER * CCTANS * SCPT(1.-(FMACH* CCS(SUR(J,4)))**2 )
C
CALL LINTP(OMEGA, PT, OCUFV, SCUPV, 13, 2)
CALL LINTP(SUR(J,4), RMIN, XSLP, YRMIN, 11, 2)
IF(RT.LT.RMIN ) RT = RMIN
DRT = 0.0
C
CALL LINTP(RNLER, CRT1, YRN, YDRT1, 9, 2)
CALL LINTP(RNLER, CRT2, YRN, YDRT2, 5, 2)
IF( TC.GT.C.CC ) DRT = DRT1 * (TC - C.CC)/C.CC
IF( TC.GT.0.06 ) DRT = DRT1 + DRT2 * (TC - C.06)/C.06
RT = RT + DRT
C
RI = RT + (0.824 - RT) * (CLD + CONCL)/C.6
C
IF(RI.GT.0.874 ) RI = 0.874
C
YW1 = SUP(J,12)
YW2 = C(1)
IF(NPMLS.FC.2.AND.I.EC.1) YW2 = SUR(2,12)
PCAR = PCAR + RI * (YW2 - YW1)/BC2X
C 100 CONTINUE
AT = AR * TF/COS(SWPLF)
DELR = 0.0482*AT - 0.01107*AT**2 + 0.001107*AT**3
1 - 0.0004833*AT**4
C
R = PCAR + DELR
C
FK = ((1.-R)/CLM + R/(3.14159*AR*EC)) * SREF/SPLAN
C
EF = 1./(3.14159 * AR * FK) * SREF/SPLAN
LFLCL = (LCPT * (1.-EF)
PRIMEK = C.518/SCPT(AR)
C
SWPC4 = SWPC4 + 57.2657P
IF( SWPC4.GT.FC.CC ) SWPC4 = FC.0
EP35 = DLNI(SWPC4, TF, XSWP, YTP, FEF35, 7, 4, 7, 2, 2)
EP7 = DLNI(SWPC4, TR, XSLP, YTR, FEF7, 7, 4, 7, 2, 2)
IF( AR.LE.3.5 ) EP = 1. - (1.-EP35) * AP/3.5
IF( AP.GT.3.5 ) EP = EP35 + (EP7 - EP35)*(AR-3.5)/2.5
C
EPP = EP * (1.-QNR**2)
AKD = 1./(3.14159 * AR * EPP) * SREF/SPLAN
C
C
RD = R
H = 1.1
ARTANS = AR * TAN(SWPLF)
IF( ARTANS.GT.3.5 ) H = 1.1 + 0.1*(ARTANS - 3.5)
IF( SPEED.GE.1.0 ) PRIMEK = H/CLM * SREF/SPLAN
C
IF( SPEED.LE.FML2 ) GO TO 200
ZNDM = 12. * (CCS(SWPLF)**1.6) * (SPEED-FMCPD)
FNOM = 1./(1. + ZNDM + 7NDM**2)
R = RD + FADP
FK = ((1.-R)/CLM + R/(3.14159*AR*EC)) * SREF/SPLAN
DELCL = CLCPT
C
GO TO 500
C
C 200 CALL KGIN(CLDB, FK, DELCL, SPEED, FKL1, DCL1)
IF( SPEED.LE.FML1 ) GO TO 500
C
C DRAG PCLAP IS CALCULATED BY LINEAR INTERPOLATION BETWEEN THE
C LIMITS FML1 AND FML2.

```

```

C
ZNDM = 12. * (COS(SWPLF)**1.6) * (FML2 - FPCFC)
FNOM = 1./(1. + ZNDM + ZNDM**2)
RL2 = FC * FNOM
FKL2 = ((1.-FL2)/CLM2 + FL2/(3.14159*AR*EC)) * SREF/SPLAN
CALL ADCL(FML2, DCL2)

C
CALL KGTN(0.50, FK, DELCL, FML1, FKL1, DCL1)

C
FK = FKL1 + (FKL2-FKL1)*(SPEED-FML1)/(FML2-FML1)
DELCL = DCL1 + (DCL2-DCL1)*(SPEED-FML1)/(FML2-FML1)

C
500 IF( KPRINT(22).EQ.0 ) RETURN
WRITE(6,1000) SPEED, FPACH, FK, DELCL, PETHK, AKD, AKP,
1 CLCP, CLY, SREF, SPLAN, FPCFC, FML1, FML2,
2 PLE, PLEP, CMGA, P, RPF, RT, CLC, CONCL, EF, AP, SWPC4,
3 FP, DOR, SWPL, FKL1, DCL1, FKL2, DCL2, RMIN

C
1000 FORMAT(5X, 'CDL1 DUMP' / (1X, 7F15.5) )
RETURN
END

```

```

C      SUBROUTINE ADCL(SPEED, CLOPT)
C      POLAR AXIS DISPLACEMENT
C      COMMON /INPUT/  A(3001)
C      COMMON /OUTPLT/ R(223)
C      COMMON /CALC/   C(50)
C      COMMON /BLKDAT1/ F(1411)
C      COMMON /BLKDAT2/ E(100)
C
C      CONCL = A(27)
C      SPLAN = C(4)
C      SREF  = A(11)
C      FCC   = C(26)
C      CLC   = C(13)
C      SWPLE = C(25)
C      TFCH  = A(26)
C
C      CLOPT = 0.0
C      IF (SPEED .GE. 1.0) GO TO 10
C
C      IF (CLC.GT.0.0) CLOPT = -.001 +16.934*FCC -216.2697*FCC**2
C      +17.13562*FCC**3
C      1 IF (CONCL.GT.0.0)CLOPT = CLOPT -.0017 +1.1334*CONCL
C      -1.6499*CONCL**2 +1.0665*CONCL**3
C      1 IF (TFCH.GT.C.) CLOPT = 0.51951 * CLC**0.75
C      GO TO 20
C
C      10 BETAT = 10.
C      IF (SWPLE.GT.0.) BETAT = SCRT(SPEED**2 -1.)/TAN(SWPLE)
C      DELCL = CLC * (0.25 - 0.225 * BETAT )
C      IF ( BETAT .GE. 1.11 ) DELCL = 0.0
C      CLOPT = DELCL
C
C      20 CLOPT = CLOPT + SPLAN/SREF
C      RETURN
C      END

```

```

C      SUBROUTINE KGIN(CLDB, AKIN, DECLIN, SPEED, AKGUT, DCLCUT)
C
C      COMPUTES POLAR USING LEAST-SQUARES CURVE FIT
C
C      COMMON /PLKPR1/ KPPINT(50)
C      DIMENSION CL(11), CCL(11), SA(12), RE(11)
C
C      DCL = CLDB/10.0
C      CL(1) = 0.0
C
C      DO 20 I = 1, 11
C      IF( I.NE.1 ) CL(I) = CL(I-1) + DCL
C      CLX = CL(I)
C
C      CALL CDDF(CLX, SPEED, CDR)
C
C      IF( I.EQ.1 ) CDR0 = CDR
C      CCCR = CDR - CDR0
C
C      40 CCL(I) = AKIN + (CL(I) - DECLIN)**2 + CCCR
C      20 CONTINUE
C
C      CALL LSQCF(CL, CCL, 11, 2, SA, RE, SIGMA, IT)
C
C      AKGUT = SA(1)
C      DCLCUT = -SA(2)/(2.0 + AKGUT )
C      ERR = SA(3) - AKGUT + DCLCUT**2
C
C      IF( ABS(ERR).LT.0.001.CP.KPPINT(14).EQ.0 ) GO TO 30
C
C      WRITE (6,1000) SPEED,ERR,AKGUT,DCLCUT,
C      AKIN, DECLIN, ( CL(I), CCL(I), I = 1,7 )
C      1000 FORMAT(10X,*KGIN* 5X,*PACH =*F8.5,2X*ERR =*F8.5,2X,4F8.5, /
C      1 1X,14F8.5 )
C
C      30 RETURN
C      END

```

```

C      SUBROUTINE CDL2(SPEED, CL, AEROK, DELCL, PRIMEK, AKD, AKP, CCL)
C
C      COMPUTES DRAG DUE TO LIFT
C
C      COMMON/INPUT/  A(30F1)
C      COMMON /OUTPUT/ P(22F)
C      COMMON /CALC/   C(F)
C      COMMON /BLKDAT1/ D(14I1)
C      COMMON /BLKDAT2/ E(90R)
C      COMMON /BLKPR1/ KPRINT(50)
C
C      CLPB = P(105)
C      CLDR = F(106)
C
C      20 CONTINUE
C      CDL = AEROK * (CL - DELCL)**2
C      IF( CL.LE.CLPB ) GO TO 500
C
C      DRAG DUE TO LIFT ABOVE POLAR BREAK
C      CDL = CDL + PRIMEK * (CL - CLPB)**2
C      IF( CL.LE.CLDB ) GO TO 500
C
C      DRAG DUE TO LIFT ABOVE DRAG BREAK (CLDB)
C      IF( SPEED.GE.1.0 ) GO TO 410
C      CDPDB = AEROK * (CLDB - DELCL)**2 - AKD * CLDB**2
C      1      + PRIMEK * (CLDB - CLPB)**2
C
C      DCCB = AKR * (CL - CLDR)**2
C      FCCB = FCCP + 0.08 * SQRT(DCCB)
C
C      CDL = CDPDB + FCCB + AKD * CL**2
C
C      400 IF( SPEED.LT.1.0 ) GO TO 500
C      410 CONTINUE
C      CDL = (AEROK - PRIMEK) * (CLPB - DELCL)**2
C      1      + PRIMEK * (CL - DELCL)**2
C
C
C      500 CONTINUE
C      IF( KPRINT(14).LE.0 ) GO TO 50
C      WRITE(6,1000) SPEED, CL, CDL, AEROK, DELCL, PRIMEK, CLPB,
C      1      CLDB, AKD, AKP
C      KPRINT(14) = KPRINT(14) - 1
C      50 CONTINUE
C      RETURN
C      1000 FORMAT (1X, *CDL2 DUMP*/(1X, 7F15.5 ) )
C      END

```



```

SUBROUTINE AERA(SPEED, CL, ALPHA)
C
C
C   CALCULATES ANGLE OF ATTACK
C
COMMON/INFUT/  A(30F1)
COMMON/CLTPUT/ B(223)
COMMON/CALC/    C(50)
COMMON/PLKPAT1/ D(1411)
COMMON/ELKCAT2/ E(20F)
COMMON/PLKPR1/ KPPINT(50)
C
DIMENSION CLTAP(13), ATAP(13), SLP(7,30)
DIMENSION YAB(6), YCP(8), FKVCFM(6,P), XANG(10), YRTDC(7), FFI(10,7),
1 XX(12), YY(7), XF(12,7), XIN13(10), YIN13(7), ZCUT13(10,7),
2 ZCUT14(10,7)
EQUIVALENCE (CLA,P(9)), (ALD,P(10)), (CLAW,P(106)),
1 (CLTP,P(106)), (CLMAX,P(107)), (DAMAX,P(109)),
2 (DZMAX,P(110)), (DFL,P(111)), (CLS,P(112)),
4 (ISWP,A(6)),
5 (KPNLS,A(3)), (TDC,C(12)), (TR,C(30)),
6 (SUR(1,1),A(1P4)), (YIX,C(31)), (YCX,C(44)),
7 (SWPFC,C(25)), (SWPMC,C(7)), (SWPTE,C(8)),
8 (AF1,C(23)), (AP,C(24)), (FF1,C,P(113)), (SREF,A(11)),
9 (SEXP,C(10)), (SWPR,C(27)), (CFI,P(46))
C
EQUIVALENCE (YAP(1),E(4F4)), (YCP(1),E(4C6)), (FKVCFM(1,1),E(4G2)),
1 (XANG(1),E(54F)), (YRTDC(1),E(55F)), (FRA(1,1),E(563)),
2 (XY(1),E(56E)), (XY(1),E(57F)), (XF(1,1),E(525)),
3 (XIN13(1),C(10E?)), (YIN12(1),C(1073)), (ZCUT13(1,1),
4 D(1080)), (ZCUT14(1,1),D(1150))
C
IF SWEEP = SLP(1,1)
SWEEP = SWPR * 57.2957F
KPRT = KPPINT(17)
CLVM1 = C.0
C
10 ALPHA = CL/CLA + /LD
IF( SPEED.GT.1.0 ) RETURN
IF (KPNLS.EQ.1) GO TO 15
IF (SUP(1,4).LE.0.7) GO TO 15
SWC = SUR(2,4) + 0.0F7
IF (ISWP.FO.1) SWC = SWPR + 0.087
IF (SUP(1,4).LE.SWC) GO TO 15
GO TO 200
15 CONTINUE
C
IF( AR.LE.ARLD ) GO TO 100
C
20 CONTINUE
C
IF( KPRT.GT.0 ) WRITE(6,1000) CL, ALPHA, AP, ARLE, CLMAX, CLS,
1 DAMAX, CLA, ALC
C
IF( CL.LE.CLS ) GO TO 200
C
HIGH ASPECT RATIO LIFT METHODD *****
C
DA = C.C
DCL = CLMAX - CLS
DA = ((CL - CLS)/DCL)**2 + DAMAX
30 ALPHA = ALPHA + DA
IF( CL.LE.CLMAX ) GO TO 200
ALPHA = ALPHA + 5.0
IF( ALPHA.GT.90.0 ) ALPHA = 90.0
C
GO TO 200
100 CONTINUE
LOW ASPECT RATIO LIFT METHODD *****
C
AS = 0.C
Z = 2. * CCS(SWPMC) / AP
HCP = (AMAX - AS)/114.6 /AP * 1.5 + (TE + TE**2)/(1. + TR + TR**2)
XEP = 1.0014 - 1.069*HCP + 3.0021*HCP**2 - 2.0072*HCP**3
DCLFP = (Z + SCPT(1. + Z**2))/(XEP**2 + SCPT(1. + (YFP+Z)**2))
DCLPA = (DCLFP - 1.) * CLAW * 57.3
CLPA = CLA * 57.3
DCLC = - CLA * ALC

```

```

CLVM1 = CLMAX - CLPA * SIN(AMAX/57.3) * CGS(AMAX/57.3)**2 - DCLO
CLVM2 = CLVM1 - DCLPA * SIN(AMAX/57.3) * CGS(AMAX/57.3)**2
IF( CLVM1.LE.0.0 ) DAMAX = AMAX - (CLPA/CLC - ALG)
IF( CLVM1.LE.0.0 ) CLC = CLC * (AMAX - 2.*DAMAX - ALG)
IF( CLVM1.LE.0.0 ) GO TO 20
C
CLVA = 0.0
TCCR = 0.0
APETA = 0.0
IF (SWPLE.EC.0.0) GO TO 112
C
CO = TAN(SWTF)/TAN(SWPLE)
BETA = SCPT(1. - SPEFD**2)
IF (CC.NE.1.0) ABETA = 4./TAN(SWPLE) / (1.-CC) * BETA
TCCR = TCCR
IF ( ID.EC.9 ) TCCR = 0.0
C
110 CLVCFM = DLNT(APETA, CC, XAR, YCD, FKVCFM, (,R,6,2,2)
TS2 = TAN(SWPLF)**2
FM = SORT((1. + TS2)/(BETA**2 + TS2))
CLVA = CLVCFM * FM * SFXW/SREF
C
112 CONTINUE
C
ATAP(1) = 0.0
DO 120 I = 1, 13
IF( I.CT.1 ) ATAB(I) = ATAB(I-1) + 3.0
ANG = ATAB(I)* 0.01745
DCLP = 0.0
IF( ATAB(I).LE.AS ) GO TO 115
C
C
C TIP VORTEX EFFECT
HOB = (ATAB(I) - AS)/114.6 / AR * 1.5 * (TR+TF+TP)/(1.+TR+TR*TR)
XEP = 1.0014 - 1.969*HOB + 5.0021*HOB**2 - 2.0072*HOB**3
IF( HOB.LE.0.0 ) XEP = 1.0
DCLFP = (2 + SCRT(1.+2*Z))/(XLP*Z + SCRT(1. + (XEP*Z)**2))
DCLPA = (DCLFP - 1.) * CLAW * 57.3
DCLP = DCLPA * SIN(ANG)
C
115 CONTINUE
LEADING EDGE EFFECT (PA)
RA = DLNT(ANG, TCCF, XANG, YRTEC, FRA, 10, 7, 10, 2,2)
C
C
C VORTEX BREAKDOWN EFFECT (FVL)
X = AR
FVL = 0.0
IF( X.LT.3.37 ) FVL = DLNT(X,ANG, XX, XY, YF, 12, 7, 12, 4,2)
C
CLV = (1.-RA) * FVL * CLVA * SIN(ANG)**2 * CGS(ANG)
IF( CLV.GT.CLVM2 ) CLV = CLVM2
CLP = CLPA * SIN(ANG) * CGS(ANG)**2
CLTAB(I) = DCLP + CLV + CLP + DCLP
IF( KPINT(17).EQ.2.AND.I.EQ.1 ) WRITE(6,1002)
IF( KPINT(17).EQ.2 ) WRITE(6,1001) CLTAB(I), ATAB(I), RA,
1 CLP, CLV, DCLP
C
120 CONTINUE
C
IF( KPRT.GT.0 ) WRITE(6,1003) CL, ALPHA, AR, ARLC, 7, HCP, DCLPA,
1 CLPA, DCLO, CLVM1, CLVM2, TCCR, CLVA, RA, FVL,
2 ANG
C
125 IF( CLVM1.LE.0.0 ) GO TO 20
130 CALL LNTF(CL, ALPHA, CLTAB, ATAP, 13, 4)
C
GO TO 200
C
C
C CALCULATES ANGLE OF ATTACK AT HIGH LIFT BY MODIFIED
WINSTAN METHOD.
C
290 CONTINUE
300 BETA = SCRT(1. - SPEFD**2)
CETAN = BETA * TAN(SLR(1,4))
AA = DLNT(BETAN, SWEP, XIN13, YIN13, ZOUT13, 10, 7, 10, 2,2)
EN = DLNT(BETAN, SWEP, XIN13, YIN13, ZOUT14, 10, 7, 10, 2,2)
C
C

```

```

ACLARU = CLDB/CLA + ALD
EFRK = YIY/(YIX + YGX)
CLB = AA * ACLARU * FN * EFRK / AR * CLAW + 57.29578
31C IF( CL.LE.CLDB ) GO TO 200
CLT = CLB + CL - CLDB
ALPHA = (CLT * API / (AA * CLAW + 57.29578 * EFRK)) * (1./FN)
C
1 IF( KPRT.GT.0 ) WRITE(6,1004) CL, ALPHA, AR, A, EN, ACLARU,
EFRK, CLB, CLDB, CLAW
C
200 CONTINUE
IF( ALPHA.LT.6.0 ) GO TO 205
COLA = CL * TAN(ALPHA/57.296)
IF( COLA.LT.CCL ) CCL = COLA
205 RETURN
C
1000 FORMAT(5X, *AERA DUMP HIGH ASPECT RATIO* / (1X,7F15.5) )
1003 FORMAT(5X, *AERA DUMP LOW ASPECT RATIO* / (1X,7F15.5) )
1004 FORMAT(5X, *AERA DUMP CRANKED WING* / (1X,7F15.5) )
1001 FORMAT(5X, *VORTEX LIFT*, 6F15.5)
1002 FORMAT(//5X, *VORTEX LIFT*, 5X, *CL*, 12X, *ALPHA*, 13X, *R*, 13X,
1 *CLP*, 11X, *CLV*, 11X, *DCLF*, / )
40C ENTRY AFRA1
KPRT = KPPINT(17) - 1
ALPHA = CL/CLA + ALD
IF( SPEEF.GT.1.0 ) RETURN
IF( AR.LE.ARLC.AND.CLV*1.GT.0.0 ) GO TO 125
IF( NPPLS.FC.1 ) GO TO 20
IF( SUR(1,4).LE.0.7 ) GO TO 20
IF( SUR(1,4).LE.SVF ) GO TO 20
GO TO 310
C
END

```

```

C      SUBROUTINE AFTCC(ALPHA, CDFT)
C
C      COMPUTE DRAG INCREMENT DUE TO FUSELAGE LPSWEEP
C
COMMON/INPUT/  A(30F1)
COMMON/OUTPUT/ B(223)
COMMON/CLC/    C(5C)
COMMON/BLKDAT1/ D(1411)
COMMON/BLKDAT2/ F(90F)
C
C      DIMENSION X(6), Y(4), Z(6,4)
C
DATA X / -5., 0., 5., 10., 15., 20. /,
1     Y / 0., 5., 10., 15. /,
2     Z / 2*0., .01, .045, .125, .26, 0., .005, .013, .054, .142,
3     .33, 0., .004, .0175, .065, .172, .41, 0., .014, .023,
4     .065, .215, .53 /
C
WINC = C(39)
PFUS = A(26)
AP = A(29)
IF( AR.EC.U.O ) AP = 1.0
Pdy = A(79)
SREF = A(11)
C
CDFT = 0.0
IF( PFUS.EC.U.O ) RETURN
AFUS = ALPHA - WINC
PMA = PFUS - AFUS
C
DCC = BLNT/PMA, AFUS, X, Y, Z, 6, 4, 6, 4, 2)
CDFT = DCC + PMA*X/SREF * AP/1.5
C
RETURN
END

```

```

SUBROUTINE CDRC(SFFED, AEROK, DELCL, CDC)
C
C
C
C
COMMON /INPUT/ A(30*1)
COMMON /CALC/ C(50)
EQUIVALENCE (A9,C(29)), (SEYV,C(10)), (SPLAN,C(4)), (SPEF,A(11))
CDC = 0.0
AR = AR + SPAN/SPEF
F = 1.0/(3.14159*AR*SREF*AEROK)

C
C
CREF1 = 1.0/(3.14159*AR*SREF*(1.0-E))
CREF2 = C.7 + SEYV/SREF
IF (SEYV.GE.1.0) GO TO 15
IF (SPEF.DC.1.0) GO TO 5
IF (CREF1.GT.CREF2) GO TO 15
5 CDC = CREF1 * DELCL**2
GO TO 10
15 CONTINUE
CDC = CREF2 * DELCL**2
C
10 RETURN
END

```

```

SUBROUTINE CMCW(SPEED, CMC)
C
C COMPUTES ZERO LIFT PITCHING MOMENT OF WING
COMMON/INPUT/ A(3081)
COMMON/CLIPLT/ B(223)
COMMON/CALC/ C(50)
COMMON/BLKDAT1/ D(1411)
COMMON/BLKDAT2/ E(908)
COMMON/BLKPRN/ KFPINT(50)
C
DIMENSION CMC4(20), CAM(2), SEX(2), XRC4(5), YRC4(5), ZRC4(3),
1 FRC4(5,5,3)
EQUIVALENCE (FMC4,R(114)), (TS,A(1F4)), (SEXP,C(14)),
1 (TCC,C(12)), (TWIST,C(3E)), (SEXP,C(1C)),
2 (CAM(1),A(191)), (SEX(1),A(2C3)),
3 (NPNLS,A(3)), (SWPCC,C(6)), (CR),C(1F)), (CTX,C(16)),
4 (XRC4(1),D(589)), (YRC4(1),E(594)), (ZRC4(1),D(599)),
5 (FRC4(1,1,1),E(6C2))
DATA CMC4 / 4*-0.25, 3*-0.21, -0.3, 0.0, 11*-0.2066 /
C
C ID = TS
CMCS = 0.0
C
DO 100 I = 1, NPNLS
CMCS = CMCS + CAM(I)*SEX(I) + CMC4(ID)
100 CONTINUE
C
CMC = 0.0
IF (SPEED.GE.1.) RETURN
C
CMDS = CMCS/SXPV
C
CMDB = APXR * COS(SWPCC)**2/(ARXR + 2.*COS(SWPCC)) + CMDS
C
TRY = CTX/CRX
ESWCC = SWPCC
CALL TLNT(ESWCC,APXR,TPX, CMCT, XRC4,YRC4,ZRC4,FRC4, 5,5,3, 5,5)
CMCT = CMCT * TWIST
C
FMACH = SPEED
IF (FMACH.GT.FMCR) FMACH = FMCR
CMACH = (1. + 5.*TCC*FMACH**5)/SQRT(1.-(FMACH*COS(ESWCC))**2)
C
CMG = (CMDB + CMCT) * CMACH
C
IF (KFPINT(25).GT.0) WRITE(6,1000) CMC, CMDB, CMCT, CMACH,
1 CMCS, ARXP, TCC, TWIST, ESWCC, TPX, CMCT, FMCR,
2 CTX, CRX, SWPCC, SEXP, CMC4(ID),
3 TS, (CAM(I), SEX(I), I = 1, NPNLS)
C
1000 FORMAT(5X, *CMW CUMP*/(1X,7F15.5) )
C
RETURN
END

```

```

SUBROUTINE WBAC(SPEED, XACR)
C
C
C
C
COMMON /INPLT/ A(2061)
COMMON /OUTPUT/ P(222)
COMMON /BLKPRN/ KPRINT(50)
COMMON /CALC/ B02, FCR, CR, SPLAN, YCRTE,
1 SWPCC, SWPMC, SWPTE, SWPT, SEXW, CPAR,
2 TCC, CLD, ARXF, CRX, CTX, XLF, YTX,
3 YFX, YTX, XP, YE, XH, CMFGA, SWPLE, FCP,
4 SWPP, KPASS, AP, TR, YIX, SIY, AR1, CBXP,
5 SCXP, APCP, DXCC, TWIST, WING, XWT, TCCS, CPAR2, CLDS,
6 YGX, SWPMCS, DA1, DAC1, DA2, CAC2, TTC

DIMENSION XDB(8), YDOB(8), XBDGL(5), YADM(6), FCP(5,6)
DIMENSION SUP(7,30)
EQUIVALENCE (SUP(1,1), A(184)), (NPALS, A(2)), (XACS, P(130))
1 , (ISWP, A(6)), (NBRCYS, A(1))

C
C
C
C
DATA XDB / 0., .05, .1, .15, .2, .3, .4, .5 /,
1 YDB / 0., .1, .154, .19, .219, .266, .3, .33 /

C
C
C
C
DATA XBDGL / 0.4, .7, 1.0, 1.25, 1.67 /,
1 YACN / 0., .4, .8, 1.2, 1.6, 2.0 /,
2 FCP / .54, .535, .525, .516, .5, .42, .435, .45, .46, .46,
3 .35, .377, .4, .414, .425, .265, .33, .355, .375, .394,
4 .246, .285, .32, .345, .375, .21, .25, .288, .315, .34 /

C
C
C
C
AR1 = ARXR
TR1 = CTX/CRX
TW = A(184)
FMCPO = P(114)
ELN = A(93)
CLAB = P(97)
CLAW = B(96)
SWPLEI = SUR(1,4)
SWPLED = SUR(2,4)
SWPMCI = SLR(1,22)
SWPMCC = SUR(2,22)

C
C
C
C
IF (NPALS.GT.1) AR1 = ADI
IF (NPALS.GT.1) TR1 = SUR(1,9)/CRX
IF (ISWP.EQ.0) GO TO 20
SWPLEC = SWPR
SWPMCG = SWPMCS

C
C
C
C
20 CALL ACCR(SPEED, AP1, SWPLEI, SWPMCI, TP1, SIY, TCC, TW, FMCPO, YACR, CLAI,
1 XACRX = XACR
XACSW)
IF ( NPALS.EQ.1 ) GO TO 100

C
C
C
C
CRANKED WING PLANFORMS
TR2 = CTX/CBXP

C
C
C
C
CALL ACCR(SPEED, AP2, SWPLED, SWPMCC, TP2, SCXP, TCC, TW, FMCPO,
1 YACRF, CLACP, XACSGP)

C
C
C
C
YACRO = XACRP * CBXP/CRX - YIX*0.5*TAN(SWPLED)/CRX
1 + YIX * TAN(SWPLEI)/CRX
YACSO = XACSGP * CBXP/CRX + YACRE - YACRF*CBXP/CRX

C
C
C
C
YACR = (CLAI * SIX * YACRX + CLACP * SCXP * XACRE) /
1 (CLAI * SIX + CLACP * SCXP)

C
C
C
C
YACSW = (CLAI * SIX * YACRX + CLACP * SCXP * XACSC) /
1 (CLAI * SIX + CLACP * SCXP)
100 XACWR = XACR

C
C
C
C
WING-BODY COMBINATION
XLE IS THE X-STA. OF L.F. OF INCREASED EXPOSED WING FOOT
DIA = DOR * B02 * 2.

```

```

C      IF( SPEED.GE.1.2 ) GO TO 200
SUBSONIC CALCULATION OF XACN (NOSE) AND XACPW (WING CARRY-OVER)
XLEO  = PLN + 1.6 *(XLE - PLN)
XACN  = -0.54 * YLEO/CRX
C
C      PARE  = ARXR * SCRT(1.-FMCRC**2)
IF( SPEED.LT.FMCRC ) BAPE = ARXR * SCRT(1.-SPEED**2)
CALL LNTF(FOP, FOCR, YCF, YOCR, P,4)
XACBW = 0.25 + DYOC * FOCR/CPX
C
C      IF( BAPE.GE.4. ) GO TO 190
XACBWO = C.125 * ARXR * TAN(SWPLEI) + (1.+CTX/CRX)
C
190  XACPW = (XACBWO - XACBW) * (PARE -4.)**2/16. + XACBW
XACBW1 = XACBW
XACN1  = XACN
C
C      IF( SPEED.LE.FMCRC ) GO TO 290
C
C      SUPERSONIC CALCULATION OF XACN ^ XACBW
200  BETA  = 0.663225
XACN    = 0.
XACBW   = 0.
IF( NBCDYS.EC.0 ) GO TO 290
IF( SPEED.GT.1.2 ) BETA = SCRT(SPEED**2 -1.)
C
C      AGN   = (XLE - BLN)/BLN
IF( AGN.LT.0. ) AGN = 0.
BDCL    = BETA *DIA /RLN
C
C      FIGURE 4.2.2.1-23A DATCOM *****
XCPCL   = DLNT(BDCL, AGN, XBDCL, YACN, FCP, 5,6,5, 2,2)
XACN    = XLE/CRX * (XCPCL -1.)
C
C      BQGC  = BETA * DIA/CRX
PCCT    = BETA * TAN(SWPLEI)
A1      = 0.5845
IF( PCCT.GE.1. ) A1 = C.39F5 + G.00214*ALCG(PCCT)
IF( PCCT.LT.1.3.AND.PCCT.GT.0.1 ) A1 = C.39F5 + C.00607*ALCG(PCCT)
C
C      FIGURE 4.3.2.1-27A DATCOM *****
XACPW   = C.5 + A1*PDCC -.1057*PDCC**2 +.C172*PDCC**3
IF( SPEED.GE.1.2 ) GO TO 290
C
C      XACN  = XACN1 + (XACN -XACN1) *(SPEED-FMCRC)/(1.2-FMCRC)
XACBW    = XACBW1+ (XACBW-XACBW1)*(SPEED-FMCRC)/(1.2-FMCRC)
C
290  CONTINUE
FKRW    = C.0004 +1.2662*DCP +.601**DCP**2 +.1263*DCP**3
FKWP    = 1.0025 +.7116*DCP +.42*DCP**2 -.1366*DCP**3
CLAFW   = FKWB/(FKTW + FKWB) * CLAW
CLAWB   = FKWB/(FKWB + FKWR) * CLAW
C
300  XACP  = (XACN * CLAB + XACWR * CLAWB + XACPW + CLARW)/
1        (CLAB + CLAW)
C
XACS    = (XACN * CLAR + XACSW * CLAWB + XACPW + CLARW)/
1        (CLAR + CLAW)
C
C      IF( KPRINT(IP),EO.0 ) GO TO 400
WRITE(6,1000) XACP, XACN,CLAP, XACWB,CLAWB, XACBW,CLAWB,
1          XACS, SPEED, XLEO, FOP
C
400  RETURN
1000  FORMAT(5X, *WBAC DUMP* / (5X,6F15.5) )
END

```





```

C      IF( VBAR.GE.1.0.OR.VBAR.LE.-2. ) GO TO 40
C      TRANSONIC AERODYNAMIC CENTER
C      CALL TLNT(ABAR,VRAR,ARTSW,VAL(1),YPC6,YBC6,ZPC6,FPC6,4,4,6,4,4)
C      CALL TLNT(ABAR,VRAR,ARTSW,VAL(2),YPC6,YBC6,ZPC6,FPC6,4,4,6,4,4)
C      CALL TLNT(ABAR,VRAR,ARTSW,VAL(3),YPC6,YBC6,ZPC6,FPC6,4,4,6,4,4)
C      CALL LNTP(TR, XAC2, YVAL, VAL, 3, 2)
C
40 XACR = XAC1
   IF( SPEED.LE.FMCRG.OR.SPEED.GE.1.2 ) GO TO 50
   FM1 = FMCRG + G.05
   FM2 = SQRT(1.+ TCC+.66667)
   XACR = XAC2
C
   IF( SPEED.GT.FM1.AND.SPEED.LT.FM2 ) GO TO 50
   IF( SPEED.LE.FM1 ) XACR = XAC1 + (XAC2-XAC1)*(SPEED-FMCRG)/.05
   IF( SPEED.GE.FM2 ) XACR = XAC2 + (XAC1-XAC2)*(SPEED-FM2)/
1      (1.2 - FM2)
C
50 APHI = 2.* APIC
   XACS = (1.+2.*TF)/12.* ARTSW + (1. + TF+.2/(1.+TR))/3.
   IF( AP.GT.APHI ) GO TO 60
   CALL LNTP(DY, YACS1, XDY, YAC, 6, 2)
   XACS = XACS1 + (1.+2.*TF)/17.544 * ARTSW -G.2
C
60 IF( KPRINT(18).FO.2 ) WRITE(4,100) SPEED, AR, SWPIE, SWPMC,
1      TR, SPLAN, TCC, TH, FMCRG, XACR, CLAX,
2      SB, ARTSW, YAC1, ABAR, VBAR, VAL, XAC2,
3      FM1, FM2, APHI, XACS
C
1000 FORMAT(5X, *ACCR DUMP* / (5X,6F15.5) )
C
      KPRINT(11) = KPR11
      RETURN
      END

```

OVERLAY (4,0)  
PROGRAM NUCCAM

C  
C

CALCULATES NUCLEAR DAMAGE AERODYNAMIC EFFECTS

```
COMMON /INPUT/ A(2001)
COMMON /OUTPUT/ F(223)
COMMON /CALC/ C(50)
COMMON /PLKPRF/ KPRINT(50)
COMMON /SLKTIU/ TITLE(A)
COMMON /NLCOU/ DAMCD(7,17), SYM(7,17), BCDPIN,
1 ALP(21), CL(21), CD(21), CM(21), CLT(21), CDT(21),
2 CLP(21), CDP(21), CMP(21), CLTP(21), CDTP(21),
3 FCL(6), DXAC,
4 CFGLL(21), CERT(21), X2(2), X4(2)
DIMENSION BNAME(7), BCD(7,20), SNAME(7), SLR(7,30), FMSURV(20),
1 ALT(20), NSUR(7,100), NSUR(7,100), IDAM(20)
2 EQUIVALENCE (NPEYS,A(1)), (NNAC,A(2)), (NALS,A(3)),
1 (NHT,A(4)), (NVT,A(5)), (SHEP,A(11)),
2 (METEF,A(10)), (RCUGPK,A(16)), (BNAME(1),A(30)),
3 (PCD(1,1),A(37)), (SNAME(1),A(177)),
4 (SRE(1,1),A(164)), (NSURV,A(491)),
5 (FMSURV(1),A(403)), (ALT(1),A(513)),
6 (CDP(1,1),A(1642)), (PSLR(1,1),A(2362))
7 (IDAM(1),A(1642)), (FWING(1),A(2062))
8 (CPSLR(1,1),A(154)), (BC2,C(1))
9 (C,C(2)), (CBAQ,C(11)), (TGCW,C(12)),
1 (CTX,C(16)), (CGRCD(1,1),P(160))
```

C

```
COMMON /ELKDAT3/ XFACH3(17), CMK3(17),
1 XCH3(6), HCX3(5), CIG3(6,5),
2 XCF3(6), LCD3(4), XF3(6,4),
3 YMACH2(13), CPF3(13),
4 ZFACH3(13), CPA3(13),
5 XICP4(12), HMX4(7), CDF4(12,7),
6 P/TIG3(6), LMACH(10), DKAG2(8,10),
7 XLCD(6), VMACH(6), CDF4(6,6)
NPGDS = NPGDYS + NNAC
NSUR = NPNS + NHT + NVT
```

C

C

```
ZERO OUT THE DAMCD(I,K) ARRAY
DO 6 I = 1,7
DO 2 K = 1,17
DAMCD(I,K) = 0.0
SYM(I,K) = 0.0
2 CONTINUE
6 CONTINUE
```

C

C

DETERMINE IF DAMAGE TO AIRCRAFT IS SYMMETRICAL

```
DO 8 J = 1,7
DO 7 I = 1,61,10
K = I + (J-1)/10
IF (DAMCD(I,J).LT.0.0) SYM(J,K) = 1.0
IF (DAMCD(I,J).LT.0.0) SYM(I,K+10) = 1.0
IF (DAMCD(I,J+1).LT.0.0) SYM(J,K+10) = 1.0
DAMCD(I,J) = ABS(DAMCD(I,J))
DAMCD(I,J) = ABS(DAMCD(I,J))
DAMCD(I,J+1) = ABS(DAMCD(I,J+1))
7 CONTINUE
8 CONTINUE
```

C

C

REWIND 10

C

```
DO 500 J = 1, NSURV
XMACH = FMSURV(J)
IF (KPRINT(28).EQ.1)
1 WRITE(6,5000) XMACH
```

C

```
READ (10) CL, CD, CM, ALP, CLT, CDT
READ (10) R
```

C

```
AAA = ABS(B(144)-YMACH)
IF (AAA.GT.C(61) .OR. IGT(6,1001)
1001 FORMAT(/,20), * INCORRECT FACH NUMBER WAS READ FROM *,
1 *TAPE UNIT NUMBER 10 * )
```

```

C
C      RNDFT = B(146)
C
C*****          MODE 1          *****
C*****          DRAG DUE TO ROUGHNESS ON BODIES *****
C
C      10 IF (IDAM(1).NE.1) GO TO 20
C
C      11 DO 12 I = 1, NPCDS
C          RPR = DRCC(I,1) + DECD(I,2)
C          IF (BRB.EQ.0.0) GO TO 12
C          CALCULATE DRAG OF UNMAGED BODIES
C          CALL CFK3 (IFACTH, RN/FT, YCL)1, YCL)2, K1, K2, K3, MACH, CF)
C          CALL CFK3 (PCD(I,1), RACFT, C.O, O.O, CRCT(I,1),O,G, YMACT, CF)
C          CDT = CF * PCD(I,4) / SREF
C          CALCULATE DRAG OF DAMAGED BODIES
C          CALL CFK3 (DRCD(I,1), RACFT, DECD(I,3), DRCD(I,4), DBCD(I,1),
C          DRCD(I,2), DRCD(I,1), XMACH, CF)
C          CPF = CF * FOC(I,4) / SREF
C
C          CALCULATE FORM FACTOR (FF) FROM SURVEY DRAG RESULTS
C          FF = (CDRCD(I,1) + CDRCD(I,2)) / CDRCD(I,1)
C
C          DAMCD(I,1) = (CDF - CDT) * DRCD(I,5) * FF
C          IF (DECD(I,5).GT.1.0) DAMCD(I,1) = (CDF - CDT) * DRCD(I,5)
C          * FF / PCD(I,4)
C          1 IF (KPOINT(28).EQ.1)
C            1WRITE(6,5011) J, J, CDT, CDF, CF, DAMCD(I,1)
C          12 CONTINUE
C
C*****          MODE 2          *****
C*****          DRAG DUE TO FWD-FACING STEPS ON BODIES *****
C
C      20 IF (IDAM(2).NE.1) GO TO 30
C          CALL LNTF (XMACH, CPF, YMACH3, CPF3, 13, 4)
C          DO 23 I = 1, NPCDS
C          SUMO = 0.0
C          IF (DRCD(I,11).EQ.0.0) GO TO 22
C          IF (DECD(I,11).EQ.1.0) DXCL = 0.0
C          IF (DRCD(I,11).GT.1.0) DXCL = (DECD(I,15) - DECD(I,14))
C          / (DRCD(I,11) - 1.0)
C          1 GO 21 SUMS C)FFF / C)FREESTREAM FOR ALL STEPS
C          N = DRCD(I,11)
C          DO 21 K = 1, N
C
C          YCL = DRCD(I,14) + DXCL * (K-1)
C          XI = XCL * FOC(I,1)
C          HEIGHT = DRCD(I,13)
C          CALL EFFC(RN/FT, XI, XMACH, HEIGHT, CEFF)
C          SUMC = SUMC + CEFF
C          IF (KPOINT(28).EQ.1)
C            1WRITE(6,5021) J, I, K, XI, CEFF, CPF, SUMC
C          21 CONTINUE
C
C          22 DAMCD(I,2) = CPF * SUMO + DRCD(I,12) * DECD(I,12) / SREF
C          IF (KPOINT(28).EQ.1)
C            1WRITE(6,5022) DAMCD(I,2)
C          23 CONTINUE
C
C*****          MODE 3          *****
C*****          DRAG DUE TO AFT-FACING STEPS ON BODIES *****
C
C      30 IF (IDAM(3).NE.1) GO TO 40
C
C          CALL LNTF (XMACH, CPA, YMACH3, CPA3, 13, 4)
C          DO 33 I = 1, NPCDS
C          SUMO = 0.0
C          IF (DRCD(I,21).EQ.0.0) GO TO 32
C          IF (DECD(I,21).EQ.1.0) DXCL = 0.0
C          IF (DRCD(I,21).GT.1.0) DXCL = (DECD(I,25) - DRCD(I,24))
C          / (DRCD(I,21) - 1.0)
C          1 GO 31 SUMS C)FFF / C)FREESTREAM FOR ALL STEPS
C          N = DRCD(I,21)
C          DO 31 K = 1, N

```

```

XCL = DRDD(I,24) + DXCL * (K-1)
XI = XCL + PDP(I,1)
HEIGHT = CEGD(I,23)
CALL EFFO(FNOFT, YI, XMACH, HEIGHT, CEFF)
SUMO = SLMO + CEFF
IF (KPRINT(26).EQ.1)
  WRITE(6,5031) J, I, K, XI, OEFF, CPA, SLMO
31 CONTINUE
32 DAMCD(I,3) = -CPA + SUMO + (BCD(I,22) + CBCD(I,23)/SREF
  IF (KPRINT(28).EQ.1)
    WRITE(6,5032) DAMCD(I,3)
33 CONTINUE
C
C***** MCDE 4 *****
C***** DRAG DUE TO HOLES IN FUSELAGE *****
C
40 IF (IDAM(4).NE.1) GO TO 50
C
C0 44 I = 1, NPODS
DXCL = DELTA-X FOR EACH HOLE
IF (DPCD(I,31).EQ.0.0) GO TO 44
IF (DBCD(I,31).EQ.1.0) DXCL = C.0
IF (DPCD(I,31).GT.1.0) DXCL = (DRDD(I,37)-DFCD(I,36))
  / (DBDD(I,31)-1.0)
1 CDD = 0.0
C
C0 42 SUMS CDD FOR ALL HOLES, BASED ON CORRECT X-STA., BODY(I)
N = DPCD(I,31)
C0 42 K = 1, N
C
XCL = DRDD(I,36) + (K-1) * DXCL
X = XCL + PDP(I,1)
HCX = DRDD(I,35) / X
XLCB = DBDD(I,33) / DPCD(I,35)
C
IF (DBDD(I,36).EQ.2) GO TO 41
C
CALCULATIONS FOR MISSING PANELS
IF (HCX.GT.0.1) HCX = 0.1
IF (XLCB.GT.10.0) XLCB = 10.0
DCDD = DLNT(XLCB, HCX, XLCB3, HCX3, CDD3, 6, 5, 6, 2, 2)
C
CALCULATIONS FOR CAVED-IN PANELS
41 IF (HCX.GT.0.015) HCX = 0.015
IF (XLCB.GT.120.0) XLCB = 120.0
CCDD = DLNT(XLCB, HCX, XLCB4, HCX4, CDD4, 12, 7, 12, 2, 2)
C
43 CDD = CDD + DCDD
IF (KPRINT(28).EQ.1)
  WRITE(6,5041) J, I, K, X, DCDD, CDD
42 CONTINUE
CALL LMP (XMACH, CPMK, XMACH3, CPMK3, 17, 4)
DRAG = (CDD + CPMK * CCDD(I,34) * DRDD(I,35)) / SREF
MAGNIFY DRAG OF HOLES OVER WING PY 2.31
CRAG = DRAG * (1. + 1.31 * DBCD(I,32) / CBCD(I,31))
C
DAMCD(I,K) = DAMCD(BODY NO, DAMAGE MCDE)
DAMCD(I,4) = DRAG
IF (KPRINT(28).EQ.1)
  WRITE(6,5042) CPMK, DAMCD(I,4)
44 CONTINUE
C
C***** MDDE 5 *****
C***** DRAG DUE TO WAVES ON BODIES *****
C
50 IF (IDAM(5).NE.1) GO TO 60
C
C0 52 I = 1, NPODS
IF (DPCD(I,41).EQ.0.0) GO TO 52
IF (XMACH.GE.1.0) GO TO 53
C
SUBSONIC
SUMCD = C.0
IF (DRDD(I,41).EQ.0.0) GO TO 511

```



```

C   CALCULATE DRAG FOR FLATTING NOSE OF SPHFF
DRAG2 = DLNT(DPOC(I,62),XPACH,XLED,VPACH,DFAG4,6,6,2,2)
CD2   = DRAG2 * DPOC(I,61) / SREF
C
DAMCD(I,7) = CD1 + CD2
IF (KPRINT(2P).EQ.1)
WRITE(6,5071) J, I, RMAX, PNCSE, DRAG1, DRAG2,
1 CD1, CD2, DAMCD(I,7)
71 CONTINUE
C***** MODE 11 *****
C***** DRAG DUE TO ROUGHNESS ON SURFACES *****
C
110 IF (IDAM(11).NE.1) GO TO 120
111 DO 113 I = 1, NSUR
C   CALCULATE DRAG OF UNDAMAGED WING
C   CALL CFK3 (LENGTH, RN/ET, XCL1, XCL2, K1, K2, K3, MACH, CF)
C   CALL CFK3 (SUR(I,20),PNCFT,G,0,DSUR(I,1),0,C,XPACH,CF)
CDI   = CF + SUR(I,7) / SREF
C   CALCULATE DRAG OF DAMAGED SURFACE
C   CALL CFK3 (SUR(I,20),PNCFT,DSUR(I,3),DSUR(I,4),
1 DSUR(I,1),DSUR(I,2),DSUR(I,1),XPACH,CF)
CDF   = CF + SUR(I,7) / SREF
C
DCD   = (CDF - CDI) + DSUR(I,5) / SUR(I,7)
IF (DSUR(I,5).LE.1.) DCF = (CDF - CDI) + DSUR(I,5)
C
MAGNIFY DRAG IF DAMAGE IS ON UPPER SURFACE
IF (DSUR(I,6).EQ.C.C) GO TO 112
TDC   = SUR(I,3)
XM    = 1.+3.*TDC
DCD   = DCD * XM
112 DAMCD(I,11) = DCD
IF (KPRINT(2P).EQ.1)
WRITE(6,5111) J, I, CF, CD1, CDF, DCD, XM, DAMCD(I,11)
113 CONTINUE
C***** MODE 12 *****
C***** DRAG DUE TO FWD-FACING STEPS ON SURFACES *****
C
120 IF (IDAM(12).NE.1) GO TO 130
CALL LNTP (XPACH, CPF, YPACH3, CPF3, 12, 4)
DO 125 I = 1, NSUR
SUMCDL = 0.0
SURCDH = 0.0
CO      = 4.*C / SUR(I,19) * (1. - SUR(I,17)) / (1. + SUR(I,17))
C   CALCULATE DRAG FOR STEPS ON LOW-VELOCITY SURFACES
IF (DSUR(I,11).EQ.0.0) GO TO 122
IF (DSUR(I,11).EQ.1.0) DXCC = (DSUR(I,16) - DSUR(I,15)) / 2.
IF (DSUR(I,11).EQ.1.0) DXCC = (DSUR(I,16) - DSUR(I,15))
/ (DSUR(I,11) - 1.0)
N      = DSUR(I,11)
DO 121 K = 1, N
XCC   = DSUR(I,15) + DXCC * (K-1)
IF (N.EQ.1) XCC = DSUR(I,15) + DXCC
YI    = XCC * SUR(I,20)
HEIGHT = DSUR(I,14)
CALL EFFO(PNCFT, XI, XPACH, HEIGHT, CEFF)
TSWP  = TAN(SUR(I,4))
SWPXCC = ATAN(TSWP - CO + XCC)
DRAG  = CPF * (COS(SWPXCC))**2 * CEFF * DSUR(I,13) * DSUR(I,14) / SREF
SUMCDL = SUMCDL + DRAG
IF (KPRINT(2P).EQ.1)
WRITE(6,5121) J, I, K, XI, CEFF, SWPXCC, DRAG, SUMCDL
121 CONTINUE
C
C   CALCULATE DRAG FOR STEPS ON HIGH-VELOCITY SURFACES
122 IF (DSUR(I,12).EQ.0.0) GO TO 124
IF (DSUR(I,12).EQ.1.0) DXCC = (DSUR(I,16) - DSUR(I,15)) / 2.
IF (DSUR(I,12).EQ.1.0) DXCC = (DSUR(I,16) - DSUR(I,15))
/ (DSUR(I,12) - 1.0)
N      = DSUR(I,12)
DO 123 K = 1, N
XCC   = DSUR(I,15) + DXCC * (K-1)
IF (N.EQ.1) XCC = DSUR(I,15) + DXCC
YI    = XCC * SUR(I,20)
HEIGHT = DSUR(I,14)
CALL EFFO(PNCFT, XI, XPACH, HEIGHT, CEFF)

```

```

TSWP = TAN(SUR(I,4))
SWPXCC = ATAN(TSWP - CC * XCC)
CLO = SUP(I,2)
XM = DLNT(YCC, CLF, XCC3, CLD3, XM3, 6, 4, 6, 2, 2)
DRAG = CPF*(COS(SWPXCC))**3 * QEFF * XM *
1 DSUR(I,13) * CSUR(I,14) / SREF
SUMCDH = SUMCDH + DRAG
IF (KPRINT(28).EQ.1)
1 WRITE(6,5122) J,I,K,XI,QEFF,SWPXCC,XM,DRAG,SUMCDH
123 CONTINUE
124 DAMCD(I,12) = SUMCDL + SUMCDH
125 CONTINUE
C
C***** MODE 13 *****
C***** DRAG DUE TO AFT-FACING STEPS ON SURFACES *****
C
130 IF (IDAM(13).NE.1) GO TO 140
CALL LNTP (XMACH, CPA, ZMACH3, CPA3, 13, 4)
CC 135 I = 1, NSUR
SUMCDL = 0.0
SUMCDH = 0.0
CC = 4.0/SUR(I,19) * (1. - SLP(I,17)) / (1. + SLP(I,17))
C
C CALCULATE DRAG DUE TO STEPS ON LOW-VELOCITY SURFACES
IF (DSUR(I,21).EQ.0.0) GO TO 132
IF (DSUR(I,21).EQ.1.0) DXCC = (DSUR(I,26) - DSUR(I,25)) / 2.
IF (DSUR(I,21).GT.1.0) DXCC = (DSUR(I,26) - DSUR(I,25))
1 / (DSUR(I,21) - 1.0)
N = CSUR(I,21)
CC 131 K = 1, N
XCC = CSUR(I,25) + DXCC * (K-1)
IF (N.EQ.1) XCC = DSUR(I,25) + DXCC
XI = XCC * SLR(I,20)
HEIGHT = DSUR(I,24)
CALL EFFC(FNCFI, XI, XMACH, HEIGHT, CFFF)
TSWP = TAN(SUR(I,4))
SWPXCC = ATAN(TSWP - CC * XCC)
DRAG = -CPA*(COS(SWPXCC))**2 * QEFF*CSUR(I,23)*CSLR(I,24)/SREF
SUMCDL = SUMCDL + DRAG
IF (KPRINT(28).EQ.1)
1 WRITE(6,5131) J,I,K,XI,QEFF,SWPXCC,DRAG,SUMCDL
131 CONTINUE
C
C CALCULATE DRAG FOR STEPS ON HIGH-VELOCITY SURFACES
132 IF (DSUR(I,22).EQ.0.0) GO TO 134
IF (DSUR(I,22).EQ.1.0) DXCC = (DSUR(I,26) - DSUR(I,25)) / 2.
IF (DSUR(I,22).GT.1.0) DXCC = (DSUR(I,26) - DSUR(I,25))
1 / (DSUR(I,22) - 1.0)
N = CSUR(I,22)
CC 133 K = 1, N
XCC = CSUR(I,25) + DXCC * (K-1)
IF (N.EQ.1) XCC = DSUR(I,25) + DXCC
XI = XCC * SLR(I,20)
HEIGHT = CSUR(I,24)
CALL EFFC(FNCFI, XI, XMACH, HEIGHT, CFFF)
TSWP = TAN(SUR(I,4))
SWPXCC = ATAN(TSWP - CC * XCC)
CLO = SUP(I,2)
YM = DLNT(XCC, CLF, YCC3, CLD3, YM3, 6, 4, 6, 2, 2)
DRAG = -CPA * (COS(SWPXCC))**2 * QEFF * XM *
1 DSUR(I,23) * CSUR(I,14) / SREF
SUMCDH = SUMCDH + DRAG
IF (KPRINT(28).EQ.1)
1 WRITE(6,5132) J,I,K,XI,QEFF,SWPXCC,YM,DRAG,SUMCDH
133 CONTINUE
134 DAMCD(I,13) = SUMCDL + SUMCDH
135 CONTINUE
C
C***** MODE 14 *****
C***** DRAG DUE TO HOLES IN SURFACES *****
C
140 IF (IDAM(14).NE.1) GO TO 150
CC 149 I = 1, NSUP
C
C LOW VELOCITY SURFACES
CDL = 0.0
IF (DSUR(I,31).EQ.0.0) GO TO 142

```



```

IF (DSUR(I,31).EQ.1.0) DXCC = (DSUR(I,37) - DSUR(I,36)) / 2.
IF (DSUR(I,31).CT.1.0) DXCC = (CSUR(I,37) - DSUR(I,36))
/ (DSUR(I,31) - 1.0)
C
1 DO 141 SUMS CDC FOR ALL HOLES ON LOW VELOCITY SURFACE
N = CSUR(I,31)
C
DC 141 K = 1, N
C
XCC = DSUR(I,26) + (K-1) * DXCC
X = XCC * SLR(I,20)
HCX = CSUR(I,35) / X
XLCH = DSUR(I,33) / CSUR(I,35)
C
IF (DSUR(I,38).EQ.2.0) GO TO 145
C
CALCULATE CDC FOR MISSING PANELS
IF (HCX.GT.0.1) HCX = 0.1
IF (XLCH.GT.10.0) XLCH = 10.0
CDCG = DLNT(XLCH, HCX, XLCH2, HCX2, CDC3, 6, 5, 6, 2, 2)
GO TO 146
C
CALCULATE CDC FOR CAVED-IN PANELS
145 IF (HCX.GT.0.015) HCX = 0.015
IF (XLCH.GT.120.0) XLCH = 120.0
CDCG = DLNT(XLCH, HCX, XLCH4, HCX4, CDC4, 12, 7, 12, 2, 2)
C
146 CDCL = CDCL + CDCG
C
IF (KPRINT(28).EQ.1)
WRITE(6,5141) J, I, K, X, HCX, XLCH, CDCG, CDCL
141 CONTINUE
C
HIGH VELOCITY SURFACES
142 CDCH = 0.0
IF (DSUR(I,32).EQ.0.0) GO TO 144
IF (DSUR(I,32).CT.1.0) DXCC = (DSUR(I,37) - DSUR(I,36)) / 2.
IF (DSUR(I,32).CT.1.0) DXCC = (CSUR(I,37) - DSUR(I,36))
/ (DSUR(I,32) - 1.0)
C
1 DO 143 SUMS CDC FOR ALL HOLES ON HIGH VELOCITY SURFACE
N = CSUR(I,32)
C
DO 143 K = 1, N
XCC = DSUR(I,36) + (K-1) * DXCC
X = XCC * SLR(I,20)
HCX = CSUR(I,35) / X
XLCH = DSUR(I,33) / DSUR(I,35)
C
IF (DSUR(I,38).EQ.2.0) GO TO 147
C
CALCULATE CDCG FOR MISSING PANELS
IF (HCX.GT.0.1) HCX = 0.1
IF (XLCH.GT.10.0) XLCH = 10.0
CDCG = DLNT(XLCH, HCX, XLCH2, HCX2, CDC3, 6, 5, 6, 2, 2)
GO TO 148
C
CALCULATE CDCG FOR CAVED-IN PANELS
147 IF (HCX.GT.0.015) HCX = 0.015
IF (XLCH.GT.120.0) XLCH = 120.0
CDCG = DLNT(XLCH, HCX, XLCH4, HCX4, CDC4, 12, 7, 12, 2, 2)
C
148 CLD = SUP(I,2)
YM = DLNT(XCC, CLD, XCC2, CLC3, YM2, 6, 4, 6, 2, 2)
CDCH = CDCH + CDCG * YM
IF (KPRINT(28).EQ.1)
WRITE(6,5142) J, I, K, X, HCX, XLCH, CDCG, CDCH
143 CONTINUE
C
CALCULATE DRAG FROM CDC FOR LOW AND HIGH VELOCITY SURFACES
C
144 CONTINUE
CALL LNTP (YMACH, COMK, YMACH2, COMK2, 17, 4)
C
SLMCDG = CDCL + CDCH
DRAG = SLMCDG * COMK + DSUR(I,34) * DSUR(I,35) / SREF
APC = CSUR(I,31) + CSUR(I,32)
IF (APC.GT.0.1.AND.KPRINT(28).EQ.1) WRITE(6,5143) COMK, SLMCDG, DRAG
SAMCD(I,14) = DRAG
149 CONTINUE
C
C*****
MODE 15
*****

```



```

157 DAMCD(I,15) = SUMCDL + SUMCDH
159 CONTINUE
C
C*****          MODE 16          *****
C*****          DRAG DUE TO PROTUBERANCES ON SURFACES *****
C
160 IF (IDAM(16).NE.1) GO TO 170
C
CC 163 I = 1, NSLR
SLMCD = 0.0
IF (DSUR(I,51).EQ.0.0) GO TO 162
IF (DSUR(I,51).EQ.1.0) DXCC = 0.0
IF (DSUR(I,51).GT.1.0) DXCC = (DSUR(I,51)-DSUR(I,54))
1 / (DSUR(I,51) - 1.0)
DOG = DSUR(I,53)
N = DSUR(I,51)
CC 161 K = 1, N
XCC = DSUR(I,54) + DXCC * (K-1)
HEIGHT = [SUR(I,52)]
XI = XCC * SLR(I,20)
HEIGHT = 3.0 / 12.0
CALL EFFC(FNCFI, XJ, XMACH, HEIGHT, CEFF)
DRAG = CEFF * DOG / SREF
SLMCD = SLMCD + DRAG
IF (KPOINT(28).EQ.1)
1 WRITE(6,5161) J, I, K, XI, CEFF, DRAG, SLMCD
161 CONTINUE
162 DAMCD(I,16) = SUMCD
163 CONTINUE
C
C*****          MODE 17          *****
C*****          DRAG DUE TO MISSING WINGS PARTS *****
C
CALCULATE MINIMUM DRAG DUE TO MISSING LEADING EDGE
C
170 IF (IDAM(17).NE.1) GO TO 210
DRAGLE = 0.0
DRAGTE = 0.0
DRGTIP = 0.0
C
IF (DWING(5).LE.1.0) SHARPF = 0.010
IF (DWING(5).GT.1.0) SHARPF = 0.015
DELB = DWING(3) * PC2
DRAGLE = SHARPF * DWING(4) * DELB * CRAP / SREF
IF (DWING(6).EQ.3.0) DRAGLE = 2. * DRAGLE
IF (KPRINT(28).EQ.1)
1 WRITE(6,5171) XMACH, SHARPF, DELB, CRAP, DRAGLE
C
CALCULATE MINIMUM DRAG DUE TO MISSING TRAILING EDGE
C
171 CONTINUE
C
CHARL = LENGTH OF CHORD WITH MISSING T.E. PANEL
172 CHARL = (1.-DWING(7)) * CBAR
IF (IDAM(11).EQ.1) RCHK = [SUR(1,2)]
IF (DSUR(1,2).EQ.0.0) RCHK = RGLCHK
IF (IDAM(11).NE.1) RCHK = FCUGHK
C
CALL CFK3 (LENGTH, FNCFI, XCL)1, XCL)2, K1, K2, K3, MACH, CF)
CALL CFK3 (CHARL, FNCFI, 0., 0., RCHK, 0., 0., XMACH, CF)
CALL CFK3 (CPAR, FNCFI, 0., 0., RCHK, 0., 0., XMACH, CF)
DELB = DWING(9) * RQ2
C
DRAGTE = (0.2/CF**0.33)*DWING(7)**1.33 + TDCW**1.23
1 *DELB *CRAR/SREF + (CF1*CHARL - CF *CHAR)*CFPL/SREF
IF (DWING(10).EQ.3.0) DRAGTE = 2. * DRAGTE
C
DRAGTE = BASE DRAG + CHANGE IN FRICTION DRAG
C
IF (KPRINT(28).EQ.1)
1 WRITE(6,5172) FNCFI, CHARL, RCHK, DELB, CF, CF1, DRAGTE
C
CALCULATE MINIMUM DRAG DUE TO MISSING TIP
C
CHARACTERISTIC LENGTH OF MISSING WING TIP
CFRK = AVERAGE CHORD AT POINT WHERE BREAK OCCURS

```

```

DELETA = 0.5 * (DWING(11)+DWING(12))
CPRK = CTX + (CF-CTX) * DELETA
CHGRD = 0.6667 * (CPRK + CTX - CBRK + CTX / (CPRK+CTX))
AWET = (CBRK + CTX) * (DWJNC(11)+DWJNC(12)) * PC2
C
CALL CFK3 (CHGRD,FACFT,0,0,RCPRK,C,C,M/CF,CF)
DRGTIP = - CF * AWET / SREF
C
IF (KPRINT(28).EQ.1)
1WRITE(6,5173) (CPRK, CHGRD, AWET, CF, DRGTIP
C
DAMCD(1,17) = DRACLE + DRACTE + DRGTIP
IF (KPRINT(28).EQ.1)
1WRITE(6,5174) DRACLE, DRAGTE, DRGTIP, DAMCD(1,17)
C
C
C
C
C
MINIMUM DRAG DUE TO MISSING SURFACE PARTS
C
C
C
C
SUM COMPONENTS OF MINIMUM DRAG FOR EACH SURFACE
DO 175 I = 1, NSUR
DRGSLR(I) = 0.0
DO 174 N = 1, 4
DRGSLR(I) = DRGSLR(I) + CDSUP(I,N)
174 CONTINUE
175 CONTINUE
C
C
C
CALCULATE DRAG CHANGE DUE TO MISSING SURFACE PARTS
C
C
IF (NFI.EQ.0) IHT = 1
IF (NFI.GT.0) IHT = 2
M = NPALS + IHT
IF (NFI.GT.0) DAMCD(2,17) = -DRGSLR(2)*(DWING(14)+DWING(15))/2.
DO 176 I = 1, NSUR
DAMCD(I,17) = -DRGSLR(I) * DWING(16+I-M)
176 CONTINUE
180 IF (KPRINT(28).EQ.1)
1WRITE(6,5175) (I,DRGSLR(I), DAMCD(I,17), I = 1,NSUR)
C
210 CONTINUE
C
C
C
INCREASE DRAG IF DAMAGE IS SYMMETRICAL
C
C
DO 230 K = 1, 7
DO 220 I = 1, 7
IF (SYM(I,K).EQ.1.0) DAMCD(I,K) = BDC(I,6) * DAMCD(I,K)
IF (SYM(I,K+10).EQ.1.0) DAMCD(I,K+10) = 2.0 * DAMCD(I,K+10)
220 CONTINUE
230 CONTINUE
C
CALL WRITE(J)
CALL NUCCAM2(J)
CALL NTRIM(J)
CALL WRITE2(J)
C
500 CONTINUE
C
C
C
C
FORMAT STATEMENTS FOR CHECKOUT
5000 FORMAT (1H1, /, 10X, *DUMP FROM SUPPLIERS NUCCAM, MACH =*,
1 F10.3, // )
5011 FORMAT(10X,*MODE 1, J, I, CDI, CDF, CF(DAMAGED), DAMCD(I,J,1)=*
1 ,2I3, 4F10.5)
5021 FORMAT (5X,*MODE 2, J, I, K, XI, QEFF, CPF, SUMO = *,
1 ,3I3, 4F10.3)
5022 FORMAT (20X, * DAMCD(I,J,2) = *, F10.5)
5031 FORMAT (10X,*MODE 3, J, I, K, XI, QEFF, CPA, SUMO = *,
1 ,3I3, 4F10.3)
5032 FORMAT(20X, * DAMCD(I,J,2) = *, F10.5)
5051 FORMAT (10X,*MODE 5, PFTPLDFF, SUPERSONIC, J, I, K, XI, QEFF,*,
1 * DRAG, SLMCD =*, 3I3, 2F8.3, 2F10.5)
5053 FORMAT(10X, *MODE 5, PROTPDEF, SUPERSONIC, J, I, *,
1 * DAMCD(I,J,K) = *, 2I3, F10.5)
5054 FORMAT(20X, * J, I, DAMCD(I,J,5) = *, 2I3, F10.5)
5054 FORMAT(10X,*MODE 5, INDPTED, J, I, K, Y, CFF, FRAC, *,
1 * SUMCF, CPRK =*,3I3,FR.2,FE.4,2F11.6,FE.3)
5055 FORMAT(20X, * J, I, DAMCD(I,J,5) = *, 2I3, F10.5)

```

```

5041 FORMAT(10X, *MODE 4, J, I, K, X, DCDC, CDD = *,
1 3I3, 3F10.3)
5042 FORMAT(20X, *CCM, DAMCD(I,J,4) = *, F10.3, F10.5)
5061 FORMAT(5X, *MODE 6, J, I, K, YI, CEFF, DRAG, SUMCD = *,
1 3I3, 2F10.3, 2F10.5)
5071 FORMAT(10X, *MODE 7, J, I, RMAX, RNDSE, DRAC1, DRAG2, *,
1 *CD1, CD2, DAMCD(I,J,7)=*, 2I3, 4F8.3, 3F8.5)
5111 FORMAT(10X, *MODE 11, J, I, CF(CAPAGER), CL1, CDF, CDD, *,
1 *XM, DAMCD(I,J,11)=*, 2I3, 4F8.5, F6.3, F9.5)
5121 FORMAT(4X, *MODE 12, LOW VELOCITY, J, I, K, XI, CEFF, *,
1 *SWPXC, DRAG, SUMCD=*, 3I3, 3F7.2, 7X, 2F8.5)
5122 FORMAT(4X, *MODE 12, HIGH VEL., J, I, K, XI, CEFF, *,
1 *SWPXC, XM, DRAG, SUMCD=*, 3I3, 4F7.3, 2F8.5)
5131 FORMAT(4X, *MODE 13, LOW VELOCITY, J, I, K, YI, CEFF, *,
1 *SWPXC, DRAG, SUMCD=*, 2I3, 2F7.3, 7X, 2F8.5)
5132 FORMAT(4X, *MODE 13, HIGH VEL., J, I, K, XI, CEFF, *,
1 *SWPXC, XM, DRAG, SUMCD=*, 3I3, 4F7.3, 2F8.5)
5141 FORMAT(10X, *MODE 14, LOW VEL., J, I, K, X, HX, YLCH, DCDC, *,
1 *CDD =*, 3I3, 5F7.3)
5142 FORMAT(10X, *MODE 14, HIGH VEL., J, I, K, X, HX, YLCH, DCDC, *,
1 *CDD =*, 3I3, 5F7.3)
5143 FORMAT(20X, *CCM, SUMCD, DRAG = *, F10.2, F10.3, F10.5)
5151 FORMAT(4X, *MODE 15, PCTRUCED, LOW VEL., J, I, K, CEFF, *,
1 *DRAG, SUMCD=*, 3I3, F8.5, 8X, 2F8.5)
5152 FORMAT(4X, *MODE 15, PCTRUCED, HIGH VEL., J, I, K, CEFF, *,
1 *XM, DRAG, SUMCD=*, 3I3, 4F8.5)
5153 FORMAT(4X, *MODE 15, PCTRUCED, LOW VEL., SUPERSONIC, *,
1 *J, I, SUMCD=*, 2I3, F10.5)
5154 FORMAT(4X, *MODE 15, PCTRUCED, HIGH VEL., SUPERSONIC, *,
1 *J, I, K, XM, DRAG, SUMCD=*, 3I3, 3F10.5)
5155 FORMAT(4X, *MODE 15, INDENTED, LOW VEL., J, I, K, XL, *,
1 *CDD, DRAG, SUMCD=*, 3I3, 2F8.3, 2F8.5)
5156 FORMAT(4X, *MODE 15, INDENTED, HIGH VEL., J, I, K, XM, *,
1 *CDD, DRAG, SUMCD=*, 3I3, 2F10.4, 2F14.8)
5161 FORMAT(10X, *MODE 16, J, I, K, XI, CEFF, DRAG, SUMCD = *,
1 3I3, 2F8.3, 2F10.5)
5171 FORMAT(///, 10X, *MODE 17, *MACH = *, F10.5,
1 /, 10X, *SHAKE, LELP, CRAP, BEAGLE = *, 4F10.5)
5172 FORMAT( /, 10X, *MAGNET, CHARL, SGM, DELP, CF, CFI, *,
1 *DRAG = *, F11.0, 6F10.5)
5173 FORMAT( /, 10X, *CFK, CHCR, ANET, CF, DRGTIP = *, 5F10.4)
5174 FORMAT( /, 10X, *DRAGLE, DRAGTE, DRGTIP, DAMCD(1,J,17)=*,
1 4F11.5)
5175 FORMAT( /, 10X, *DRAG DUE TO MISSING SURFACE PARTS*,
1 /, 14X, *I TOTAL DRAG OF EACH SURFACE DAMCD(1,J,17)*
2 /, ( 12X, I2, 12X, F10.5, 12X, F9.5)
END

```

```

SUBROUTINE WRITE(J)
C*****
C***** WRITE ALL PCDES STATEMENTS *****
C*****
C
COMMON /INPUT/ A(3CF1)
COMMON /FLKTL/ TITLE(6)
COMMON /NCCUT/ DAMCD(7,17), SYM(7,17), CCOMIN,
1 ALP(21), CL(21), CC(21), CM(21), CLT(21), CDT(21),
2 CIP(21), COP(21), CMP(21), CLTP(21), CDTP(21),
3 RCLA(6), DXAC,
4 CFRLI(21), CRT(21), X2(2), X4(2)
DIMENSION CDRCD(10), COSUR(10), CDABF(20),
1 FMSURV(20), ALT(20), CLIC(20), CLPI(20), IDAM(20),
2 FNAME(7), SNAME(7)
EQUIVALENCE (A(1), NPLOYS), (A(2), NNAC), (A(3), NPPLS),
1 (A(4), NMT), (A(5), NVT), (A(491), NSURV),
2 (A(492), NCLAS), (A(493), FMSURV(1)),
3 (A(513), ALT(1)), (A(573), CLIC(1)),
4 (A(593), CLPI(1)), (A(1642), IDAM(1)),
5 (A(30), FNAME(1)), (A(177), SNAME(1))
C
NPGDS = NPODYS + NNAC
NSUR = NPPLS + NMT + NVT
JJJ = ICAM(1) + IDAM(2) + IDAM(2) + IDAM(4) + ICAM(5) +
1 IDAM(6) + IDAM(7) + ICAM(8) + ICAM(5) + ICAM(10)
WRITE(6,1000) (TITLE(I), I = 1,6)
WRITE(6,1001) FMSURV(J)
IF (JJJ.EQ.0) GO TO 40
WRITE(6,1002) (FNAME(I), I = 1,NPGDS)
IF (IDAM(1).EQ.1) WRITE(6,1003) (DAMCD(I,1), I=1,NPGDS)
IF (IDAM(2).EQ.1) WRITE(6,1004) (DAMCD(I,2), I=1,NPGDS)
IF (IDAM(3).EQ.1) WRITE(6,1005) (DAMCD(I,3), I=1,NPGDS)
IF (IDAM(4).EQ.1) WRITE(6,1006) (DAMCD(I,4), I=1,NPGDS)
IF (IDAM(5).EQ.1) WRITE(6,1007) (DAMCD(I,5), I=1,NPGDS)
IF (IDAM(6).EQ.1) WRITE(6,1008) (DAMCD(I,6), I=1,NPGDS)
IF (IDAM(7).EQ.1) WRITE(6,1009) (DAMCD(I,7), I=1,NPGDS)
C
SUM DRAG ON EACH PCDY
DO 10 I = 1, 10
CDBD(I) = 0.0
10 CONTINUE
C
DO 20 I = 1, NPGDS
SUM = 0.0
DO 20 K = 1, 10
SUM = SUM + DABCD(I,K)
20 CONTINUE
CDBD(I) = SUM
30 CONTINUE
40 JJJ = ICAM(11)+ICAM(12)+IDAM(13)+ICAM(14)+IDAM(15)+
1 ICAM(16)+IDAM(17)+ICAM(18)+IDAM(19)+IDAM(20)
IF (JJJ.EQ.0) GO TO 60
WRITE(6,1010) (CDRCD(I), I = 1,NPGDS)
WRITE(6,1011) (SNAME(I), I = 1,NSUR)
IF (IDAM(11).EQ.1) WRITE(6,1012) (DAMCD(I,11), I=1,NSUR)
IF (IDAM(12).EQ.1) WRITE(6,1013) (DAMCD(I,12), I=1,NSUR)
IF (IDAM(13).EQ.1) WRITE(6,1014) (DAMCD(I,13), I=1,NSUR)
IF (IDAM(14).EQ.1) WRITE(6,1015) (DAMCD(I,14), I=1,NSUR)
IF (IDAM(15).EQ.1) WRITE(6,1016) (DAMCD(I,15), I=1,NSUR)
IF (IDAM(16).EQ.1) WRITE(6,1017) (DAMCD(I,16), I=1,NSUR)
IF (IDAM(17).EQ.1.AND.NPPLS.EQ.1) WRITE(6,1018)
1 (DAMCD(I,17), I=1,NSUR)
IF (IDAM(17).EQ.1.AND.NPPLS.EQ.2) WRITE(6,2001) DAMCD(I,17),
1 (DAMCD(I,17), I=3,NSUR)
C
SUM DRAG ON EACH SURFACE
DO 50 I = 1, 10
COSUR(I) = 0.0
50 CONTINUE
C
DO 70 I = 1, NSUR
SUM = 0.0
DO 60 K = 11,17
SUM = SUM + DAMCD(I,K)
60 CONTINUE
COSUR(I) = SUM

```

```

70 CONTINUE
WRITE(6,1019) (CPSUR(I), I = 1, NSLP)
C
C
C
SUM DRAG ACCORDING TO MODE OF DAMAGE
C
C
C
R O D I E S
80 DO 100 K = 1,7
SUM = 0.0
DO 90 I = 1, NPODS
SUM = SUM + DAMCD(I,K)
90 CONTINUE
COMODE(K) = SUM
100 CONTINUE
C
C
C
S U R F A C E S
DO 120 K = 11, 17
SUM = 0.0
DO 110 I = 1, NSUP
SUM = SUM + DAMCD(I,K)
110 CONTINUE
COMODE(K) = SUM
120 CONTINUE
C
C
C
WRITE(6,1020)
CD1 = COMODE(1) + COMODE(11)
CD2 = COMODE(2) + COMODE(12)
CD3 = COMODE(3) + COMODE(13)
CD4 = COMODE(4) + COMODE(14)
CD5 = COMODE(5) + COMODE(15)
CD6 = COMODE(6) + COMODE(16)
CD7 = COMODE(7)
CDF = COMODE(17)
WRITE(6,1021) COMODE(1), COMODE(11), CD1
WRITE(6,1022) COMODE(2), COMODE(12), CD2
WRITE(6,1023) COMODE(3), COMODE(13), CD3
WRITE(6,1024) COMODE(4), COMODE(14), CD4
WRITE(6,1025) COMODE(5), COMODE(15), CD5
WRITE(6,1026) COMODE(6), COMODE(16), CD6
WRITE(6,1027) COMODE(7), COMODE(17), CD7
WRITE(6,1028) COMODE(17), CDF
C
C
C
SUM TOTAL DAMAGE DRAG ON PODIES AND WINGS
AAA = 0.0
BBB = 0.0
DO 130 I = 1,7
AAA = AAA + COMODE(I)
BBB = BBB + COMODE(I+10)
130 CONTINUE
CCC = AAA + BBB
WRITE(6,1029) AAA, BBB, CCC
CCOIN = CCC
1000 FORMAT(2H1, //, 21X, 6A10)
1001 FORMAT( //, 21X, *SUMMARY OF DAMAGE EFFECTS ON MINIMUM DRAG*,
1 //, 32X, *N A C H = *, F5.2, //)
1002 FORMAT ( //, 5X, *BODY DAMAGE MODE *, 4X, 7A10)
1003 FORMAT ( //, 5X, *ROUGHNESS *, 2X, 7F10.5)
1004 FORMAT ( //, 5X, *FWD-FACING STEPS *, 2X, 7F10.5)
1005 FORMAT ( //, 5X, *AFT-FACING STEPS *, 2X, 7F10.5)
1006 FORMAT ( //, 5X, *HOLES *, 2X, 7F10.5)
1007 FORMAT ( //, 5X, *WAVINESS *, 2X, 7F10.5)
1008 FORMAT ( //, 5X, *FRONTUPFRANCES *, 2X, 7F10.5)
1009 FORMAT ( //, 5X, *PLUNTINGS *, 2X, 7F10.5)
1010 FORMAT ( //, 5X, *TOT. DRAG, FA, BODY = *, 2X, 7F10.5)
1011 FORMAT ( //, 5X, *SURFACE DAMAGE MODE *, 4X, 7A10)
1012 FORMAT ( //, 5X, *ROUGHNESS *, 2X, 7F10.5)
1013 FORMAT ( //, 5X, *FWD-FACING STEPS *, 2X, 7F10.5)
1014 FORMAT ( //, 5X, *AFT-FACING STEPS *, 2X, 7F10.5)
1015 FORMAT ( //, 5X, *HOLES *, 2X, 7F10.5)
1016 FORMAT ( //, 5X, *WAVINESS *, 2X, 7F10.5)
1017 FORMAT ( //, 5X, *FRONTUPFRANCES *, 2X, 7F10.5)
1018 FORMAT ( //, 5X, *MISSING PARTS *, 2X, 7F10.5)
2001 FORMAT ( //, 5X, *MISSING PARTS *, 2X, 7F10.5, *-----*,
1 //, 5)
1019 FORMAT ( //, 5X, *TOT. DRAG, FA, SURFACE = *, 2X, 7F10.5)
1020 FORMAT ( //, 17X, *NET DRAG FROM EACH MODE OF DAMAGE*,
1 //, 29X, *PODIES SURFACES *PODIES + SURFACES*)

```

```

1021 FORMAT ( 5X, *ROUGHNESS *, 2F10.5, F16.5)
1022 FORMAT ( 5X, *FWD-FACING STEPS *, 2F10.5, F16.5)
1023 FORMAT ( 5X, *AFT-FACING STEPS *, 2F10.5, F16.5)
1024 FORMAT ( 5X, *HOLES *, 2F10.5, F16.5)
1025 FORMAT ( 5X, *WAVINESS *, 2F10.5, F16.5)
1026 FORMAT ( 5X, *PROTUPERANCES *, 2F10.5, F16.5)
1027 FORMAT ( 5X, *PLUNTHNESS*, 12X, F10.5, 6X, *N/A *, F15.5)
1028 FORMAT ( 5X, *MISSING PARTS *, 6X, *N/A *, F10.5, F16.5)
1029 FORMAT (/, 5X, *TOTAL DAMAGE DRAG *, 2F10.5, F16.5)
RETURN
END

```



```

C      SUBROUTINE CFK3(XL,RNOFT,XCL1,XCL2,RK1,RK2,RK3,ZMACH,CFR)
C      CALCULATES SKIN FRICTION COEFF. FOR A PLATE WITH LP TO
C      THREE ZONES OF ROUGHNESS
C
1     CF(X,Y) = A*MAX1(T**2+C.43C/(ALOG10(RNL*X*T**1.67*F)))**2.56,
C           T/(1.89+1.62*ALOG10(X/Y))**2.5
C
C     SCF(X) = T**2+C.43C/(ALOG10(PNL*X*T**1.67*F))**2.56
C
C     SCFP(X) = T**2+C.43C/(ALOG10(RNL*X*T**1.67*F))**2.56
C           -T**2+C.478C7/(ALOG10(RNL*X*T**1.67*F))**3.56
C
1     RCFP(X,Y) = T/(1.89+1.62*ALOG10(X/Y))**2.5
C           -1.758C *T/(1.89+1.62*ALOG10(X/Y))**3.5
C
C     T = 1./(1.+C.17E*ZMACH**2)
C     F = 1.+C.03916*ZMACH**2*T
C
C     RNL = RNOFT
C     RK3 = RK3
C
C     CFR = SCF(XL)
C     IF(RK1.GT.C.0) CFR = CF(XL,RK1)
C     IF(XCL1.EC.C.C.AND.XCL2.EC.C.0) RETURN
C
C     IF(XCL1.GT.C.0) GO TO 100
C     INCREASED ROUGHNESS STARTS AT LEADING EDGE
C     X1 = XL + XCL2
C     CFR = SCF(X1)
C     IF(RK2.GT.C.0) CFR = CF(X1,RK2)
C     IF(XCL2.GE.1.0) RETURN
C
C     J = 2
C     CFY1 = CFR
C     DX = X1
C     RK = RK3
C     GO TO 10
C
C 100 CONTINUE
C
C     X1 = XL + XCL1
C     RKA = RK1
C     RK = RK2
C     J = 1
C
C 5     CFX1 = SCF(X1)
C     IF(RKA.GT.C.0) CFX1 = CF(X1,RKA)
C     DX = X1
C
C 10     CFS = SCF(DX)
C     CFY = CFS
C     IF(RK.GT.C.0) CFX = CF(DX,RK)
C     DPS = SCFP(DX)
C     IF(CFY.NE.CFS) DPS = FCFP(DX,RK)
C     IF(DPS.EC.C.0) GO TO 200
C
C     DX1 = DX - (DX *CFY - Y1 #CF(X1))/DPS
C     IF(DX1.LE.C.0) DX1 = 0.5 *DX
C     E1 = AFS(DX1 -CX)
C     E2 = G.0001 *X1
C     IF(F1.LE.E2) GO TO 20
C     DX = DX1
C     GO TO 10
C
C 20     IF(J.EC.2) GO TO 30
C     Y1 = DY +(XCL2 -XCL1)*XL
C     IF(XCL2.GE.1.0) GO TO 25
C     RKA = RK2
C     RK = RK3
C     J = 2
C     GO TO 5
C
C     INCREASED ROUGHNESS ENDS AT TRAILING EDGE
C
C 25     XP = X1
C     RK3 = RK2

```

```

      GO TO 35
C
C
30  XP   = DY1 + XL *(1.-XPL2)
35  CFYP = SCF(XP)
    IF (RK3.GT.C.C) CFYP = CF(XP,RK3)
    CFR  = (XP/XL) * CFYP
    GO TO 300
-
200 WRITE (6,1000) DY,CFY,RK
    CFR = 0.C
300 CFXL = SCF(XL)
C
1000 FORMAT(10X*SUBROUTINE CFK3 WILL NOT CONVERGE*/1X,3F15.7 )
C
    RETURN
    END

```

```

C      SLROUTINE EFFQ(RNOFT, XI, XMACH, HEIGHT, CFFF)
C      CALCULATE EFFECTIVE DYNAMIC PRESSURE RATIO
C      CFFF = 1.0
C      IF (XI.LE.0.0) RETURN
C      BLT = 0.37 * XI / (RNOFT * XI)**0.2
10  IF (HEIGHT.LE.PLT) CFFF = .7778 * (HEIGHT / PLT)**0.286
C      IF (HEIGHT.GT.PLT) CFFF = 1.0 - 0.222 * (PLT. / HEIGHT)
C      RETURN
C      END

```

```

SUBROUTINE NUCCAM2(J)
C*****
C***** EFFECT OF MISSING PARTS ON LIFT CLAVE, *****
C***** PCLAR SHAPE, AND MOMENTS *****
C*****
COMMON /INPUT/ A(3081)
COMMON /OUTPUT/ P(222)
COMMON /CALC/ C(50)
COMMON /BLKERT/ KPRINT(50)
COMMON /PLKCLA/ SPLAMP, TDCW, TRP, ARP, RMSTR, CLD, DM1,
1 DCRP, EPSL, SWPMC, SREFP
COMMON /NUCOUT/ GAMC(7,17), SYX(7,17), GCPMIN,
2 ALP(21), CL(21), CP(21), CM(21), CLT(21), CDT(21),
3 CLP(21), CGP(21), CMP(21), CLTP(21), CDTP(21),
4 RCLA(6), DXAC,
C CROLL(21), CLRT(21), X2(2), X4(2)
C
DIMENSION DWING(20), FMSUPV(20), IDAM(20)
1 ,DCLA2(2), DXAC2(2), DK2(2), DCLAT(2)
2 ,SUR(7,20), DSLR(7,100)
C
EQUIVALENCE (A(3(62), DWING(1)) , ( P(54) , XK ) ,
1 (P(61) , R ) , ( P(50) , CLA ) ,
2 (C(6) , SWPCC ) , ( C(29) , APL ) ,
3 (A(401) , FMSUPV ) , (A(492) , NCLAS ) ,
4 (C(4) , SPLAN ) , (C(1) , RM2 ) ,
5 (C(2) , DCRP ) , (C(2) , CR ) ,
6 (P(56) , CMG ) , (P(60) , ALG ) ,
7 (P(96) , CLAW ) , (C(16) , CTX ) ,
8 (C(25) , SWPLF ) , (C(15) , CRX ) ,
9 (A(493) , FMSUPV(1)) , (A(12) , CMAC ) ,
10 (A(1642) , IDAM(1)) , (A(11) , SREF ) ,
11 (C(23) , XH ) , (P(66) , EP ) ,
12 (P(98) , CLAT ) , (A(5) , NPALS ) ,
13 (A(184) , SIF(1,1)) , (A(23(2) , DSLR(1,1))
14 (P(101) , HSTAP)
C
COMMON /FLKDAT3/ YVACH3(17), CMK3(17),
1 XLP3(6), HFX3(5), CFC3(6,5),
2 XCP3(6), CLF3(4), YF3(6,4),
3 YVACH3(13), CPF3(13),
4 ZVACH3(13), CP43(13),
5 XLP4(12), HFX4(7), CFC4(12,7),
6 RTTP3(8), UMACH(10), DRAG3(8,10),
7 XLP(6), VMACH(6), DPAG4(6,6),
8 DETA1(6), AR1(5), CL1(6,5),
9 FTA2(6), SWPCC2(5), CL2(6,5),
10 FTA3(6), CLA3(6),
11 DETA4(6), AR4(5), CL4(6,5),
12 DETA5(6), SWPCC5(5), CL5(6,5),
13 FTA6(6), CLA6(6),
14 DCRP(6), DCCC(6), CLA(14(6,6)
C
COMMON /BLKDAT4/ DETA11(6), AR11(5), YAC1(6,5),
1 DETA12(6), SWPCC12(5), YAC2(6,5),
2 DETA13(6), YAC3(6),
3 DETA14(6), AR14(5), YAC4(6,5),
4 DETA15(6), SWPCC15(5), YAC5(6,5),
5 DETA16(6), YAC6(6),
6 DCCC7(5), YAC7(5),
7 DCCC8(5), YAC8(5),
8 FL01(6), CL11(6),
9 FL02(6), FFF2(6),
10 ZFR(10), XCLA(10)
C
SWPCC = SWPCC * 57.2056
101 DETA11 = 1.5
DCLA(1) = C.G
CONTINUE
DCLA = G.C
DK = 0.0
DXAC = C.0
DCLAT(1) = 0.0
DCLAT(2) = C.0
DCLA1 = C.0
DCLA2 = 0.0

```

```

DCLA3 = C.0
DCLA4 = C.0
DCLA5 = C.0
DCLA6 = C.0
DXAC1 = C.0
DXAC2 = C.0
DXAC3 = C.0
DXAC4 = C.0
DXAC5 = C.0
DXAC6 = C.0
DXAC7 = C.0
DXAC8 = C.0
X3(1) = C.0
X3(2) = C.0
X4(1) = C.0
X4(2) = C.0
DCLA2(1) = C.0
DCLA2(2) = C.0
DXAC2(1) = C.0
DXAC2(2) = C.0
DK1 = C.0
DK2(1) = C.0
DK2(2) = C.0

```

```

C
IF (KPRINT(29).EQ.1)
WRITE(6,1090) FMSURV(J), ( P(I), I=1,223)

```

```

C
XMACH = FMSURV(J)
AAA = ARS( B(144) - XMACH)
IF (AAA.GT.0.CG1) WRITE(6,1001)
1001 FORMAT (//,20X,' INCORRECT MACH NUMBER WAS READ FROM *,
1 *TAPE UNIT NUMBER 10 * )
IF (IDAM(17).FC.0) GO TO 146

```

```

C ***** DEFINE INCREMENT IN LIFT-CURVE SLOPE *****
C ***** FOR DAMAGED L.E. AND T.E. , DCLA1 *****
C

```

```

C
XXY = C.0
YYY = C.0
IF (DWING(3).EQ.0.C) GO TO 110
DCLA1 = DLNT(DWING(3), AF , DELTA1, AF1, CLA1, 6,5,6,2,2)
DCLA2 = DLNT(DWING(3), SWPCC, DELTA2, SWPCC2, CLA2, 6,5,6,2,2)
CALL LNTP(DWING(2), DCLA3, DELTA3, CLA3, 6, 4)
XXY = DCLA1 + DCLA2 + DCLA3
IF (DWING(6).LT.3.) XXY = 0.5 * XXY
110 IF (DWING(9).EQ.0.C) GO TO 120
DCLA4 = DLNT(DWING(9), AF , DELTA4, AR4, CLA4, 6,5,6,2,2)
DCLA5 = DLNT(DWING(9), SWPCC, DELTA5, SWPCC5, CLA5, 6,5,6,2,2)
CALL LNTP(DWING(8), DCLA6, DELTA6, CLA6, 6, 4)
YYY = DCLA4 + DCLA5 + DCLA6
IF (DWING(10).LT.3.) YYY = 0.5 * YYY

```

```

C
120 DCLA = DELTA-CLA-TOTAL
DCLAX = XXY + YYY

```

```

C
CLARLE IS CL-ALPHA RATIO TO ACCOUNT FOR CHORD OF OUTLET
CLARLE = DLNT(DWING(7), DWING(1), DPCB, DCCC, CLARLE, 6,5,6,2,2)
CLARTE = DLNT(DWING(7), DWING(7), DPCB, DCCC, CLARTE, 6,5,6,2,2)
IF (DWING(6).LT.3.) CLARLE = CLARLE + 0.5 * (1.0-CLARLE)
IF (DWING(10).LT.3.) CLARTE = CLARTE + 0.5 * (1.0-CLARTE)

```

```

C
CLAP = CLA-PRIME, PRIME DENOTES DAMAGED CONFIGURATION
CLAP = (CLA + DCLAX) * CLARLE * CLARTE
IF (KPRINT(29).EQ.1)
WRITE(6,1010) FMSURV(J), DCLA1, DCLA2, DCLA3, DCLA4, DCLA5,
1 DCLA6, CLARLE, CLARTE, CLAP

```

```

C
DEFINE LIFT CURVE INCREMENT REQUIRED FOR CELL TRIM
C

```

```

RCLA(1) = (CLA+XXY) * CLARLE - CLA
RCLA(2) = (CLA+YYY) * CLARTE - CLA
IF (KPRINT(29).EQ.1)
WRITE(6,1020) DCLA1

```

```

C ***** DEFINE AERODYNAMIC CENTER SHIFT *****
C ***** FOR DAMAGED L.E. AND T.E. , DXAC1 *****
C

```

```

    YXX = 0.0
    YYY = 0.0
    IF (DWING(3).EQ.0.0) GO TO 130
    DXAC1 = DLNT(DWING(3), AR , DETA11, AF11 , XAC1, 6,5,6,2,2)
    DXAC2 = DLNT(DWING(3), SLPCC, DETA12, SACC12, XAC2, 6,5,6,2,2)
    CALL LNTP(DWING(2), DXAC2, DETA13, XAC3, 6, 4)
    CALL LNTP(DWING(1), DXAC7, DCCCF, XAC7, 5, 4)
    XXX = DXAC1 + DXAC2 + DXAC3 + DXAC7
    IF (DWING(6).LT.3.) XXX = 0.5 * XXX
    IF (KPRINT(29).EQ.1)
1WRITE(6,1(35)) DXAC1, DXAC2, DXAC3, DXAC7, XXX
130 IF (DWING(9).EQ.0.0) GO TO 140
    DXAC4 = DLNT(DWING(9), AR , DETA14, AF14 , XAC4, 6,5,6,2,2)
    DXAC5 = DLNT(DWING(9), SLPCC, DETA15, SACC15, XAC5, 6,5,6,2,2)
    CALL LNTP(DWING(8), DXAC5, DETA16, XAC6, 6, 4)
    CALL LNTP(DWING(7), DXAC6, DCCCF, XAC6, 5, 4)
    YYY = DXAC4 + DXAC5 + DXAC6 + DXAC6
    IF (DWING(10).LT.3.) YYY = 0.5 * YYY
    IF (KPRINT(29).EQ.1)
1WRITE(6,1(35)) DXAC4, DXAC5, DXAC6, DXAC6, YYY
140 DXAC1 = XXX + YYY
C
    IF (KPRINT(29).EQ.1)
1WRITE(6,1(36)) DXAC1
C
C***** DEFINE INCREMENT IN SPAN EFFICIENCY *****
C***** FOR DAMAGED L.E. AND T.E , DF1 *****
C
C THIS METHOD NOT COMPLETE, MEANWHILE SET
DF1 = 0.0
IF (DWING(1).GT.0.0.OR.DWING(7).GT.C.C) DF1 = -0.05
C
C***** DEFINE INCREMENT IN POLAR SHAPE FACTOR *****
C***** FOR DAMAGED L.E. AND T.E , DK1 *****
C
    YXX = DWING(3)
    IF (DWING(6).LT.3.) XXX = 0.5 * XXX
    RP = R*(1.0-XXX) + 0.2*R*XXX
    EFP = EF + DF1
    XKP = (1.0-RP)/(CLAP*57.29578+SREF/SPLAN) + RP/(3.14159*AP*EFP)
    XK = XKP + (SREF/SPLAN)
    DK1 = XKP - XK
    IF (KPRINT(29).EQ.1)
1WRITE(6,1(40)) R, FP, EF, EFP, XK, XKP, DK1
C
C***** DEFINE INCREMENT IN LIFT-CURVE SLOPE *****
C***** FOR MISSING WING TIPS, DCLA2 *****
C
C DO 141 J=1,2
DCLA2(J) = 0.0
DXAC2(J) = 0.0
DK2(J) = 0.0
DCLAT(J) = 0.0
141 CONTINUE
IF (DWING(11).EQ.0.0.AND.DWING(12).EQ.C.C) GO TO 145
C
C LET I=1 INDICATE LEFT WING
C LET I=2 INDICATE RIGHT WING
DO 143 I = 1, 2
DR = BG2 * 2.0 * DWING(10+I)
C
C DEFINE VARIABLES REQUIRED TO CALL SUBROUTINE AER2
TCW = C(12)
DMSTR = 0.0
CLD = C(13)
D*1 = C.0
FPSL = 0.0
SVPMC = C(7)
SREF = A(11)
RW = 2. * RW2
BWP = RW - DR
CTXP = CR - (BWP/RW) * (CR-CTX)
SPLAN1 = (CR+CTX) * RW / 2.0
SPLAN2 = (CR+CTXP) * RW / 2.0
SPLANP = SPAN + (SPLAN2 - SPAN1)
ARP = BWP*2 / SPANP
TRP = CTRP / CR

```

```

D      = DGB * PW
DCPP  = C / FWP
SREFF = SREF

C
C      CALC. INCREMENT IN POTENTIAL LIFT-CURVE SLOPE
C
CLAWP = 0.0
IF (DCBP.LT.1.0) CALL AFP2(XMACH,CLAWP)
POTENT = CLAWP - CLAV

C
C      CALC. INCREMENT IN VORTEX LIFT-CURVE SLOPE
C
SONICM = 1.0 / (SIN(90.0/57.295 - SWPLE))
VCRK = 1.0 - (XMACH-1.0) / (SONICM-1.0)
IF (XMACH.LE.1.0) VCRK = 1.0
IF (XMACH.GT.SONICM) VCRK = 0.0
CALL LNTP(AR, VCR1, ZAP, XCLA, 10, 4)
CALL LNTP(AR, VCR2, ZAP, XCLA, 10, 4)
VORTEX = (VCR2 - VCR1) * VCRK

C
C      CALC. TOTAL DELTA CLA ON ONE WING
C
DCLA2(I) = 0.5 * (POTENT + VCPTX)
IF (KPFINT(29).EQ.1.AND.I.EQ.1)
1WRITE(6,1045) I
IF (KPFINT(29).EQ.1.AND.I.EQ.2)
1WRITE(6,1046) I
IF (KPFINT(29).EQ.1)
1WRITE(6,1050) RWP,CTXP,SFLANP,ARP,TRP,(CLAWP,CLAW,
? SONICM, VCRK, POTENT, VORTEX, DCLA2(I))

C
C      DEFINE LIFT CURVE INCREMENT REQUIRED FOR ROLL TRIM
C
RCLA(2+I) = DCLA2(I)

C***** DEFINE AERODYNAMIC CENTER SHIFT *****
C***** FOR MISSING WING TIPS, DXAC2 *****

C
C      CALCULATIONS BASED ON EXPOSED WING
CBAR1 = 0.6667 * ((CRX+CTX) - (CRX * CTX) / (CFX + CTX))
CBAR2 = 0.6667 * ((CRX+CTXP) - (CRX * CTPX) / (CFX + CTPX))
TR = CTX / CRX
TRP = CTPX / CRX
ETA1 = .2333 * ((1. + 2. * TR) / (1. + TR))
ETA2 = .2333 * ((1. + 2. * TRP) / (1. + TRP))
PEXP = BW - P
RPEXP = BWP - D
Y1 = ETA1 + PEXP / 2.0
Y2 = ETA2 + RPEXP / 2.0
XLE1 = Y1 + TAN(SWPLE)
XLE2 = Y2 + TAN(SWPLE)
IF (XMACH.LT.1.0) DX = (XLE2+.25*CBAR2)-(XLE1+.25*CBAR1)
IF (XMACH.GE.1.0) DX = (XLE2+.50*CBAR2)-(XLE1+.50*CBAR1)
DXAC2(I) = DX / CMAC
DXAC2(1) = 0.5 * DXAC2(I)
IF (I.EQ.1) X3(1) = Y1 + DCP * RC2
IF (I.EQ.1) X3(2) = Y2 + DCP * PC2
IF (I.EQ.2) X4(1) = Y1 + DCP * RC2
IF (I.EQ.2) X4(2) = Y2 + DCP * PC2

C
C      IF (KPFINT(29).EQ.1)
C      1WRITE(6,1061) CBAR1,CBAR2,ETA1,ETA2,XLE1,XLE2,DXAC2(I)

C***** DEFINE INCREMENT IN SPAN EFFICIENCY *****
C***** FOR MISSING WING TIPS, CE2 *****

C
C      FOR NOW, ASSUME
C      CE2 = 0.0

C***** DEFINE INCREMENT IN POLAR SHAPE FACTOR *****
C***** FOR MISSING WING TIPS, DK2 *****

C
C      EFP = EF + DE2
C      DK2(1) = (1.0/(2.0+3.14159)) * ((CF+CTXP)/(PB*EFP)-(CF+CTX)/(PB*EF))
C      DK2(I) = DK2(1) * (SFEF/SPLAN)
C      DK2(I) = 0. * DK2(I)
C      IF (KPFINT(29).EQ.1)
C      1WRITE(6,1064) EFP, DK2(I)

```

```

143 CONTINUE
C ***** DETERMINE EFFECTS OF MISSING HCF17. TAIL *****
C
145 DO 144 I = 1,2
    DCLAT(I) = CLAT * (1.0 - DWING(13+I)) - CLAT
    DCLAT(I) = 0.5 * DCLAT(I)
    RCLA(4+I) = DCLAT(I)
144 CONTINUE
C ***** DEFINE EFFECT OF HOLES IN WING OF LIFT-CLIVE *****
C ***** SLOPE AND POLAR-SHAPE FACTOR *****
C
146 DCLAMD = 0.0
    DKHCL = 0.0
    IF (IDAM(14).NE.1) GO TO 210
C
    HOLE AREA = NO. OF HOLES * LENGTH * WIDTH
    AREAUP = DSUR(1,32) * DSUR(1,33) * CSLF(1,34)
    AREALC = DSUR(1,31) * DSUR(1,33) * CSLF(1,34)
    FLOW AREA = HOLE AREA * POROSITY FACTOR
    FLOW1 = AREAUP * DSUR(1,39)
    IF (AREALC.LT.AREAUP) FLOW1 = AREALC * DSUR(1,39)
C
    FLOW2 = 0.0
    IF (MPNLS.EQ.1) GO TO 200
    AREAUP = DSUR(2,32) * DSUR(2,33) * DSUP(2,34)
    AREALC = DSUR(2,32) * DSUR(2,33) * DSUP(2,34)
    FLOW2 = AREAUP * DSUR(2,39)
    IF (AREALC.LT.AREAUP) FLOW2 = AREALC * DSUR(2,39)
C
200 IF (SYM(1,14).EQ.1.C) FLOW1 = 2. * FLOW1
    IF (SYM(2,14).EQ.1.C) FLOW2 = 2. * FLOW2
    FLOW = FLOW1 + FLOW2
    FLOWR = FLOW / (SPAN/2.)
C
    CALL INTF(FLOWR, CLAFAC, FLOW1, CLAP1, 5, 4)
    CALL INTF(FLOWR, DEFAC, FLOW2, EFF2, 5, 4)
    DCLAMD = CLA * CLAFAC - CLA
    DKHCL = 1. / (3.14 * EF * DEFAC * AR) - 1. / (3.14 * EF * AP)
    IF (KPRINT(29).EQ.1)
1WRITE(6,110) FLOW1, FLOW2, FLOWR, CLAFAC, DEFAC, DCLAMD, DKHCL
210 CONTINUE
C ***** SUM EFFECTS ON UNTRIMMED CL, CD, AND CM *****
C ***** CAUSED BY DAMAGE *****
C
    DCLA = DCLA1 + (DCLA2(1) + DCLA2(2) + DCLAT(1) + DCLAT(2) + DCLAMD)
    DXAC = DXAC1 + DXAC2(1) + DXAC2(2)
    DK = DK1 + DK2(1) + DK2(2) + DKHCL
    IF (KPRINT(29).EQ.1)
1WRITE(6,1070) DCLA, DXAC, DK, (RCLA(I), I = 1,6)
C
    CALCULATE UNTRIMMED LIFT, DRAG, AND MOMENT
    FOR THE DAMAGED AIRPLANE
C
    CLAP = CLA + DCLA
    IF (KPRINT(29).EQ.1)
1WRITE(6,1095)
    DO 150 I = 1, NCLAS
C
    XCL = CLA * (ALP(I) - ALC)
    YCLP = CLAP * (ALP(I) - ALD)
    CLP(I) = CL(I) + (XCLP - YCL)
C
    XCLT = CLA * (ALP(I) - ALC) + (DCLAT(1) + DCLAT(2)) * (ALP(I) - HSTAF)
    XCD = XK + XCL**2
    XCDT = XK + XCLT**2
    CDTAIL = XCD - XCDT
    CALL INTF(CLP(I), CDX, CL, CD, NCLAS, 4)
    CDP(I) = CDX + CDDMIN + DK * CLP(I)**2 + CDTAIL
C
    CLWNGP = (CLAM + DCLA1 + (DCLA2(1) + DCLA2(2) + DCLAMD) * (ALP(I) - ALC)
    CWING = -CLWNGP * FAFAC
    CMTAIL = -(DCLAT(1) + DCLAT(2)) * (ALP(I) - HSTAF) * YH/CMAC

```



```

CALL LINTP(CLP(I), CMX, CL, CM, NCLAS, 4)
CMP(I) = CMY + CMWING + CMTAIL
IF (KFPINT(2C).EQ.1)
1 WRITE(6,1100)(ALP(I), CL(I), CLP(I), CM(I), CMTAIL, CMY,
2 CMF(I), CM(I), CMWING, CMTAIL, CMY, CMP(I))
C 150 CONTINUE
C SWPCC = SWPCC / 57.2958
1010 FORMAT ( ///, 10X, *DUMP FROM SUBROUTINE MUCCAM2, *MACH = *,
1 FE.3, //, 10X, * DCLA1, DCLA2, DCLA3, DCLA4, DCLA5, *,
2 *DCLA6 = *, 6F10.4, /, 10X,
3 *CLARTF, CLARTF, CLAP = *, 3F10.4)
1020 FORMAT ( 10X, *DCLA1 = *, F10.4)
1030 FORMAT ( /, 10X, *DYAC1, DYAC2, DYAC3, DYAC7, YXX = *, 5F10.4)
1035 FORMAT ( 10X, *DYAC4, DYAC5, DYAC6, DYAC7, YYY = *, 5F10.4)
1036 FORMAT ( 24X, *CXAC1 = *, F10.4)
1040 FORMAT ( /, 10X, *K, PP, FF, EFP, YK, XKP, CK1 = *, 7F8.4)
1045 FORMAT ( //, 10X, *DATA FOR LEFT WING, I = *, I2)
1046 FORMAT ( //, 10X, *DATA FOR RIGHT WING, I = *, I2)
1050 FORMAT ( 10X, *RWP, CTXP, SPLAMP, AFP, TRP, CLAP, CLAW, *,
1 *SONIC, VCRK, POTENT, VORTEX, DCLA2(I) **,
2 5F10.5)
3
1061 FORMAT ( 10X, *CBAP1, CBAP2, ETA1, ETA2, XLF1, XLE2, *,
1 *DYAC2(I) = *, 7F8.3)
1064 FORMAT ( 10X, *FCF, DK2(I) = *, 25X, 2F10.4)
1070 FORMAT ( /, 10X, *TOTAL INCREMENTS FOR THE CONFIGURATION *,
1 /, 10X, *DCLA, DYAC, DK = *, 3F10.4, /, 10X,
2 *DCLA(1), DCLA(2), DCLA(3), DCLA(4), DCLA(5), DCLA(6) = *,
3 6F10.5)
1090 FORMAT (1H1, ///, 10X, *DATA IN COMMON BLOCK OUTPLT, (B MATRIX)*,
1 //, 10X, *MACH = *, F10.3, //, (10X, 10F10.5))
1095 FORMAT(1H1, //, 3X,
1 * ALP(I) CL(I) CLP(I) CM(I) CMTAIL *,
2 * CMX CM(I) CMWING *,
3 * CMTAIL CMX CMP(I) *, //)
1100 FORMAT(?, 12F10.5)
1110 FORMAT( /, 10X, *FACTORS FOR HOLES IN WING, FLOW1, FLOW2, FLOW3 = *,
1 3F10.4, /, 25X, *CLAFAC, DEFAC, ECLAFAC, CMFCL = *, 4F10.4)
RETURN
END

```



```

C
C
RCL3 = ROLA(3) * (ALP(I))-ALC)
CLWING = (CLAW/2.)* (ALP(I)-ALC)
CPUND = (CLWING * X3(1)) / (2. *PC2)
CRC = (CLWING+RCL3) * X3(2) / (2. *PC2)
CRCLL3 = CPD - CRUND

C
C
RCL4 = ROLA(4) * (ALP(I)-ALC)
CLWING = (CLAW/2.)* (ALP(I)-ALC)
CPUND = -CLWING * X4(1) / (2. *PC2)
CRC = -(CLWING+RCL4) * X4(2) / (2. *PC2)
CRCLL4 = CRC - CPUND

C
C
RCL5 = ROLA(5) * (ALP(I)-HSTAP)
CRCLL5 = RCL5 * (X5 / (2. *PC2))

C
C
RCL6 = ROLA(6) * (ALP(I)-HSTAP)
CRCLL6 = -RCL6 * (X6 / (2. *PC2))

C
C
THE TOTAL ROLLING MOMENT IS DEFINED AS:
CRCLL(I) = CRCLL1+CRCLL2+CRCLL3+CRCLL4+CRCLL5+CRCLL6
IF (KPRINT(20).EQ.1)
1WRITE(6,1040) RCL1, RCL2, RCL3, RCL4, RCL5, RCL6,
2CRCLL1, CRCLL2, CRCLL3, CRCLL4, CRCLL5, CRCLL6,
3CRCLL(I)
XBAP = CLWING(13) * PC2

C
C
DELTA-CL REQUIRED ON TRIMMING DEVICE IS
CLRT = ABS((CRCLL(I) * 2.0 * PC2) / XBAP )

C
C
RESULTING DRAG
CDRT(I) = CLRT * TAN(ALP(I) / 57.2957)

C
C
CALCULATE TOTAL LIFT AND DRAG FOR THE DAMAGED AIRCRAFT
INCLUDING PITCH AND ROLL TRIM

CLTP(I) = CLP(I) + CLRT
CDTP(I) = CDP(I) + CDCT + CDRT(I)

C
IF (KPRINT(30).EQ.1)
1WRITE(6,1050) CLP(I), CDCT, CLTP(I),
3CDP(I), CDCT, CDRT(I), CDTP(I)
100 CONTINUE

C
1010 FORMAT(1H1, //, 10X, *DUMP FROM SUBROUTINE NTRIM, MACH=*, F10.3, //, 31X,
1 *CLAT, CLPH, XACWB, >ACS *, /, 10X,
2 *FCR UNDAAGED A/C*, 4F10.3)
1020 FORMAT( /, 10X, *FCR DAMAGED A/C*, 4F10.3)
1025 FORMAT( /, 5X,
1 *RCLA(1), RCLA(2), RCLA(3), RCLA(4), RCLA(5), RCLA(6) =*,
2 6F8.3, /, 5X,
3 *X1, X2, X3(1), X3(2), X4(1), X4(2), X5, X6 =*,
4 6F8.3, /)
1030 FORMAT( /, 22X, *ALPHA, CL1, CD1 =*, F10.3, 2F10.5)
1040 FORMAT( /, 5X,
1 *RCL1, RCL2, RCL3, RCL4, RCL5, RCL6 =*,
2 6F8.3, /, 5X,
3 *CRCLL1, CRCLL2, CRCLL3, CRCLL4, CRCLL5, CRCLL6 =*,
4 6F8.3, /, 10X, *TOTAL ROLLING MOMENT, CRCLL(I) =*, F10.3)
1050 FORMAT( /, 10X, *CLP(I), CDCT, CLTP(I), =*, 3F10.4, /,
2 10X, *CDP(I), CDCT, CDRT(I), CDTP(I) =*, 4F10.5)
PUTUPN
END

```



```

2030 FORMAT ( 9X, *CL X ALPHA CM CD X ALPHA CM*,
1 * CD X ALPHA CM FOLL CD MIN*,
2 * CCL CD TOTAL X*, /, 14X, *Y*, 26X,
3 *X*, 26X, *Y*, 54X, *)
2040 FORMAT ( 6X, F6.3, * X *, F6.2, F6.4, F6.5, * Y *, F6.2,
1 F6.4, F6.5, * X *, F6.2, 2F6.4, F6.5, 2F6.5, * X *)
2050 FORMAT ( //, 35X, *I P I M M E C A L E F C D Y N A M I C *,
1 D A T A *)
2060 FORMAT ( 9X, *CL X ALPHA CD TRIM CD FOLL X ALPHA *,
1 * CD X ALPHA CD TRIM CD FOLL CD MIN (CL*,
2 * CD TOTAL X *, /, 14X, *Y*, 26X, *X*, 26X, *)*, 54X, *X*)
2070 FORMAT ( 6X, F6.3, * X *, F6.2, 6X, F6.5, * X *, F6.2, 9X, F6.5,
1 * X *, F6.2, F6.5, 4F6.5, * X *)
3000 FORMAT(1H1, //, 15X, *CALCULATEC DATA AT EVEN VALUES OF ALPHA*, //,
1 5X, * ALPHA(I) CL(I) CLP(I) CLT(I) CLTP(I) *,
2 * CM(I) CMP(I) CD(I) CCF(I) COT(I) *,
3 * CDT(I) *, //, (5X, 11F10.5))
3010 FORMAT (/////,
1 15X, *ALP = ANGLE OF ATTACK *, //,
2 15X, *CL CD, CM = UNDA/AGED, L*TRIMMED *, //,
3 15X, *CLP CDF, CMP = DAMAGED, LATRIMMED *, //,
4 15X, *CLT CDT = UNDAAGED, TRIMMED *, //,
5 15X, *CLTP, CDTP = DAMAGED, TRIMMED * )
RETURN
END

```

## DISTRIBUTION LIST

### DEPARTMENT OF DEFENSE

Assistant to the Secretary of Defense  
Atomic Energy

ATTN: Executive Assistant

Defense Documentation Center  
12 cy ATTN: DD

Defense Nuclear Agency

ATTN: DDST

ATTN: SPAS

ATTN: STSP

4 cy ATTN: TITL

Field Command  
Defense Nuclear Agency  
ATTN: FCPR

Livermore Division, Field Command, DNA  
Lawrence Livermore Laboratory  
ATTN: FCPRL

Commandant  
NATO School (SHAPE)  
ATTN: U.S. Documents Officer

Under Secretary of Defense for Rsch. & Engrg.  
ATTN: Strategic & Space Systems (OS)

### DEPARTMENT OF THE ARMY

Harry Diamond Laboratories  
ATTN: DELHD-N-NP  
ATTN: DELHD-N-P, J. Gwaltney

U.S. Army Ballistic Research Labs.  
ATTN: DRXBR-BLE, W. Taylor  
ATTN: DRDAR-BLV, J. Meszaros

U.S. Army Materiel Dev. & Readiness Cmd.  
ATTN: DRCDE-D, L. Flynn

U.S. Army Nuclear & Chemical Agency  
ATTN: Library

### DEPARTMENT OF THE NAVY

Naval Material Command  
ATTN: MAT 08T-22

Naval Research Laboratory  
ATTN: Code 2627

Naval Surface Weapons Center  
ATTN: K. Caudle

Naval Weapons Evaluation Facility  
2 cy ATTN: L. Oliver

Office of Naval Research  
ATTN: Code 464

Strategic Systems Project Office  
ATTN: NSP-272

### DEPARTMENT OF THE AIR FORCE

Aeronautical Systems Division, AFSC

ATTN: ENFT, R. Bachman

4 cy ATTN: ENFTV, D. Ward

10 cy ATTN: ENFTA, R. Sarver

Air Force Materials Laboratory  
ATTN: MBE, G. Schmitt

Air Force Weapons Laboratory

ATTN: SUL

ATTN: DYV, G. Campbell

4 cy ATTN: DYV, A. Sharp

Foreign Technology Division, AFSC  
ATTN: PDBF, Mr. Spring

Strategic Air Command/XPFS  
ATTN: XPFS, B. Stephan

### DEPARTMENT OF DEFENSE CONTRACTORS

Aerospace Corporation  
ATTN: W. Barry

Avco Research & Systems Group  
ATTN: J. Patrick  
ATTN: P. Grady

Boeing Company  
ATTN: S. Strack  
ATTN: R. Dyrdaahl  
ATTN: E. York

Boeing Wichita Company  
ATTN: R. Syring

Calspan Corporation  
ATTN: M. Dunn

Effects Technology, Inc.  
ATTN: R. Parisse  
ATTN: E. Bick  
ATTN: R. Wengler

General Dynamics Corporation  
Pomona Division  
ATTN: R. Schemensky

General Dynamics Corporation  
Forth Worth Division  
ATTN: G. Howell

General Electric Co.-TEMPO  
Center for Advanced Studies  
ATTN: DASIAC

General Research Corp.  
Santa Barbara Division  
ATTN: T. Stathacopoulos

Kaman Avidyne  
Division of Kaman Sciences Corp.  
ATTN: E. Criscione  
ATTN: N. Hobbs  
ATTN: R. Ruetenik

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Kaman Sciences Corporation  
ATTN: D. Sachs

Los Alamos Technical Associates  
ATTN: P. Hughes

McDonnell Douglas Corporation  
ATTN: J. McGrew

Prototype Development Associates, Inc.  
ATTN: C. Thacker  
ATTN: J. McDonald

R&D Associates  
ATTN: C. MacDonald  
ATTN: F. Field  
ATTN: J. Carpenter

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Rockwell International Corp.  
ATTN: R. Sparling

Sandia Laboratories  
ATTN: Doc. Con. for A. Lieber

Science Applications, Inc  
ATTN: D. Hove

SRI International  
ATTN: G. Abrahamson