

UNCLASSIFIED

AD NUMBER: AD0853820

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to DoD only; Administrative/Operational Use; Jun 1969. Other requests shall be referred to Space and Missile Systems Organization, Norton AFB, CA.

AUTHORITY

SAMSO ltr dtd 19 Jan 1972

ADVANCED DECOY TECHNOLOGY PROGRAM
ADTECH IV
FINAL REPORT (U)

APPENDIX I, PART II
PROGRAM DESCRIPTION--OPTIMUM DECOY DESIGN PROGRAM

Prepared by

AVCO GOVERNMENT PRODUCTS GROUP
MISSILE SYSTEMS DIVISION
201 Lowell Street
Wilmington, Massachusetts 01887

AVMSD-0465-68-RR, APP. I
Contract F04701-68-C-0012

June 1969

Sponsored by

Advanced Research Projects Agency
Department of Defense
ARPA Order No. 441, Amendment No. 12

THIS DOCUMENT IS SUBJECT TO SPECIAL EXPORT CONTROLS
AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR
FOREIGN NATIONALS MAY BE MADE ONLY WITH PRIOR APPROVAL
OF SPACE AND MISSILE SYSTEMS ORGANIZATION (SMISO)

THE DISTRIBUTION OF THIS REPORT IS LIMITED BECAUSE IT
CONTAINS TECHNOLOGY REQUIRING DISCLOSURE ONLY WITH-
IN THE DEPARTMENT OF DEFENSE.

Att: SMISO *NA-AFS, Calif. 90045-*

Prepared for

SPACE AND MISSILE SYSTEMS ORGANIZATION
DEPUTY FOR REENTRY SYSTEMS
AIR FORCE SYSTEMS COMMAND
Norton Air Force Base, California 92409

D D C
RECEIVED
JUN 19 1969
REGISTERED
B

FOR OFFICIAL USE ONLY

AD853820

SAMSO TR-69-140, APP. I

This document consists of 810 pages,
126 copies, Series A

ADVANCED DECOY TECHNOLOGY PROGRAM
ADTECH IV
FINAL REPORT (U)
APPENDIX I, PART II
PROGRAM DESCRIPTION--OPTIMUM DECOY DESIGN PROGRAM

Prepared by

AVCO GOVERNMENT PRODUCTS GROUP
MISSILE SYSTEMS DIVISION
201 Lowell Street
Wilmington, Massachusetts 01887

AVMSD-0465-68-RR, APP. I
Contract F04701-68-C-0012

by

R. A. MacFarlane
E. R. Nickerson

June 1969

Sponsored by

Advanced Research Projects Agency
Department of Defense
ARPA Order No. 441, Amendment No. 12

THIS DOCUMENT IS SUBJECT TO SPECIAL EXPORT CONTROLS
AND EACH TRANSMITTAL TO FOREIGN GOVERNMENTS OR
FOREIGN NATIONALS MAY BE MADE ONLY WITH PRIOR APPROVAL
OF SPACE AND MISSILE SYSTEMS ORGANIZATION (SMSS)

THE DISTRIBUTION OF THIS REPORT IS LIMITED BECAUSE IT
CONTAINS TECHNOLOGY REQUIRING DISCLOSURE ONLY WITH-
IN THE DEPARTMENT OF DEFENSE.

Prepared for

SPACE AND MISSILE SYSTEMS ORGANIZATION
DEPUTY FOR REENTRY SYSTEMS
AIR FORCE SYSTEMS COMMAND
Norton Air Force Base, California 92409

UNCLASSIFIED ABSTRACT

(U) This technical report describes analyses and techniques used in the design and evaluation of advanced decoy concepts. The work described addresses both the design of specific penetration aid elements and the formulation of techniques for their evaluation. The three major technical areas covered in this report are:

1. Investigation of a penetration aid technique that degrades the measurement capability of the radar sensor.
2. The design of a computer program to solve the decoy design problem with flexibility in the selection of optimization criteria and constraints.
3. Studies of the use of certain discrimination techniques for a hard point defense system.

This appendix to this report contains detailed description of the optimum decoy design program.

TABLE OF CONTENTS

PART I

1.0	INTRODUCTION AND PRELIMINARY OPERATIONS	I-I
1.1	Introduction.....	I-2
1.2	Main Program.....	I-6
1.3	Input Subroutines	I-13
1.3.1	READIT	I-14
1.3.2	ZREADX	I-31
1.4	Presetting Operations	I-48
1.4.1	SR2490.....	I-49
1.4.2	ZPRM	I-56
1.4.3	ZPRS	I-60
2.0	OPTIMIZATION TECHNIQUES	I-64
2.1	Penalty Function Transformation	I-66
2.1.1	FEV	I-67
2.1.2	SCREEN	I-73
2.1.3	REDUCE	I-80
2.2	Gradient of Penalty Function - FCN.....	I-83
2.3	Search Logic	I-88
2.3.1	Davidon Method	I-89
2.3.1.1	DAVDON	I-91
2.3.1.2	READY	I-99
2.3.1.3	AIM	I-105
2.3.1.4	FIRE.....	I-113
2.3.1.5	DRESS.....	I-120
2.3.1.6	STUFF.....	I-126
2.3.1.7	RANDOM.....	I-130
2.3.1.8	MATMP	I-131
2.3.2	Rosenbrock Method	I-133
2.3.2.1	ROSBRK.....	I-134
2.3.2.2	GRAM.....	I-141
2.3.3	One Variable Fibonacci	I-144
2.3.3.1	MIMAX	I-145
2.3.3.2	FMIMAX.....	I-156
2.3.4	Two Variable Fibonacci	I-158
2.3.4.1	GIMAX	I-159
2.3.4.2	GMIMAX.....	I-164

3.0	BASIC ANALYSIS CALCULATIONS	I-167
3.1	Trajectory Calculations	I-169
3.1.1	Initializing and Printout Operations	I-170
3.1.1.1	VIXEN	I-171
3.1.1.2	CHNTBL	I-188
3.1.1.3	RITOUT	I-198
3.1.2	Numerical Integration - ADM4RK	I-200
3.1.3	Evaluation of Derivatives - DEREQ	I-215
3.1.4	Preliminary Calculations	I-226
3.1.4.1	PRELIM	I-227
3.1.4.2	ARFDT2	I-239
3.1.4.3	COMP62	I-241
3.1.4.4	LNTERP	I-248
3.1.4.5	TABLE	I-251
3.1.5	Aerodynamic Heating, Mass Loss, and Nose Blunting Conditions	I-254
3.1.5.1	AERODY	I-255
3.1.5.2	MASSLO	I-266
3.1.5.3	EVIL	I-271
3.1.5.4	TOMALO	I-286
3.1.5.5	NOSEBL	I-291
3.1.6	Rotational Calculations	I-295
3.1.6.1	ROTATE	I-296
3.1.6.2	BESSEL	I-308
3.1.6.3	JNXBES	I-311
3.1.6.4	DRLIM	I-316
3.1.6.5	NEUMAN	I-318
3.1.6.6	NEUMPO	I-325
3.1.6.7	NEUMQO	I-329
3.1.6.8	DIVMLT	I-333
3.1.6.9	NEUMNO	I-335
3.1.6.10	NEUMN1	I-339
3.1.7	Drag Calculations - DRAGCO	I-343
3.1.8	Translational Calculations	I-374
3.1.8.1	TEQUAT	I-375
3.1.8.2	MATMPY	I-382
3.2	Wake Calculations	I-384
3.2.1	Preliminary Calculations - WAKE	I-386
3.2.2	Flow Field Calculations	I-393
3.2.2.1	FLOWF	I-394
3.2.2.2	AR3DIM	I-423
3.2.2.3	AR2DIM	I-425

3.2.3	Radar Cross Section Calculation	I-427
3.2.3.1	RCSEC	I-428
3.2.3.2	FIB1	I-437
3.2.3.3	FUN1	I-440
3.3	Miscellaneous Calculations	I-448
3.3.1	MISC	I-449
3.3.2	POLCAL	I-454
PART II		
4.0	COMPARISON OF DECOY WITH REENTRY VEHICLE - F123	I-457
4.1	Integration of Special Functions	I-478
4.1.1	INTGRL	I-479
4.1.2	ADD	I-489
4.1.3	INFCOF	I-491
4.1.4	INTERP	I-495
4.1.5	LINFIT	I-497
4.1.6	WRITEM	I-498
4.2	Storage of Geometric Parameters - SAVEDV	I-501
4.3	Miscellaneous Printout Operations	I-503
4.3.1	HEADER	I-504
4.3.2	TIMERS	I-506
5.0	EFFECTIVENESS OPERATIONS - EFFECT	I-508
6.0	CLASSIC FUNCTIONS FOR TESTING OPTIMIZATION TECHNIQUES - CLASSC	I-514
7.0	PLOTTER INTERFACE SUBROUTINES	I-518
7.1	AVPLT	I-519
7.2	MAXMIN	I-525
7.3	PLT	I-527
8.0	LIBRARY ROUTINES	I-528
8.1	ACOSR	I-529
8.2	ASINR	I-531
8.3	ATANQR	I-533
8.4	FIXIBCOM (MIBCOM) (dummied)	I-535
8.5	FERROR (dummied)	I-536
8.6	LA000000 (HEDING, READIN, SETUP, WHERE)	I-537
8.7	Dummied Plotting Routines (ENDJOB, EZPLOT, FRAMEV, IDFRMV, PLTND)	I-552
	APPENDIX I - Program Listing and Preset Data	I-553
	APPENDIX II - Overlay and Other Information Pertinent to running the program	I-793

BLANK PAGE

4. Numerical Procedure

First the coefficient counter M and the accumulator ANS are initialized to zero. Then ITP is tested. If ITP is greater than one, statement 10 is executed next, setting NAZU equal to NAX and control then goes to statement 30. In this case, a polynomial in the three unknowns X, Y and Z will be evaluated. If ITP is less than or equal to 1, statement 20 is executed next where NAZU is set equal to 1 and statement 30 is reached. In this case, the polynomial will be a function of only two unknowns X and Y.

At statement 30, the DO loop ending at statement 130 is entered to perform the calculations needed to evaluate the polynomial. The value of this polynomial is returned to MISC in the variable ANS. When the evaluation is complete, the RETURN statement following 130 is executed, returning control to MISC.

5. Other Information

- A. POLCAL is called by SUBROUTINE MISC only.
- B. POLCAL does not call or reference any other subprogram.

4.0 COMPARISON OF DECOY WITH REENTRY VEHICLE

Comparison of the Decoy with Reentry Vehicle

The comparison of the decoy with the reentry vehicle consists of the evaluations of certain differences and the determination of integrals of special functions. SUBROUTINE F123 performs the difference calculations using SAVEDV to store the values, before calling INTGRL. SUBROUTINE INTGRL evaluates the integrals using in the process FUNCTION ADD, and subroutines INFCOF, INTERP, LINFIT, and WRITEM. F123 also calls in subroutines HEADER AND TIMERS. These integrals include the corridor functions use' to constrain the decoy within specified corridors and the effectiveness integrals which are used in the probability of discrimination calculations in SUBROUTINE EFFECT.

SUBROUTINE F123(XXX)

1. Purpose

F123 can optionally

- a. Read the data for a reentry vehicle trajectory from input or call SUBROUTINE VIXEN to calculate this trajectory data.
- b. Call VIXEN to calculate the data for a decoy trajectory.
- c. Compare the data for a reentry vehicle trajectory with the data for a decoy trajectory and save tables of differences.
- d. Call INTGRL to calculate and plot a corridor integral.
- e. Call INTGRL to calculate the sigma square integral to be used by SUBROUTINE EFFECT.
- f. CALL INTGRL to calculate a table of influence coefficients and produce influence coefficient plots at specified altitudes.

2. Input

Name	Common Location	Source of Input	Description
NOTE The numbers in the Common Location column refer to positions in the <u>ØCCUR</u> array.			
ALST	122	READIT	Stopping value of α envelope, in degrees
CASE	128	READIT	Case number
DATE	127	READIT	Date
DELY	219	READIT	Linear thrust offset along Y axis in inches
DELZ	220	READIT	Linear thrust offset along Z axis in inches
EMØ	129	READIT	Memo number
GAMF0	105	READIT	Initial flight path angle in degrees
GAMMA	028	READIT or SR2490	Ratio of specific heats
LA1	138	READIT	Axial length of initial configuration, in inches
LA2	144	READIT	Axial length of vehicle after shape change at ZTURN, in inches
LP	4000	VIXEN	Error code
MW	117	READIT or SR2490	Molecular weight of air, gram/mole
PHI0	112	READIT	Initial Euler angle Φ , in degrees
PSIZET	223	READIT	Thrust misalignment angle, in degrees
PSI0	114	READIT	Initial Euler angle Ψ , in degrees
RB1	136	READIT	Base radius of initial configuration, in inches

2. Input (cont'd)

Name	Common Location	Source of Input	Description
RB2	142	READIT	Base radius of vehicle after shape change at ZTURN, in inches
RN1	135	READIT	Nose radius of initial configuration, in inches
RN2	141	READIT	Nose radius of vehicle after shape change, at ZTURN, in inches
SIG	116	READIT or SR2490	Collision cross section in angstroms
THEAL0	113	READIT	Initial Euler angle θ_0 , in degrees
THEZET	224	READIT	Thrust misalignment angle in degrees
NOTE Unless otherwise indicated, the entries in the Common Location Column refer to positions in the <u>PCCUR</u> array.			
BCB, 40	6181-6220	ZREADX	Array of lower corridor values for ballistic coefficient
BCD, 40	6101-6140	ZREADX	Array of lower corridor values for deceleration in g's
BCV, 40	6021-6060	ZREADX	Array of lower corridor values for velocity in ft/sec.
BCWL1, 40	6501-6540	ZREADX	Array of lower corridor values for wake length in meters for first radar frequency
BCWL2, 40	6581-6620	ZREADX	Array of lower corridor values for wake length in meters for second radar frequency
BCWL3, 40	6661-6700	ZREADX	Array of lower corridor values for wake length in meters for third radar frequency
BCWR1, 40	6261-6300	ZREADX	Array of lower corridor values for wake radar cross section at first radar frequency
BCWR2, 40	6341-6380	ZREADX	Array of lower corridor values for wake radar cross section at second radar frequency

Input (cont'd)

Name	Common Location	Source of Input	Description
BCWR3, 40	6421-6460	ZREADX	Array of lower corridor values for wake radar cross section at third radar frequency
BETAPL, 160	641 - 800	ZREADX or VIXEN	Ballistic coefficients for the input or calculated trajectory points
DVAR, 50	11707-11756	SAVEDV	Array in which design characteristics are saved
DVH, 50	11607-11656	ZREADX	Input values of design variables for the second perturbation of comparison decoy
DVL, 50	11657-11706	ZREADX	Input values of design variables for the first perturbation of comparison decoy
H , 40	5841-5880	ZREADX	Input altitudes corresponding to corridor points, in feet
SB, 40	6821-6860	ZREADX	Standard deviation of radar measurement errors for ballistic coefficient
SD, 40	6781-6820	ZREADX	Standard deviation of radar measurement errors for deceleration in g's
SV, 40	6741-6780	ZREADX	Standard deviation of radar measurement errors for velocity in ft/sec.
SWL1, 40	6981-7020	ZREADX	Standard deviation of radar measurement errors for wake length in meters at first radar frequency
SWL2, 40	7021-7060	ZREADX	Standard deviation of radar measurement errors for wake length in meters at second radar frequency
SWL3, 40	7061-7100	ZREADX	Standard deviation of radar measurement errors for wake length in meters at third radar frequency.

Input (cont'd)

Name	Common Location	Source of Input	Description
TPLØT, 160	1-160	ZREADX or VIXEN	Times for the input or calculated trajectory points in seconds
VOGPLT, 160	481-640	ZREADX or VIXEN	Deceleration in g's for input or calculated trajectory points
VPLØT, 160	321-480	ZREADX or VIXEN	Velocities in ft/sec. for the input or calculated trajectory points
WL1P, 160	1281-1440	ZREADX or VIXEN	Wake length for the first radar frequency input or calculated trajectory points in meters
WL2P, 160	1441-1600	ZREADX or VIXEN	Wake length for the second radar frequency input or calculated trajectory points in meters
WL3P, 160	1601-1760	ZREADX or VIXEN	Wake length for the third radar frequency input or calculated trajectory points in meters
WR1P, 160	801-960	ZREADX or VIXEN	Wake cross section for the first radar frequency for input or calculated trajectory points, see IDBL for units
WR2P, 160	961-1120	ZREADX or VIXEN	Wake cross section for the second radar frequency for input or calculated trajectory points
WR3P, 160	1121-1280	ZREADX or VIXEN	Wake cross section for the third radar frequency for input or calculated trajectory points
WW	-----	LINFIT	Coefficient of the independent variable in a linear fit
XX	-----	LINFIT	Constant in a linear fit
ZPLØT, 160	161-320	ZREADX or VIXEN	Altitudes for the input or calculated trajectory points, feet

Input (cont'd)

Name	Common Location	Source of Input	Description
SWR1, 40	6861-6900	ZREADX	Standard deviation of radar measurement errors for wake cross section at first radar frequency
SWR2, 40	6901-6940	ZREADX	Standard deviation of radar measurement errors for wake cross section at second radar frequency
SWR3, 40	6941-6980	ZREADX	Standard deviation of radar measurement errors for wake cross section at third radar frequency
TCB, 40	6221-6260	ZREADX	Array of upper corridor values for ballistic coefficient
TCD, 40	6141-6180	ZREADX	Array of upper corridor values for deceleration in g's
TCV, 40	6061-6100	ZREADX	Array of upper corridor values for velocity in ft/sec.
TCWL1, 40	6541-6580	ZREADX	Array of upper corridor values for wake length in meters for first radar frequency
TCWL2, 40	6621-6660	ZREADX	Array of upper corridor values for wake length in meters for second radar frequency
TCWL3, 40	6701-6740	ZREADX	Array of upper corridor values for wake length in meters for third radar frequency
TCWR1, 40	6301-6340	ZREADX	Array of upper corridor values for wake radar cross section at first radar frequency
TCWR2, 40	6381- 6420	ZREADX	Array of upper corridor values for wake radar cross section at second radar frequency
TCWR3, 40	6461-6500	ZREADX	Array of upper corridor values for wake radar cross section at third radar frequency

Input (cont'd)

Name	Common Location	Source of Input	Description
*** NOTE*** The following quantities are integers:			
IREF	IØCCUR(301)	ZREADX	Trajectory processing option code, see ZREADX
IØP, 90	IØCCUR(1-90)	ZREADX	Input integer codes, see users manual
LØPT	NØCCUR(07)	READIT	Trajectory option code, see READIT
LPLØT	IØCCUR(302)	ZREADX or VIXEN	Number of points in the input or calculated trajectory
MHEAT	NØCCUR(10)	READIT	Option code for mass loss calculation
MØDE	IØCCUR(303)	ZREADX	Fundamental option code, see ZREADX
MØPT	NØCCUR(03)	READIT	Option code for aerodynamic heating calculation
NCØMDV, 50	IØCCUR(91-140)	ZREADX	Identification code numbers (indices in the ØCCUR array) of design variables to be perturbed under MØDE =2 influence coefficient calculations
NCP	IØCCUR(304)	ZREADX	Number of corridor points
NDECØY	IØCCUR(305)	ZREADX	Vehicle type identification code, see ZREADX
NDVCH	IØCCUR(306)	ZREADX	Number of entries in NCØMDV table, i. e., number of design variables
NØSEØP	NØCCUR(05)	READIT	Option code for noseblunting calculation, see READIT
NTHRST	NØCCUR(22)	READIT	Option code for thrusting, see READIT

3. Output

Name	Common Location	Description
NOTE Numbers in Common Location Column refer to positions in <u>ØCCUR</u> array for the following quantities:		
ALMAX, 200	1044-1243	Maximum in α at the Vixen point altitudes in radius
ALMIN, 200	1244-1443	Minimum in α at the Vixen point altitudes in radius
ALST	122	The stepping control on envelope α , preset to 0.2 degrees
CASE	128	Case number
CDB	99	Base drag coefficient
CDFINF	2793-2800	Skin friction drag coefficient
CDI	100	Induced drag coefficient
CDP	98	Pressure drag coefficient including
CDP0	101	Pressure drag coefficient for zero angle of attack
CM	202	Moment coefficient about c.g., C_m
CN	203	Normal force coefficient, C_n
CØDLAM	94	$7.6489 \times 10^{-9} M_w / \sigma^2$, used in PRELIM to define CØDRAG which is then used in DRAGCØ
CPSZET	216	Cosine of the thrust offset angle ψ_s PSIZET
CTHZET	215	Cosine of the thrust offset angle θ_s THEZET

Output (cont'd)

Name	Common Location	Description
DECDFP	235	The pressure induced skin friction drag coefficient
DECFTC	236	The transverse curvature induced skin friction drag coefficient
DELCDP	234	The induced pressure drag coefficient
DELY	219	The linear thrust offset component along Y axis, in feet
DELZ	220	Linear thrust offset along Z axis, in feet
FACTR9	197	Numerical factor used in DRAGCØ
FMAX, 200	2244-2443	Frequencies at the altitudes corresponding to ALMAX values, cycles/sec
FMIN, 200	2444-2643	Frequencies at the altitudes corresponding to ALMIN values, cycles/sec
GAMFØ	105	Initial flight path angle in radians
LA1	138	See input, in feet
LA2	144	See input, in feet
LP	4000	Error code
MDØT, 32	2708-2739	Distribution along the body of ablative mass loss rates, in $\text{lbm/ft}^2\text{-sec}$
MX	210	thrusting moment about X axis
MY	211	thrusting moment about Y axis
MZ	212	thrusting moment about Z axis
OCCUR, 4000	1-4000	Common array

Output (cont'd)

Name	Common Location	Description
PEPSB, 8	2801-2808	Distribution along body of the ratio of edge pressure to stagnation pressure dimensionless
PHI0	112	Euler angle Φ , in radians
PSI0	114	Euler angle Ψ , in radians
QD0T, 32	2676-2707	Distribution along the body of cold wall aerodynamic heating rates
Q0	110	Initial value of the angular rate \dot{Q} in radians/sec.
RBD0T	60	Time rate of change of base radius due to ablation, ft/sec
RB1	136	See input, in feet
RB2	142	See input, in feet
RND0T	59	Time rate of change of nose radius due to ablation, ft/sec.
RN1	135	See input, in feet
RN2	141	See input, in feet
SING0	221	Sine of initial flight path angle
SMR0	111	Initial value of the angular rate R in radians/sec
SPSZET	218	Sine of input PSIZET
STHZET	217	Sine of input THEZET
TAMAX, 200	1444-1643	Times corresponding to maximums in α , in seconds
TAMIN, 200	1644-1843	Times corresponding to minimums in α , in seconds

Output (cont'd)

Name	Common Location	Description
TBMAT, 3	3643-3645	Components of thrust in body axis system, lb
THEAL0	113	Initial Euler angle θ_x , in radians
WDOT	86	Rate of change of vehicle weight due to ablation, lbm/sec
ZMAX, 200	1844-2043	Altitudes corresponding to maximums in α , in feet
ZMIN, 200	2044-2243	Altitudes correspondint to minimums in α , in feet
NOTE Unless otherwise designated, numbers in Common Location column refer to positions in the <u>PCCUR</u> array.		
BCB, 40	6181-6220	Array of lower corridor values for ballistic coefficients
BCD, 40	6101-6140	Array of lower corridor values for deceleration in g's
BCV, 40	6021-6060	Array of lower corridor values for velocity in ft/sec
BCWL1, 40	6501-6540	Array of lower corridor values for wake length in meters for first radar frequency
BCWL2, 40	6581-6620	Array of lower corridor values for wake length in meters for second radar frequency
BCWL3, 40	6661-6700	Array of lower corridor values for wake length in meters for third radar frequency
BCWR1, 40	6261-6300	Array of lower corridor values for wake cross section for first radar frequency, see IDBL for units
BCWR2, 40	6341-6380	Array of lower corridor values for wake cross section for second radar frequency, see IDBL for units

Output (cont'd)

Name	Common Location	Description
BCWR3, 40	6421-6460	Array of lower corridor values for wake cross section for third radar frequency
DB, 160	2081-2240	The array of difference between the R/V and decoy ballistic coefficient
DD, 160	1921-2080	The array of difference between the R/V and decoy deceleration in g's
DV, 160	1761-1920	The array of difference between the R/V and decoy velocity in ft/sec
DVAR, 50	11707-11756	Stored values of certain elements of the ØCCUR array
DWL1, 160	2721-2880	The array of difference between the R/V and decoy wake length in meters for first radar frequency
DWL2, 160	2881-3040	The array of difference between the R/V and decoy wake length in meters for second radar frequency
DWL3, 160	3041-3200	The array of difference between the R/V and decoy wake length in meters for third radar frequency
DWR1, 160	2241-2400	The array of difference between the R/V and decoy wake cross section for first radar frequency, see IDBL for units
DWR2, 160	2401-2560	The array of difference between the R/V and decoy wake cross section for second radar frequency, see IDBL for units
DWR3, 160	2561-2720	The array of difference between the R/V and decoy wake cross section for third radar frequency, see IDBL for units
HØLD, 150	7101-7250	A 3 by 50 array of stored design parameters
SAVE, 4320	7251-11570	A 9 by 160 by 3 array of stored differences in design parameters for determining influence coefficients.

Output (cont'd)

Name	Common Location	Description
SB, 40	6821-6860	See input
SD, 40	6781-6820	See input
SV, 40	6741-6780	See input
SWL1, 40	6981-7020	See input
SWL2, 40	7021-7060	See input
SWL3, 40	7061-7100	See input
SWR1, 40	6861-6900	See input
SWR2, 40	6901-6940	See input
SWR3, 40	6941-6980	See input
TCB, 40	6221-6260	See input
TCD, 40	6141-6180	See input
TCV, 40	6061-6100	See input
TCWL1, 40	6541-6580	See input
TCWL2, 40	6621-6740	See input
TCWL3, 40	6701-6740	See input
TCWR1, 40	6301-6340	See input
TCWR2, 40	6381-6420	See input
TCWR3, 40	6461-6500	See input
TT, 200	5561 - 5760	Combined altitude array
TTM, 200	TTMCOM	Combined time array
TS, 160	3201-3360	Array of saved R/V trajectory times

Output (cont'd)

Name	Common Location	Description
TPLØT, 160	1-160	Array of R/V or decoy trajectory times
WL1S, 160	4481-4640	Array of stored wake length values in meters at first radar frequency for reentry vehicle
WL2S, 160	4641-4800	Array of stored wake length values in meters at second radar frequency for reentry vehicle
WL3S, 160	4801-4960	Array of stored wake length values in meters at third radar frequency for reentry vehicle
WR1S, 160	4001-4160	Array of stored values for wake cross section in units specified by IDBL at first radar frequency for reentry vehicle
WR2S, 160	4161-4320	Array of stored values for wake cross section in units specified by IDBL at second radar frequency for reentry vehicle
WR3S, 160	4321-4480	Array of stored values for wake cross section in units specified by IDBL at third radar frequency for reentry vehicle
ZPLØT, 160	161-320	Array of altitudes for R/V or decoy trajectory
ZS, 160	3361-3520	Array of saved altitudes for R/V trajectory

NOTE The following quantities are integers:

II	IØCCUR(308)	Second subscript of the HØLD array
INALPH	NØCCUR(30)	Option to Input α
JJ	IØCCUR(310)	Second subscript of SAVE array
K	IØCCUR(313)	Number of altitudes in the combined altitude array, TT
LØPT	NØCCUR(07)	Trajectory option code
NCØN	IØCCUR(312)	Location in the ØCCUR array of design parameter being varied

Output (cont'd)

Name	Common Location	Description
NOTE The following quantities are integers (cont'd)		
NDDCOY	INFTST	Control code used in INTGRL
NGEOM	NØCCUR(15)	Input geometry option code
NLOW	--	A control code used in calling HEADER
NTHRST	NØCCUR(22)	Input thrusting option code

4. Numerical Methods

The following abbreviations will be used:

rvt means reentry vehicle trajectory.

bdt means basic decoy trajectory.

cdt means comparison decoy trajectory.

F123 saves three sets of tables for use by other subroutines. In the first set of tables the values of up to nine characteristics of rvt are saved for each altitude at which SUBROUTINE VIXEN outputs data. To obtain the second set of tables, the values of the corresponding characteristics for a bdt or a cdt is subtracted from that of the rvt for each characteristic at each altitude. The third table saves up to three sets of these differences in the SAVE array.

In the table below, the first column contains the name of the array in which the values of the characteristic for the rvt are saved. The second column contains the name of the array in which the differences are saved. The third column contains the name of the characteristic whose differences are being saved.

<u>rvt Values</u>	<u>Differences</u>	<u>Characteristic Name</u>
VS	DV	velocity
DS	DD	deceleration
BS	DB	ballistic coefficient
WL1S	DWL1	wake length at radar frequency one
WL2S	DWL2	wake length at radar frequency two
WL3S	DWL3	wake length at radar frequency three
WR1S	DWR1	cross section at radar frequency one
WR2S	DWR2	cross section at radar frequency two
WR3S	DWR3	cross section at radar frequency three

The two statements: - 100 CALL TIMERS(1, TIMTAB)
CALL TIMERS(-1, TIMTAB)

provide for the possibility of printing out the elapsed time between execution of the two statements. In the present deck this subroutine is a dummy producing no output.

After initializing NDC and NDVC, F123 calls SUBROUTINE SAVEDV to save the values of ϕ CCUR(133-145), ϕ CCUR(205-209) and ϕ CCUR(222) in DVAR(1-19) respectively. These values will be restored to the ϕ CCUR array later. Next IREF is tested. If IREF is not equal to two go to 497. The transfer to 497 means that the rvt is to be input or calculated when IREF is three or one respectively. If IREF equals two, NDECØY is tested. If NDECØY equals one go to 497. In this case, either a bdt or the rvt is to be processed. If NDECØY equals two, one cdt will be calculated for each design parameter being perturbed. If NDECØY equals three, two cdt's will be calculated for each design parameter being perturbed. If NDECØY is either two or three, the DØ loop ending at 111 is entered to save the values of the design parameters being perturbed. These are saved in the HØLD array. HØLD(1,i) saves the value for the basic decoy. HØLD(2,i) saves the value for the first comparison decoy and HØLD(3,i) saves the value for the second comparison decoy. NDVC is the number of design parameters to be perturbed in one case. The input arrays DVL and DVH contain the perturbed values for the first and second comparison decoys respectively.

At 497, the DØ loop ending at 777 is entered and IREF is tested. If IREF \neq 2 control passes to 498. Control passes to 401, 402 or 403 when NDECØY

is one, two and three respectively. At each of these three statements the values of $NLOW$ and $NHIGH$ are set and statement 404 is executed next. At 404, NCN is defined. This is the location in the OCCUR array of the I th parameter being perturbed. Next, the call to $SAVEDV$ restores the values saved in $DVAR$ to the OCCUR array. Then going to 499, $NLOW$ and $NHIGH$ are set for the rvt and 499 follows.

At 499, the $D\text{O}$ loop ending at 444 is entered to obtain and process trajectory data. If KK equals two, $\text{OCCUR}(NCN)$ is set equal to $DVL(II)$ for the first comparison decoy. If KK equals three, $\text{OCCUR}(NCN)$ is set equal to $DVH(II)$ for the second comparison decoy. Next, if $M\text{ODE}$ is not equal to one, the case number is increased by .001 to aid in identifying printed output and plots. Next, the case, date and memo number are printed and $IREF$ is tested again, and if = 3 passes to 28. The sequence following this test and ending at 56 tests the input options and may reset some of them. The next three set values for numerical constants. The following three set input options equal to storage quantities in order to carry them over to the following input case unless they are reset. The next eight statements change units from inches (input units) to feet (units used internally). The next ten statements set drag coefficient quantities to zero. Then $L\text{OPT}$ is tested. If $L\text{OPT}$ equals zero go to 201. If $L\text{OPT}$ is not equal to zero, the next nine statements set rotational quantities equal to zero. If $L\text{OPT} = 0$, the quantities related to maximums and minimums in α are zeroed. The next four find the trigonometric functions of the thrust offset angles. The next six change angles from degrees to radians. In the next thirteen, the mass loss and heating quantities are set equal to zero.

The call to TIMERS next prepares to measure the time taken to calculate trajectory. Then VIXEN is called to calculate the trajectory and TIMERS is called twice, the first time to print the time elapsing for the trajectory calculation and the second for the time elapsing between the execution of statement 100 and this call to TIMERS. The following six statements set the thrust quantities equal to zero. The next eight reconvert geometric quantities from feet to inches. Next the value of ALST is changed from degrees to radians and the next three statements set options equal to stored input quantities to prepare for the next case.

Now IREF is tested. If IREF equals two, go to 30. If IREF equals one, the data for the rvt is saved in the DØ loop ending at 22. Then LP is checked. If LP < 6 control passes to 407. If LP equals six, the rvt calculations have failed. A message to this effect is printed, SAVEDV is called to restore values from DVAR to the ØCCUR array, and control returns to the main program.

At 30, HEADER is called to indicate the type of decoy trajectory being calculated and print selected quantities associated with it. Then, in the DØ loop ending at 33, the differences described in the table above are saved. Next LP is tested as above; if equal to 6, a message is written, SAVEDV is called, and the program goes next to 777. If LP is less than 6, control passes to 507.

In the sequence of statements beginning at 507 and ending at 50, the altitudes in the input array H and the altitudes at which VIXEN produces output are combined into one array, monotonically decreasing. If the same altitude is found in both arrays, it is counted only once. The altitude values are put in

the TT array and the corresponding times are saved in the TTM array. The number of elements in each of these arrays is K. Next in the DØ loop ending at 333, the differences needed by the influence coefficients SUBRØUTINE INFCØF are saved. The nine statements between 333 and 444 provide for the calling of SUBRØUTINE INTGRL which may calculate corridor integrals and sigma square integrals as well as call the plotting routine, AVPLT, and the influence coefficients SUBRØUTINE INFCØF. After 777 SAVEDV is called to restore values to the ØCCUR array and control returns to the main program.

5. Other Information

- A. F123 is called by the main program.
- B. F123 calls
 - (1) SUBRØUTINE HEADER
 - (2) SUBRØUTINE INTGRL
 - (3) SUBRØUTINE SAVEDV
 - (4) SUBRØUTINE VIXEN
 - (5) The dummied SUBRØUTINE TIMERS
- C. F123 calls the IBM supplied finctions SIN and CØS.

4.1 Integration of Special Functions

SUBROUTINE INTGRL (KA, KB, LX, KD, KE, Q, R, S, SIG, DC)

1. Purpose

INTGRL is intended to perform one of four main tasks depending upon certain input options.

A. To evaluate a corridor integral and print a tabular array of data at each of a number of altitudes.

B. To evaluate a sigma square integral.

C. Call the plot routine to obtain a corridor plot.

D. Call the plot routine to obtain an influence coefficient plot.

2. Input

Name	Source	Common Block	Description
A, 27	SR2490 or ZREADX	PCCUR (11571-11597)	See text
B, 40	LINFIT	PCCUR (4961-5000)	See text
C, 40	LINFIT	PCCUR (5001-5040)	See text
D, 160	LINFIT	PCCUR (5041-5200)	See text
E, 160	LINFIT	PCCUR (5201-5360)	See text
F, 40	LINFIT	PCCUR (5361-5400)	See text
G, 40	LINFIT	PCCUR (5401-5440)	See text
H, 40	SR2490 or ZREADX	PCCUR (5841-5880)	Altitudes
IOP, 90	ZREADX	IOPCUR (1-90)	Option c
KA	F123		Used to array n
KB	F123		Used to array n
KD	F123		See text
KE	F123		See text
KX	F123	IOPCUR (313)	The num altitudes
LPLOT	VIXEN	IOPCUR (302)	The num altitudes
LX	F123		Flag sp array a integral
NCP	SR2490 or ZREADX	IOPCUR (304)	Number
NDEC0Y	F123	INFTST	Test pa
OPCUR, 4000	Unlabeled Common	OPCUR (1-4000)	

A

lock	Description	Preset
571-11597)	See text	A(I) = 1 If I=1 (MOD3) otherwise A(I)=0
61-5000)	See text	
01-5040)	See text	
41-5200)	See text	
01-5360)	See text	
61-5400)	See text	
01-5440)	See text	
41-5880)	Altitudes at which corridor limits are input	0
-90)	Option code	1
	Used to determine which elements of the IOP array need to be tested	
	Used to determine which elements of the IOP array need to be tested	
	See text	
	See text	
13)	The number of elements in the T array of altitudes	
02)	The number of elements in the Z array of altitudes	
	Flag specifying which three elements of the A array are to be used in evaluating the corridor integral	
04)	Number of elements in the H array	1
	Test parameter for influence coefficient plots	1
4000)		

B

2. Input (Concl'd)

Name	Source	Common Block	
Q, 40	SR2490 or ZREADX		Array o
R, 160	SR2490 or F123		See Tex
S, 40	SR2490 or ZREADX		Array o
SIG, 40	SR2490		Standar
T, 200	SR2490 or ZREADX	PCCUR (5561 - 5760)	See Tex
TPLØT, 160	VIXEN	PCCUR (1-160)	Array o
Z, 160	VIXEN	PCCUR (161-320)	of altitu
			Array o
			data

NOTE

DC is a special input of alphanumeric data

A

Block	Description	Preset
	Array of lower corridor limits	0
	See Text	0
	Array of upper corridor limits	0
	Standard deviation in error of measurement	
561 - 5760)	See Text	0
-160)	Array of times corresponding to the Z array of altitudes	
61-320)	Array of altitudes for which VIXEN stores trajectory data	

3. Output

Name	Common Block	Description	Preset
A, 27	See Input	See ZREADX for AA	See input
B, 40	See Input	See LINFIT description	
C, 40	See Input	See LINFIT description	
D, 160	See Input	See LINFIT description	
E, 160	See Input	See LINFIT description	
F, 40	See Input	See LINFIT description	
G, 40	See Input	See LINFIT description	
HDIF	PCCUR (11758)	H(1) - H(NCP)	0
OCCUR, 4000	unlabelled common	Some values are changed for use by the optimizer	0
Q1		Lower corridor limit at initial altitude	
Q2		Lower corridor limit at any other altitude.	
R1		Value of R at initial altitude	
R2		Value of R at any other altitude	
S1		Upper corridor limit at initial altitude	
S2		Upper corridor limit at any other altitude	
T, 200	See Input	See text	0
T1		Used by ADD, see Text	
T2		Used by ADD, see Text	
XX, 200		See Text	
Y, 200		See Text	
ZZ, 200		See Text	

4. Numerical Procedures

Before the logical flow in INTGRL is discussed, it will be useful to describe in some detail what we call the corridor integral. The input array H contains a set of altitudes at each of which an upper corridor limit and a lower corridor limit has been input for one or more of nine different arrays. The appropriate set of upper and lower corridor limits are sent to INTGRL through the list in the arrays S and R respectively. The number of elements in each of these arrays is NCP, also an input quantity. The array Z contains the altitudes at which VIXEN outputs trajectory data. The array R contains the difference between the value of some specified characteristics of the reentry vehicle and the value of the same characteristic of a decoy at each of these altitudes. The number of elements in the R and Z arrays is LPLØT, which is defined in VIXEN.

SUBROUTINE LINFIT is used to fit straight line segments between the pairs of points $(H(I), Q(I))$ and $(H(I+1), Q(I+1))$ $I = 1, 2, \dots, NCP - 1$. The coefficients of the fit are saved in the arrays B and C. Similarly, the points (H, S) are fitted and the coefficients saved in the arrays F and G, and the points (Z, R) are fitted and the coefficients saved in the arrays D and E. We can then express the values of Q, R, and S at any altitude ALT by

$$Q = ALT * B(I) + C(I)$$

$$R = ALT * D(I) + E(I)$$

$$S = ALT * F(I) + G(I)$$

The area between the segmented lines formed by Q and R is called the corridor. So long as the value of R is greater than or equal to the value of Q and less than or equal to the value of S, no integration is done.

If R is less than Q on an interval between ALT1 and ALT2 the function $F(Z) = A(LX)*Z^2 + A(LX + 1)*Z + A(LX + 2)$ is integrated over the area defined by lines R, Q, ALT1 and ALT2 with respect to Z and the absolute value of the result is added to DA, the running sum of the integral. If R is greater than S on some interval between the altitudes ALT1 and ALT2, the same function is integrated over the region defined by the lines S, R, ALT1 and ALT2, these two being specific values of Z. The function subprogram ADD is used to perform this integration.

The subroutine flow proceeds as follows: Immediately upon entering INTGRL, IOP(KA + 2*KB) is tested. If IOP(KA + 2*KB) equals 0, no corridor integral calculations are desired and statement 1120 is executed next. If it equals 1, KQ is defined and tested in the next two statements. IF KQ = 1, the heading for the tabular output associated with the corridor integral is printed. The next two statements define JGØ. IF JGØ is 0, the slope print out for the corridor integral will be omitted. LINFIT is then called three times to save the fitting coefficients described above. The next four statements initialize the test parameters NX and ITEST, the accumulator for the corridor integral DA, and the altitude at which R first leaves the corridor DB. Q1, R1, and S1, calculated next, are the respective values of Q, R, and S at the initial altitude Z(1). These values are then saved as XX(1), YY(1), and ZZ(1) for the corridor plots. Next the value of NX is determined. NX will be 0, 1, or 2 depending on whether R1 is less than Q1, between Q1 and S1, or above S1. If Q1 is already outside the corridor, the next two statements will change ITEST to 1 and set DB = Z(1). Then if KQ = 0, statement 40 will be executed next. If KQ = 1, NX is tested and the WRITE statement at 10, 30, or 20 will be executed to write the first line of the corridor integral table as NX is 0, 1, or 2 respectively. Then statement 40 is executed.

Following the statement 40, NS and NR are defined and the $D\emptyset$ loop ending at 1111 is entered to calculate the corridor integral.

The array T, used here, was generated in F123 by combining the H and Z arrays. If an element is common to both arrays, it is included in the T array only once. The T, H, and Z arrays are monotonically decreasing. It will usually happen that for some J, $T(M)$ will lie between $H(J)$ and $H(J+1)$. If so, the value of J is found and saved in the $D\emptyset$ loop ending at 44. If not, J is set equal to NR. Similarly, it will usually happen that $T(M)$ lies between $Z(I)$ and $Z(I+1)$ for some I. If so, the value of I is found and saved in the $D\emptyset$ loop ending at 55. Otherwise I is set = NS.

The six statements starting at 60 set $Q2$, $R2$, and $S2$ equal to the values of Q, R, and S respectively at the altitude $T(M)$ and then save these values in $XX(M)$, $YY(M)$ and $ZZ(M)$ respectively. The next three statements find the value of NY which will be 1, 4, or 7 accordingly as $R2$ is less than $Q2$, between $Q2$ and $S2$, or greater than $S2$. NX and NY now describe the behavior of the R line between the altitudes $T(M-1)$ and $T(M)$. $IG\emptyset$ is set = $NX + NY$ and the computed $G\emptyset T\emptyset$ is executed, sending control to the statement having the statement number = $100*IG\emptyset$. This transfer will result in the execution of some or all of the following six types of statements.

A. $T2$, if not = 0, is defined as the altitude at which R crosses either S or Q. It will then be either the higher or the lower of the two altitudes used to define the region of integration.

B. $T1 = T(M-1)$.

$T1$, when used is the higher altitude defining a region of integration.

C. $DA = DA + ADD$

The function subprogram ADD performs the integration and adds the result to DA.

D. CALL WRITEM

If $KQ = 1$, a line of output is printed in the corridor integral table for the altitude $T(M)$.

E. IF ITEST = 0, GØ TØ 950

If ITEST = 0, statement 950 is executed next. If not, the following statement is executed.

F. GØ TØ 999

Statement 999 is executed next.

A description of the region of integration at each of the statements 100, 200, ... 900 follows. The following notation is used in the description. The general form is at XXX L1, L2, L3, L4, L5, where XXX is the statement number, L1 is the lower line of the region of integration which may be either a triangle or a trapezoid; it is always either R or S. L2 is the upper line for the region; it is always Q or R. L3 is the higher of the two altitudes defining the region and L4 is the lower altitude. L3 and L4 are parallel. L5 may be any of the four RQB, RSA, RQI and RSI meaning respectively R crosses Q from below lower corridor, R crosses S from above upper corridor, R crosses Q from inside corridor, and R crosses S from inside corridor. If there is no crossing, L5 is omitted.

At 100 the lines R, Q, T1, T2 define the region of integration.

At 200, R, Q, T2, T(M), RQ1 define the region of integration.

At 300, two regions are needed since R goes from above S to below Q. These regions of integration are described respectively by the lines S, R, T1, T2, RSA and by the lines R, Q, T2, T(M), RQ1.

At 400, R, Q, T1, T2, RQB describe the region of integration.

At 500, no integration is required. R is between Q and S for the entire interval.

At 600, S, R, T1, T2, RSA describe the region of integration.

At 700, two regions are required since R goes from below Q to above S. These regions of integration are described respectively by the lines R, Q, T1, T2, RQB and by the lines S, R, T2, T(M), RSI.

At 800, S, R, T2, T(M), RSI define the region of integration.

At 900, S, R, T1, T(M), RSI define the region of integration.

At statement 950, ITEST is changed to 1, then DB is set = T2 and statement 999 is executed.

At 999, NX is redefined to specify the relation of the beginning of the next R line segment to the corridor.

After statement 1111 has been executed KX-1 times, DA and DB are put into the OCCUR array for possible use by the optimizer. Next, if KQ = 0, the values of DA and DB are printed along with the name of the variable in the R array.

Next, $I\Phi P(KA + KB)$ is tested. If it equals 0, statement 2300 is executed. If it equals 1, we compute the sigma square integral. We use the trapezoidal rule to integrate the function $(R/SIG)^2$ from $H(1)$ to $H(I+1)$ $I = 1, 2, \dots, NCP-1$ with respect to H. The running sum is kept in the variable SUM. The values of SIG are input in the SRS array. As a preliminary $RL\Phi$ is set = $[R(1)/SIG(1)]^2$ and SUM is set = 0. The values of R at the altitudes $H(2), H(3), \dots, H(NCP)$ are found by linear interpolation in

the DØ loop ending at 2222. SUBROUTINE INTERP finds these values. RHI is defined as $(R(I)/SIG(I))^2$. The value of the integral from H(I-1) to H(I) is computed and added to SUM. RLØ is then set equal to RHI and the loop executed again. SUM is then put into the OCCUR array for possible use by the optimizer. Next IØP(KA) is tested. If it equals 0, statement 4000 is executed. If it equals 1, AVPLT is called to produce a corridor plot.

At 4000, if NDECOY is less than 3, control returns to F123. If it equals 3, IØP(KA + 5*KB) is tested. If it equals 1, AVPLT is called to produce an influence coefficient plot and control returns to F123. If it is equals 0 control returns to F123 immediately.

5. Other Information

- A. INTGRL is called by F123 only
- B. INTGRL calls
 - 1. SUBROUTINE LINFIT
 - 2. SUBROUTINE WRITEM
 - 3. SUBROUTINE INTERP
 - 4. SUBROUTINE AVPLT
 - 5. FUNCTION ADD

FUNCTION ADD (L,R,S,A,B,C,D,E)

1. Purpose

Add evaluates $\int_R^S F(t)dt$ over a trapezoidal region in the t, y plane defined as the area enclosed by the four lines:

1. $t = R$
2. $t = S$
3. $y = A*t + B$
4. $y = C*t + D$

$$F(t) = E_L t^2 + E_{L+1} t + E_{L+2}$$

2. Input

<u>Name</u>	<u>Source</u>	<u>Description</u>
A	INTGRL	curve fit coefficient
B	INTGRL	curve fit coefficient
C	INTGRL	curve fit coefficient
D	INTGRL	curve fit coefficient
E	INTGRL	AA, weighting coefficients for corridor integrals, 11571-11597
L	INTGRL	integer indicating which AA's are to be used
R	INTGRL	altitudes in feet
S	INTGRL	altitudes in feet

The E array is input under the name AA. It reached SUBROUTINE INTGRL through the PCCUR common array and INTGRL then sends it to ADD through the argument list.

E(3), E(6), E(9), E(12), E(15), E(18), E(21), E(24), and E(27) are preset to 1.0 and the rest of the E array is preset to zero in subroutine SR2490. R and S are altitudes in feet.

3. Output

<u>Name</u>	<u>Description</u>
ADD	corridor integral increment

4. Numerical Procedure

Since the height of the trapezoid is $(C-A)*t + (D-B)$, we may write

$$ADD = \int_R^S F(t) * ((C-A)*t + (D-B)) * dt.$$

This can be integrated explicitly as a function of t to give the integral I.

$$I = (C-A) \left(E_L \frac{t^4}{4} + E_{L+1} \frac{t^3}{3} + E_{L+2} \frac{t^2}{2} \right) + (D-B) \left(E_L \frac{t^3}{3} + E_{L+1} \frac{t^2}{2} + E_{L+2} t \right)$$

The value of I at $t = R$ is subtracted from its value at $t = S$ and the result is placed in location ADD.

5. Other Information

FUNCTION ADD is called by SUBROUTINE INTEGRAL only.
FUNCTION ADD does not call or reference any subprogram.

SUBROUTINE INFCOF (DC, IZ, ZQ)

1. Purpose

SUBROUTINE INFCOF calculates a table of influence coefficients and possibly second derivatives, at each altitude for which VIXEN stores trajectory information

2. Input

*indicates integer quantity

Name	Common Block	Source	Description
DC, 2	-	INTGRL	name of delta
H, 150	PCCUR(7101-7250)	F123	holds design variable values
II	I \emptyset CCUR(308)*	F123	second subscript of H
IZ, 19	-	INTGRL	locations, in \emptyset CCUR array, of 19 special parameters
JJ	I \emptyset CCUR(310) *	F123	first subscript of SAVE
LPL \emptyset T	I \emptyset CCUR(302) *	VIXEN	number of output altitudes
NC \emptyset N	I \emptyset CCUR(312) *	F123	location in the \emptyset CCUR array of design parameter being varied
NDEC \emptyset Y	I \emptyset CCUR(305) *	SR2490, ZREADX	test parameter
S, 4320	PCCUR(7251- 11570)	F123	array of saved deltas
T, 160	PCCUR(1-160)	VIXEN	time array
XQ, 40	-	INTGRL	array containing design parameter names
Z, 160	PCCUR(161-320)	VIXEN	altitude array

3. Output

SUBROUTINE INFCOF produces only printed output.

4. Numerical Procedure

The following definitions help to explain the meaning of the influence coefficients and the second derivative and the method of calculating them. The three deltas which may be printed out at each altitude are

First delta equals $S(JJ, I, 1)$

Second delta equals $S(JJ, I, 2)$

Third delta equals $S(JJ, I, 3)$

The array called S here is called SAVE in F123 which produces it. The first subscript, JJ, specifies the appropriate of the nine possible vehicle characteristics for which deltas have been saved. The second subscript, I, indicates the altitude at which the deltas were computed. The third subscript is 1, 2, or 3, when the decoy being compared with the reentry vehicle is respectively a basic decoy, a first comparison decoy and a second comparison decoy.

First influence coefficient equals

$$\frac{S(JJ, I, 2) - S(JJ, I, 1)}{H(2, II) - H(1, II)}$$

Second influence coefficient equals

$$\frac{S(JJ, I, 3) - S(JJ, I, 1)}{H(3, II) - H(1, II)}$$

The array called H here is called HOLD in F123 where it is generated. The subscript II comes from F123 and specifies which design parameter is being perturbed. The first subscript of H has the same meaning as the third subscript of S above.

Second derivative equals (second influence coefficient minus first influence coefficient) / (H(3, II) minus H(2, II))

The flow proceeds as follows: The DØ loop ending at 17 tests to determine if the design parameter being perturbed is one of the nineteen for which names have been stored in the XQ array. If so, JP specifies which of these names will be printed as part of the table identification. Otherwise, JP is set equal to zero and the value of the integer NCON is printed instead. Next X21, the difference in H, is defined to be used in calculating the first influence coefficient. Then, if NDECØY equals two, statement 100 is executed next. If NDECØY equals 3, X32, the difference in H, is defined to be used for calculating the second influence coefficient and X31 is defined to be used in calculating the second derivative.

At 100 AMID is defined as the arithmetic average of H(1, II) and H(2, II) for print out. Some preliminary information is printed, then a table heading having the five columns: ALTITUDE, TIME, FIRST DELTA, SECOND DELTA, INFLUENCE COEF is written. In the DØ loop ending at 222 one influence coefficient is calculated at each altitude and a line of five numbers is printed out at each altitude. The control returns to INTGRL.

At 500, the procedure is quite similar. The three arithmetic averages AMID, BMID and CMID are computed for preliminary print out. The table heading this time has eight columns: ALTITUDE, TIME, FIRST DELTA, SECOND DELTA, THIRD DELTA, FIRST INFLU COEF, SECOND INFLU COEF, SECOND DERIV.

In the DØ loop ending at 555, two influence coefficients and a second derivative are computed at each altitude and the eight specified values are printed under the appropriate headings. Then control returns to INTGRL.

5. Other Information

A. INFCØF is called by INTGRL only.

B. INFCØF does not call or reference any other subprogram.

SUBROUTINE INTERP (M, N, W, X, Y, Z)

1. Purpose

Given the two tables X and Y(X) and, in addition, a particular value of X, called W, SUBROUTINE INTERP finds Z equal to Y(W) by linear interpolation.

2. Input

<u>Name</u>	<u>Source</u>	<u>Description</u>
M	WAKE	dimension of X and Y arrays in WAKE
N	WAKE	size of X and Y tables
W	WAKE	particular value of X
X	WAKE	table of independent variables
Y	WAKE	table of dependent variables

3. Output

<u>Name</u>	<u>Description</u>
Z	value of Y(W) found by linear interpolation

4. Numerical Procedure:

First N is tested to determine if it is greater than zero and less than or equal to M. If it is, statement 20 is executed. If it is not, an error message is printed and SUBROUTINE EXIT is called to terminate the computer run.

At statement 20 $X(1)$ is compared with $X(2)$ to determine whether the X array is increasing or decreasing. The X array must be either monotonically increasing or monotonically decreasing. If it is not, results will be unreliable. If $X(1)$ is less than $X(2)$ control is transferred to statement 50 and the value of I, such that W is between $X(I-1)$ and $X(I)$, is found. If W is outside the range of the X table, I will be either 2 or N, accordingly as W is less than $X(1)$ or greater than $X(N)$ respectively. The value of Z is then computed at statement 40. If $X(1)$ is greater than $X(2)$, the value of I is found in the DØ loop ending at 22 and Z is again calculated at statement 40.

5. Other Information

- A. SUBROUTINE INTERP is called by SUBROUTINE WAKE.
- B. SUBROUTINE INTERP calls the IBM SUBROUTINE EXIT.

SUBROUTINE LINFIT (ND, NENTRY, X, Y, A, B)

1. Purpose

Given the NENTRY points (X, Y), LINFIT finds the equation of the line passing through the points (X, Y), (X_i, Y_i), in the form $y = A_i x + B_i$, for $i = 1, 2, \dots, (NENTRY-1)$. These coefficients can then be used by the calling program for making a linear fit when the interval required is known in advance.

2. Input

<u>Name</u>	<u>Source</u>	<u>Description</u>
ND	F123 and/or INTGRL	adjustable dimension
NENTRY	F123 and/or INTGRL	number of elements in X, Y, A and B
X, ND	F123 and/or INTGRL	array of independent variables
Y, ND	F123 and/or INTGRL	array of dependent variables

3. Output

<u>Name</u>	<u>Symbol</u>	<u>Description</u>
A, ND	A _i	coefficient of X in linear fit
B, ND	B _i	constant in linear fit

4. Numerical Procedure

$$A_i = (y_{i+1} - y_i) / (x_{i+1} - x_i) \quad \text{and} \quad B_i = y_i - A_i x_i \quad \text{are the equations}$$

evaluated by SUBROUTINE LINFIT.

5. Other Information

A. LINFIT is called by SUBROUTINE INTGRL and by SUBROUTINE F123.

B. It does not call or reference any other subprogram.

SUBROUTINE WRITEM(IX, Z, T, Q, R, S, DA, DB, SLOPE, N)

1. Purpose

Each time SUBROUTINE WRITEM is called, a single line may be printed in the corridor integral table.

2. Input

The source of all input quantities is SUBROUTINE INTGRL

*indicates integer quantity

<u>Name</u>	<u>Description</u>
DA	cumulative value of the integral
DB	altitude at which delta enters or leaves corridor
IX*	test parameter for the write statement to be used
N*	control code; if N equals zero, print out is deleted
Q	lower corridor value at altitude Z
R	value of delta at altitude Z
S	upper corridor value at altitude Z
SLOPE	slope of delta on latest line segment
T	time at altitude Z
Z	altitude

3. Output

Only printed output is produced as described in Section 4.

4. Numerical Methods

First N is tested. If N equals zero, no printing is done and control returns to INTGRL. Otherwise, IX is divided by ten so that it can be used as the index of a computed go to statement.

Control is then transferred to 10, 20, 30, 40, 50, or 60 when IX is 1, 2, 3, 4, 5, or 6 respectively. At each of the six write statements at 10, 20, 30, 40, 50, and 60, the first six columns of the integral table are filled. These contain the values of altitude, time, lower corridor, delta, upper corridor and integral value. At 20, the leave corridor column is also filled. At 30, the enter corridor column is filled. At 40, the slope column is filled. At 50, both the leave corridor and slope columns are filled. At 60, both the enter corridor and slope columns are filled. After the line has been printed, control returns to INTGRL.

5. Other Information

- A. SUBROUTINE WRITEM is called by SUBROUTINE INTGRL only.
- B. SUBROUTINE WRITEM calls no other subprograms.

4.2 Storage of Geometric Parameters

SUBROUTINE SAVEDV(K, DVAR, OCCUR)

1. Purpose

SAVEDV stores the values of OCCUR(133-145), OCCUR(205-209) and OCCUR(222) in DVAR(1-19) respectively and later restores these values to the OCCUR array.

2. Input

*indicates integer quantity

<u>Name</u>	<u>Source of Input</u>	<u>Description</u>
DVAR, 50	F123	array to save selected OCCUR values
K *	F123	test parameter
OCCUR, 4000	F123	common array

3. Output

<u>Name</u>	<u>Description</u>
DVAR, 50	See Input
OCCUR, 4000	See Input

4. Numerical Procedures

If the input test parameter K is one, statement 100 is executed. If K is two, statement 200 is executed.

A 100 the values of OCCUR(133-145), OCCUR(205-209) and OCCUR(222) are saved as DVAR(1-19) respectively and control returns to F123.

At statement 200 the values stored in the DVAR array at statement 100 are restored to the appropriate locations in the OCCUR array and control returns to F123.

5. Other Information

A. SAVEDV is called by F123 only.

B. SAVEDV does not call or reference any other subprograms.

4.3 Miscellaneous Printout Operations

SUBROUTINE HEADER(K)

1. Purpose

HEADER prints the type of vehicle whose trajectory is being processed along with the values of nineteen design parameters, the value of NGEØM, an input option, and the value of LP an error test parameter .

2. Input

Name	Source	Common Block	Description
K	F123		test parameter
NØCCUR, 30	READIT	NØCCUR	input options
ØCCUR, 4000		ØCCUR	common array

3. Output

SUBROUTINE HEADER produces printed output only.

4. Numerical Procedure

Immediately upon entering HEADER, K is tested. The statements 10, 20 and 30 are executed next when K is 1, 2, and 3 respectively.

At 10 the heading "BASIC DECOY TRAJECTORY" is printed and control is transferred to 40.

At 20 the heading "COMPARISON DECOY CHARACTERISTICS" is printed and control is transferred to 40.

At 30 the heading "REFERENCE REENTRY VEHICLE CHARACTER-
ISTICS" is printed and statement 40 follows.

At 40 the values of nineteen design parameters, the option NGEØM
and the test parameter LP are printed with appropriate headings. Then
control returns to F123.

5. Other Information

- A. HEADER is called by F123 only.
- B. HEADER does not call or reference any subprograms.

SUBROUTINE TIMERS

1. Purpose

SUBROUTINE TIMERS is a dummy subroutine.

2. Input

None

3. Output

None

4. Numerical Procedure

The original TIMERS has the facility to print the time elapsing between the execution of any two specified statements in a subroutine. It has been replaced by the dummy because this print out is not desired at present.

5. Other Information

A. SUBROUTINE TIMERS is called by F123 only.

B. SUBROUTINE TIMES does not call or reference any other subprogram.

5.0 EFFECTIVENESS OPERATIONS

Effectiveness Operations

The implementation of the decoy effectiveness model technology is contained in SUBROUTINE EFFECT.

SUBROUTINE EFFECT

1. Purpose

SUBROUTINE EFFECT calculates quantity PD which is the probability that a decoy will be discriminated. SUBROUTINE EFFECT receives from input a quantity PFD (the probability of false dismissal of a reentry vehicle and calculates a quantity SIGMA. Next T is found such that

$$\text{PFD} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^T e^{-\frac{x^2}{2}} dx$$

This is done by a Newton-Raphson iteration scheme. Finally PD is calculated as

$$\text{PD} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{T + \text{SIGMA}} e^{-\frac{x^2}{2}} dx$$

2. Input

Name	Source	Common Block	Preset	Description
IØP, 90	SR2490, ZREADX	IOCCUR(1-90)	1	option parameter
HDIF	INTGRL	PCCUR(11758)	0	total length of corridor
PFD	SR2490, ZREADX	PCCUR(11757)	0	probability of false dismissal
SRS, 9	SR2490, ZREADX	PCCUR(11598-11606)	0	number of smooth radar samples

3. Output

Name	Common Block	Description
PD	ØCCUR (3962)	probability of discrimination
SIGMA	ØCCUR (3965)	The sum of nine weighted squares divided by HDIF. SIGMA equals the square root of the quotient

4. Numerical Procedure

The DØ loop ending at 11 accumulates the sum of nine weighted squares. Then the two statements following 11 divide this sum by HDIF and set SIGMA equal to the square root of the quotient. TØLD is set equal to -2 as the initial guess for T. Y is the corresponding value of the integral at this value of T. K is an iteration counter. The IBM function subprogram

$$\text{DERF}(T) = \frac{2}{\sqrt{\pi}} \int_0^T e^{-x^2} dx$$

determines the error function used in the integration. If we substitute $X/\sqrt{2}$ for X then the desired integral can be expressed as

$$\text{INT} = \frac{1}{2} \cdot (1 + \text{DERF}(\frac{T}{\sqrt{2}}))$$

The iteration starts at statement 10 where K is increased by 1. If K reaches 20 the program goes to 99 where PD is set equal to one, an error message is written and control returns to FEV.

If convergence is reached, the program goes to 99 where the value of the integral of (T + SIGMA) is calculated and control returns to FEV.

If K is less than 20 and convergence has not been achieved, TØLD and Y are reset and the program goes to 10 to increase K and try again.

5. Other Information

- A. SUBROUTINE EFFECT is called by SUBROUTINE FEV.
- B. SUBROUTINE EFFECT calls the IBM routines DABS and DERF.

6.0 CLASSIC FUNCTIONS FOR TESTING OPTIMIZATION
TECHNIQUES

Classic Functions

The following section contains the description of SUB-ROUTINE CLASSC, which contains classic functions to be used in testing the optimization procedures.

SUBROUTINE CLASSC (F321)

1. Purpose

SUBROUTINE CLASSC contains classic functions which are designed to provide difficult tests of how well a minimization routine works.

2. Input

*indicates integer quantity

<u>Name</u>	<u>Common Location</u>	<u>Source of Input</u>	<u>Description</u>
A, 514	ØCCUR(301 - 814)	READIT	curve fit coefficients which are read in
IØP	NØCCUR(3)*	READIT	code which determines the function to be minimized; input as MØPT
X, 4000	ØCCUR (1-4000)	FEV	the ØCCUR locations defined in FEV

3. Output

LP	ØCCUR(4000) *		error control code
X, 4000	ØCCUR(1 - 4000)		the output ØCCUR locations correspond to the function value

4. Numerical Procedure

SUBROUTINE CLASSC begins by testing the control code IØP. If IØP equals 3, 4, or 5 control passes to statements 300, 401, or 501 respectively. If IØP has none of these values it evaluates the equation

$$F = 100. (X_2 - X_1^2)^2 + (1. - X_1)^2 + A_1$$

then sets X_{100} equal to F. If the absolute value of IØP is 1, control then passes to statement 999. If not, the following equations are evaluated before control passes to 999.

$$B = A_{11} + (X_1 - A_7) \{ A_{10} + (X_1 - A_7) [A_9 + (X_1 - A_7) A_8] \}$$

$$C = A_{16} + (X_1 - A_{12}) \{ A_{15} + (X_1 - A_{12}) [A_{14} + (X_1 - A_{12}) A_{13}] \}$$

$$X_{101} = A_2(X_2 - B)$$

$$X_{102} = A_3(X_2 - C)$$

Statement 300 begins the calculations for another function with the zeroing of the quantities FA, FB, FC, and FD. The quantities FA, FA2, FA3, and FA4 are then evaluated from the equations below and their values assigned respectively to X_{200} , X_{201} , X_{202} , and X_{203} .

$$FA = X_1^2 + X_2^2 + 2. X_3^2 + X_4^2 - 5. X_1 - 5. X_2 - 21. X_3 + 7. X_4 + A_{17}$$

$$FA2 = 8. + X_4 - X_3 + X_2 - X_1 - X_4^2 - X_3^2 - X_2^2 - X_1^2$$

$$FA3 = 10. + X_4 + X_1 - 2. X_4^2 - X_3^2 - 2. X_2^2 - X_1^2$$

$$FA4 = 5. + X_4 + X_2 - 2. X_1 - 2. X_1^2 - X_2^2 - X_3^2$$

Control then passes to statement 999.

Statement 401 evaluates using the equation below the fourth function whose value is then assigned to X_{300} , before control passes to 999.

$$F = 4. X_1 + X_2 + A_{25} (1./X_1 + 1./X_2) + A_{26}$$

The fifth function is evaluated by statement 501 and its value assigned to X_{400} before statement 999 is reached.

$$F = A_1 X_1^2 + A_2 X_1 X_2 + A_3 X_2^2 + A_4 X_1 + A_5 X_2 + A_6.$$

Statement 999 sets the dummy variable F321 to zero, then LP is set equal to 1 before the return to FEV is executed.

5. Other Information

- A. SUBROUTINE CLASSC is called by SUBROUTINE FEV.
- B. SUBROUTINE CLASSC calls on the IBM supplied routines IABS and FDXPI (exponentiation).

7.0 PLOTTER INTERFACE SUBROUTINES

Plotting

The subroutines utilizing in plotting, AVPLT, MAXMIN, and PLT are described in the following sections.

SUBROUTINE AVPLT (IZ, XQ, X, Y, Z, KY, IC)

1. Purpose

AVPLT can produce any of four kinds of plots, depending on the value of KY. If $KY = 1$, influence coefficient plots will be prepared at a specified number of altitudes, NPA. NPA may be a maximum of 160. Three points are plotted on each graph. These points are not connected by straight lines. If $KY = 2$, one corridor plot is produced. Three curves are plotted on the same graph. The points of each curve are connected by straight lines. If $KY = 3$, one plot of Y vs. X is produced. The points are connected by straight lines. If $KY = 4$, one plot of $\log_{10} Y$ vs. X will be produced. The points are connected by straight lines. If $KY = 5$, the plot file will be closed and the computer run will be terminated.

2. Input

*indicates integer quantity

Name	Source of Input	Common Block	Preset
HOLD, 150	F123	PCCUR (7101-7250)	0
IC *	Calling Program	-	-
II *	F123	IOCCUR (308)	-
IMPLØT *	SR2490	IOCCUR (309)	0
IØP, 90 *	SR2490 or ZREADX	IOCCUR (1-90)	1
IZ, 19 *	Calling Program	-	-
JJ *	F123	IOCCUR (310)	-
KX *	F123	IOCCUR (313)	-
KY *	Calling Program	-	-
LPLØT *	VIXEN	IOCCUR (302)	-
NCØN *	F123	IOCCUR (312)	-
NDECØY *	SR2490 or ZREADX	IOCCUR (305)	1
NPA *	SR2490 or ZREADX	IOCCUR (307)	1
NPV, 160 *	SR2490 or ZREADX	IOCCUR (141-300)	1
SAVE, 4320	F123	PCCUR (7251-11570)	0
T, 200	F123	PCCUR (5561-5760)	0
X	Calling Program	-	-
XQ, 40	Calling Program	-	-
Y	Calling Program	-	-
Z	Calling Program	-	-
ZPLØT, 160	VIXEN	PCCUR (161-320)	0

A

Block	Preset Value	Description
-7250)	0	design variable being perturbed
)	-	If KY = 3 or 4, test parameter for log plots. If KY = 1 or 2, contains the vertical axis title.
)	-	Second subscript in HOLD matrix
)	0	Test parameter for opening plot file
0)	1	Option parameter
)	-	Array containing plot title information
)	-	First subscript of the variable SAVE
)	-	Number of points for each curve on corridor plot
)	-	Test parameter for kind of plot
)	-	Number of altitudes at which information is output from VIXEN.
)	-	Location in OCCUR array of design parameter being perturbed.
)	1	Test parameter for corridor plot title
)	1	The number of altitudes at which influence coefficient plots are produced.
-300)	1	Array containing indices of altitudes at which influence coefficients flows will be produced.
-11570)	0	Dependent variable for influence coefficient plots.
-5760)	0	Expanded array of altitudes
)	-	Dependent variable for all plots except influence coefficient plots
)	-	Array containing horizontal axis
)	-	Either second dependent variable for corridor plots or dependent variable for third or fourth type of plots
)	-	Third dependent variable for corridor plots
320)	0	Array of altitudes at which VIXEN outputs information

B

BLANK PAGE

3. Output

Name	Common Block	Preset Value	Description
IMPLOT*	IOCCUR (309)	0	Test parameter for opening plot file

4. Numerical Methods

Immediately upon entering AVPLT, the parameter IMPLOT is tested. If IMPLOT = 0, IDFRMV is called to open the plot file. IMPLOT is then changed to 1 and statement 10 is executed. If IMPLOT = 1, statement 10 is executed next.

At statement 10, KY is tested. If KY = 1 or 2, statement 20 is executed. If KY = 3 or 4, control passes to statement 110. If KY = 5, control passes to statement 230 where the plot file will be closed and the computer run terminated. After statement 20, the vertical axis plot title is defined.

Next, in the DØ loop ending at statement 33, the program starts to compare each element of the IZ array with NCON. If any IZ = NCON, statement 40 is executed next, the required subscript of IZ having been saved as JP. If none of the IZ's = NCON, JP is set to zero, and statement 45 is executed.

At statement 40, JS and Jr are defined. These specify which two elements of the XQ array will be used for the horizontal axis title of an influence coefficient plot or of a corridor plot if a design variable is being perturbed. Statement 45 is executed next.

At statement 45, KY is tested. If KY = 2, statement 70 is executed next. If KY = 1, the three statements following statement 45 define the W array of independent variables which then remain constant for each of the NPA influence coefficient plots to be set up in the DØ loop ending at statement 66. The W's are the three values of the design variable being perturbed. W(1) is the smaller of the two values for the comparison decoys. W(3) is the largest of these two values. W(2) is the value for the basic decoy. The HØLD array values are set in F123 in the DØ loop ending at statement 111. The values in the SAVE array were computed in F123 in the DØ loop ending at statement 33 and then saved in F123 in the DØ loop ending at statement 333. The J defined at the beginning of the DØ loop ending at statement 66 is the index of the altitude at which the values put into the R array were saved. JJ specifies which of the 9 possible arrays of differences is to be used. The statements following the definition of R(3) define the plot titles. Then IOP(76) is tested.

Consequently, if IOP(76) = 1, the program jumps to statement 60 where the AVCO plot routine EZPLØT is called to set up a plot and then statement 66 is executed. If IOP(76) = 0, XL, XU, YL, and YU which give necessary scaling information for the Aerospace plotter are calculated. PLT is then called three times to produce a plot and statement 66 is executed next. The statement after 66 returns control to INTGRL. The sequence of statements beginning at 70 and ending just before 110 produce one corridor plot. Three curves are plotted on the same graph, X vs. T, Y vs. T, and Z vs. T. X is the expanded array of lower corridor values. Z is the expanded array of upper corridor values. Y is the expanded array of deltas. These deltas are calculated in F123 in the DØ loop ending at statement 33 and expanded in INTGRL. T is the expanded array of altitudes calculated in F123 in the DØ loop ending at statement 50. XL and XU are respectively the smallest and largest values in the T array. YL is the smallest element

in the X, Y and Z arrays and YU is the largest element in these three arrays. These values are needed by the plotter for scaling purposes. The plot titles are defined next, then at statement 95, IOP(76) is tested. If it is zero, XL, XU, YL, and YU are changed as required for the Aerospace plotter. The 3 calls to PLT are then executed and control is returned to INTGRL. If IOP(76) = 1, statement 100 is executed next, then control return to INTGRL.

The sequence of statements beginning at 110 and ending just before 230 can produce any of 28 possible plots depending upon the meaning of X, Y, KY and IC(1).

If N = 1, a plot of Y vs. X will be produced. If N = 2, a plot of log to the base 10 of Y vs. X will be produced. At statement 110, N is set = 1, then if IC(1) = 0 and KY = 4, N is changed to 2. The DØ loop ending at 111 then tests to determine if all of the Y's are equal. If all the Y's are found to be equal, no plot is produced and control returns to VIXEN. If they are not all equal, statement 120 is executed next. The values of Y are then put into the W array in case it proves necessary to take logs. This will be necessary if the Aerospace plotter is called to produce a log-linear plot. The log of each W is taken in the DØ loop ending at 122 if necessary. Statement 125 and the subsequent write statement put the plot title into the B1 array. Then if IOP(76) = 1, statement 130 is executed calling EZPLOT. Control then returns to VIXEN. If IOP(76) = 0, XL, XU, YL, and YU are calculated, and the Aerospace plotter PLT is called to produce a plot. Control then returns to VIXEN.

The statements beginning at 230 close the appropriate plot file and terminate the computer run.

5. Other Information

A. SUBROUTINE AVPLOT is called by either SUBROUTINE INTGRL or SUBROUTINE VIXEN.

B. SUBROUTINE AVPLT calls the SUBROUTINE MAXMIN.

C. SUBROUTINE AVPLT calls the Aerospace routine PLT and the following AVCO library subroutines:

1. SUBROUTINE IDFRMV
2. SUBROUTINE EZPLØT
3. SUBROUTINE FRAMEV
4. SUBROUTINE ENDJØB
5. SUBROUTINE PLTND

D. SUBROUTINE AVPLT calls the following IBM functions:

1. DMIN1
2. DMAX1
3. DLØG10

E. Statements of the form WRITE(9,XXXX), where XXXX is a format number, place information concerning plot titles on tape 9. This information is read back by statements of the form READ(9,YYYY), where YYYY is a format number.

SUBROUTINE MAXMIN(NTABLE, NDIMEN, X, XMIN, XMAX)

1. Purpose

MAXMIN finds the largest value and the smallest value in the X array containing NTABLE numbers. The smallest value is placed in location XMIN and the largest is placed in location XMAX.

2. Input

<u>Name</u>	<u>Source</u>	<u>Description</u>
NDIMEN	AVPLT	adjustable dimension for X array
NTABLE	AVPLT	number of elements in X array
X, XDIMEN	AVPLT	array of numbers to be searched

3. Output

<u>Name</u>	<u>Description</u>
XMIN	smallest number in the X array
XMAX	largest number in the X array

4. Numerical Procedure

Both XMIN and XMAX are set equal to X_1 . Then $X_2, X_3, \dots, X_{NTABLE}$ are compared with XMIN. If any of these numbers is smaller than XMIN, then XMIN is set equal to that X. Parallel to this $X_2, X_3, \dots, X_{NTABLE}$ are being successively compared with XMAX. If any X is greater than XMAX, then XMAX is set equal to this X.

5. Other Information

A. SUBROUTINE MAXMIN is called by AVPLT

B. SUBROUTINE MAXMIN calls no other subroutines or functions.

SUBROUTINE PLT

1. Purpose

SUBROUTINE PLT is a dummy subroutine included so that the calls to the AEROSPACE plotter can be simulated. At AEROSPACE, this should be removed before the deck is compiled or it will replace the AEROSPACE plotter.

2. Input

None

3. Output

None

4. Numerical Procedure

When PLT is entered, control is returned immediately to AVPLT.

5. Other Information

A. SUBROUTINE PLT is called by AVPLT only.

B. SUBROUTINE PLT does not call or reference any othersubprogram.

8.0 LIBRARY SUBROUTINES

8.1 Arc Cosine Function - ACØS (ACØSR, ACØSD)

		 research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER F4-022
PROGRAMMER N. DeBolt	MACHINE IBM 360/75	DATE 19 June 1967	PAGE 1 OF 1	
			BCD NAME ACOS, ACOSR, ACOSD	

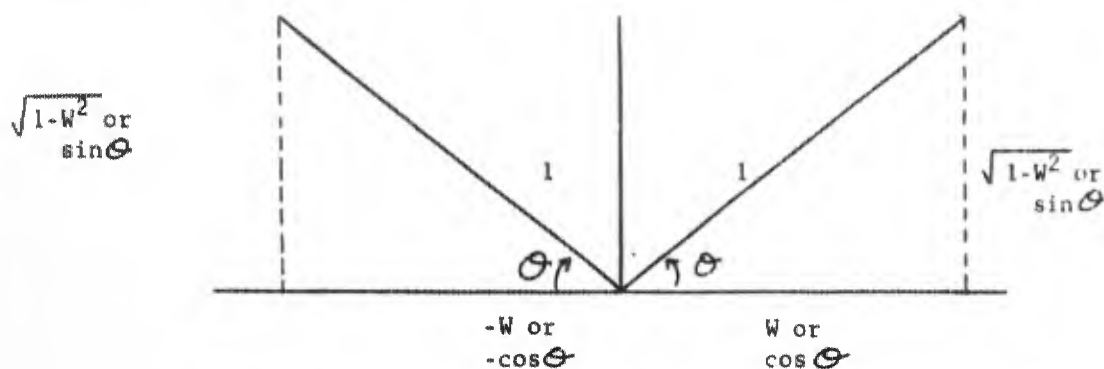
IDENTIFICATION: Arc Cosine Function

USAGE:

X = ACOS(W) X returned in radians
 X = ACOSR(W) X returned in radians
 X = ACOSD(W) X returned in degrees

In all the above cases, W must be between -1.0 and +1.0. Also, W must be given in REAL*8 and X is returned in REAL*8.

EXPLANATION: (Method)



The arc cosine is computed from the arc tangent i.e.

$$\theta = \arccos(W) = \arctan\left(\frac{\sqrt{1-W^2}}{W}\right)$$

Since $\sqrt{1-W^2}$ is always positive (because the Fortran square root function always returns positive numbers) and W can be positive or negative, the angle returned will be between 0 and π (as shown in the graph).

NOTE: This routine was compiled under OPT = 2.

8.2 Arc Sine Function - ASIN (ASINR, ASIND)

		Avco CORPORATION research and advanced development division PROGRAMMER'S HANDBOOK	WRITE-UP NUMBER	F4-024
			PAGE	1 OF 1
PROGRAMMER	MACHINE	DATE	BCD NAME	
M. DeBolt	IBM 360/75	26 June 1967	ASIN(W), ASINR(W), ASIND(W)	

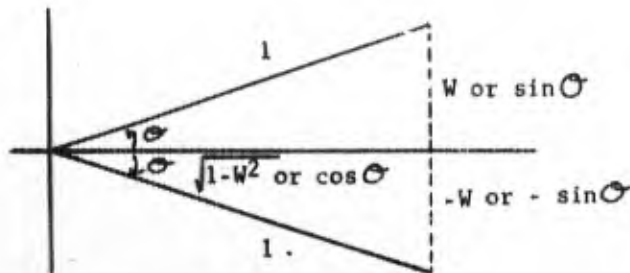
IDENTIFICATION: Arc Sine Function

USAGE:

X = ASIN(W) X returned in radians
 X = ASINR(W) X returned in radians
 X = ASIND(W) X returned in degrees

In all the above cases W must be between -1.0 and +1.0. Also W must be given as REAL*8 and X is returned as REAL*8.

EXPLANATION: (Method)




The arcsine is computed from the arc tangent, i.e.

$$= \text{ARCSIN}(W) = \text{ARCTAN} \left(\frac{W}{\sqrt{1-W^2}} \right)$$

Since $\sqrt{1-W^2}$ is always positive (because the Fortran square root function always returns positive numbers) and W can be positive or negative, the angle returned will be either in the first or fourth quadrant (as shown in the graph). The angles in the fourth quadrant are returned in their positive form because if the angles are negative, they are added to 2π . For example for the angle $-\pi/6$ the answer is returned as $\frac{5\pi}{6}$.

NOTE: This routine was compiled under OPT = 2.

8.3 Arc Tangent Function - ATANQ (ATANQR, ATANQD)

 Alvco <small>CORPORATION</small> research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER IHC-025	
		PAGE 1 0 1	
PROGRAMMER M. DeBolt	MACHINE IBM 360/75	DATE 26 June 1967	BCD NAME ATANQ(Y,X), ATANQR(Y,X), ATANQD(Y,X)

IDENTIFICATION: Arc Tangent Function

USAGE:

A = ATANQ(Y,X) A is returned in radians
A = ATANQR(Y,X) A is returned in radians
A = ATANQD(Y,X) A is returned in degrees

X and Y must both be REAL*8 and the answer A is also returned as a REAL*8. If X and Y are both equal to zero, the program will terminate and an error will be given (IHC2651).

EXPLANATION:

The arc tangent is computed by using the Fortran supplied routine DATAN2, i.e., it finds the angle whose tangent is Y/X (arguments as above). The only difference between these routines (DATAN2 and ATANQ) is that ATANQ always returns the angle in its positive form from 0 to 2π . It checks to see if an angle is negative and if it is, it is added to 2π .

NOTE: This routine was compiled under OPT = 2.


IHC-025

S.4 FIXIBCOM (MIBCOM) has been dummied out of the program.

Originally this routine was a modification of the IBM system routine IBCOM.

8.5 FERRØR has been dummied out of the program. Originally
it was related to the error traceback of MIBCOM.

8.6 Variable Field Input Subroutines - LA000000 (WHERE, SETUP,
READIN, HEDING)

		 research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER F4-005
PROGRAMMER L. Atkinson, R430	MACHINE OS/360	DATE 8 September 1966	PAGE 1 of 10	
			BCD NAME 1) WHERE, 2) SETUP 3) READIN, 4) LADING	

Language: OS/360 Fortran
IV and BAL

IDENTIFICATION:

Title: Variable Field Input Subroutines

PURPOSE:

To perform the same input operations on OS/360 as BCDCON/SYMBLS* does on the 7094. These routines are also more flexible in the inputting of subscripted variables.

USAGE:

Calling Sequence:

```
CALL WHERE (TABLE)
CALL SETUP (SHNAME , L, XY, N1, N2,....)
CALL READIN (INCØL1)
CALL LADING (MTAPE)
```

Complete description of procedures for proper use of these subroutines:

- A. WHERE: This routine is called to inform the routines of the location of a table which will be generated by SETUP and used by READIN. The size of the table must be set by the programmer such that the number of words in the table must be greater than or equal to the number of input variables times seven. The best way to do this is to use the explicit type statement:

REAL* 4 TABLE (ND),

where $ND \geq 7$ times the number of input variables. WHERE must be called before any of the other routines are, and should never be called more than once.

- B. SETUP: This routine is similar to BCDCON in that it generates a table of 8 character EBCDIC names versus associated core addresses. Variable length information must be included in the argument list and subscript information is included as arguments rather than in a hollerith field, as is the case with BCDCON. It is

* Reference: Writeup F2-21 of the Programmer's Handbook.



research and advanced development division

PROGRAMMER'S HANDBOOK

WRITE-UP NUMBER

PAGE 2 OF 10

PROGRAMMER
L. Atkinson, R430

MACHINE
ØS/360

DATE
8 September 1966

BCD NAME 1) WHERE 2) SETUP
3) READIN, 4) HEDING

Language: ØS/360 Fortran
IV and BAL

called once for each variable as follows:

The first argument is the external name of the variable and must have at least eight characters; however, only the first eight characters will be used as the table entry, so any symbols greater than eight characters having their first eight the same will generate the same symbol in the table and only the first one encountered will be effective. If the name has less than eight characters, it must be left adjusted and the eight character field filled with blanks.

The second argument, L, is the length of the variable, in bytes; i.e.

- 1 for LOGICAL*1,
- 2 for INTEGER*2,
- 4 for LOGICAL*4, INTEGER*4, REAL*4,
- 8 for REAL*8.

The third argument is the internal variable name (i.e., the core location with which the external name is to be associated). This can be subscripted to associate the symbol with core locations other than the beginning of an array.

The remaining arguments are the subscripts. (These are omitted if the variable is not dimensioned). Except for the right most subscript, these must match the dimensions of the variable exactly. The value for the right most subscript need not be correct, but a value must be given in the argument list. The maximum number of subscripts is seven; if more are given, only the first seven will be used. The array length is also somewhat restricted: e.g., if a variable has n subscripts then the product of the left-most (n-1) of them must be less than or equal to 32767. For example, a singly-subscripted variable may have dimension 50000 (say) but a doubly-subscripted variable may not have its first subscript greater than 32767, although its second subscript may be as large as desired.

This routine, unlike BCDC#N and H#LNAM, does distinguish between single-valued variables and one-dimensional arrays. This can be seen by the following:

```
Given:  CALL SETUP (8X , 8, X, 10)
        CALL SETUP (8Y , 8, Y)
```

108
WFO 1-1181
10-66

		 research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER
PROGRAMMER		MACHINE	DATE	PAGE 3 OF 10
L. Atkinson, R430		ØS/360	8 September 1966	BCD NAME 1) WHERE, 2) SETUP 3) READIN, 4) HEDING
				Language: ØS/360 Fortran IV and BAL

and the data card: X 5.0 3.0 Y 8.0 6.0, the equivalent Fortran statements are:

X(1) = 5.0

X(2) = 3.0

Y = 8.0

Y = 6.0

Therefore the 8.0 will be overwritten, instead of the 6.0 going into the next double-word location above Y. Incidentally, if Y were subscripted on a data card, READIN would terminate execution with an error message since Y is not supposed to be subscripted.

SETUP may be called anywhere in a program (even in other sub-routines) as long as it is called after WHERE has been called and before READIN is called. However SETUP must not be called more than once for any one variable since the calling instructions are not overwritten as BCDCØN does.

The only error is if WHERE is not called before SETUP, in which case SETUP is never executed, and when READIN is called, execution will be terminated with an error message.

- C. READIN: This routine reads in the input data (from Fortran Data Set Reference No. 5, so the "data" card is //ØØ.FØØ5FØØ1 DD *), lists the cards on the output tape (FØØ6FØØ1), obtains the location where the data is to be stored by finding the symbol punched on the card in the SETUP-generated table, converts the data as prescribed, and then stores it into the core location obtained above. This process is continued until a 1-9 punch in column one is encountered, at which point this number is stored in INCØL1, the rest of the card is processed, and control is returned to the calling program.

NOTE: A zero punch in column one will set INCØL1 equal to zero but will not terminate a case. If the routine encounters an E (part of END-ØF-JØØ) in column one, execution is terminated immediately by calling EXIT.

Either set of characters (i.e., either the 029 or 026 keypunch) or a combination of both may be used on data cards. However in alphanumeric fields, if EBCDIC characters (such as parentheses, equals, apostrophe) are desired, the 029 keypunch should be used.

PROGRAMMER
 L. Atkinson, R430

MACHINE
 OS/360

DATE
 8 September 1966

Language: OS/360 Fortran
 IV and BAL

The breaking character used to separate items (fields) on the cards is a blank. Therefore the way in which data is presented to the subroutine is completely variable as it appears in columns 2-72, with the exception that a symbol or a piece of data may not be continued from one card to another. The items on a card may be divided into two categories, symbols, and data fields.

1. Symbols. These define core locations and are of two types, name symbol and subscript field.

a) The name symbol is a set of characters, the first eight of which are used to look up the entry in the table generated by SETUP. The name must begin with an alphabetic character (or \$) and is terminated by a blank or left parenthesis. From the table entry the base core location, the variable length, and the dimension information is obtained. Execution is terminated with an error message if a symbol is not found in the table.

b) The subscript field, enclosed in parentheses, modifies the core location defined by the last name symbol and sets the addressing increment for successive storing of data. Multiple subscripts in the field are separated by commas, and as with 7094/SYMBOLS, blanks may appear anywhere within the parentheses. Data may be stored sequentially (i.e., varying the first, or left most, subscript) or by varying any one of the other subscripts. If the subscripts are not given, the data will be stored sequentially (note this is different from 7094/SYMBOLS). If any number of subscripts (less than or equal to the number of subscripts as dimensioned) is given, the data will be stored by varying the right-most subscript (the same as 7094/SYMBOLS). However, storing on any one of the subscripts may be forced by putting the subscript number immediately after the right hand parenthesis of the subscript field.

Examples: DIMENSION X(4,6,3)

1) X 5.0 6.0 8.0


X(1) 5.0 6.0 8.0

X(1, 1, 1) 5.0 6.0 8.0

X(1, 1, 1) = 5.0

X(2, 1, 1) = 6.0

are all equivalent to

		 <small>INCORPORATION</small> research and advanced development division PROGRAMMER'S HANDBOOK	WRITE-UP NUMBER
PROGRAMMER	MACHINE	DATE	PAGE 5 OF 10
L. Atkinson, R430	08/360	8 September 1966	BCD NAME 1) WHERE, 2) SETUP 3) READLN, 4) HEDING

Language: 08/360 Fortran
IV and BAL

$X(3, 1, 1) = 8.0$

ii) $X(1, 1) \quad 5.0 \quad 6.0 \quad 8.0$ } are equivalent to
 $X(1, 1, 1)2 \quad 5.0 \quad 6.0 \quad 8.0$ }

$X(1, 1, 1) = 5.0$
 $X(1, 2, 1) = 6.0$
 $X(1, 3, 1) = 8.0$

iii) $X(1, 1, 1) \quad 5.0 \quad 6.0 \quad 8.0$ } are equivalent to
 $X(1, 1, 1)3 \quad 5.0 \quad 6.0 \quad 8.0$ }

or even $X(1)3 \quad 5.0 \quad 6.0 \quad 8.0$

$X(1, 1, 1) = 5.0$
 $X(1, 1, 2) = 6.0$
 $X(1, 1, 3) = 8.0$

This really comes in handy when one wants to store by varying the second subscript (say), starting at some point in the middle of the array. Example: to start at $X(3, 2, 2)$, in 7094/SYMBOLS one would have to write $X(27, 2)$ since $27 = 4 \times 6 \times 1 + 3$ and must be computed from the dimensions. Now one only needs to write $X(3, 2, 2)2$

As in 7094/SYMBOLS there is a provision for using subscripts to define areas to be set to a value defined by a piece of data which follows.
Example:

$X(3, 2 - 4, 2) \quad 5.0$ is equivalent to
 $X(3, 2, 2) = 5.0$
 $X(3, 3, 2) = 5.0$
 $X(3, 4, 2) = 5.0$

Any one of the subscripts may be of the form $K1 - K2$ (where $K1 < K2$). Also, if another data field appears next, it will be stored at the location corresponding to $K2 + 1$, with the other subscripts the same.

1-1101
10-68

PROGRAMMER
 L. Atkinson, R430

MACHINE
 S/360

DATE
 8 September 1966

BCD NAME 1) WHERE, 2) SETUP
 3) READIN, 4) READING
 Language: S/360 Fortran
 IV and BAL

Example: If "6.0" follows "5.0" in the previous example, it will be stored in X(3, 5, 2).

If the number of subscripts on a data card exceeds the number given in the argument list of SETUP for that variable, execution is terminated with an error message.

2. Data Fields. These can be divided into six items:

- a) Integers are defined by lack of a decimal point, E, D, or an imbedded plus or minus sign. They can be positive or negative, and a plus is assumed if there is no sign in front. For a length 4 variable the value must not exceed 2147483647 (2³¹ - 1) and if the value exceeds 32767 for an INTEGER*2, the value stored will be modulo 32768. Also one must be careful when reading an integer into a length 8 variable since the second word of the double word will not be zeroed.
- b) Floating point numbers are defined by a decimal point in the field. However the decimal point may be omitted if an E or D, or imbedded plus or minus or both appears in the field, as XKEYY, XX+YY, XXE-YY.

In this case the decimal point is assumed to be just to the left of the exponent field.

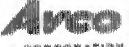
The maximum field length to the left of the decimal point is nine digits. Of the total number of digits in the field, only the first sixteen are converted. Any digits after these are ignored until an exponent field (E, D, +, or -) or a field terminator (blank or parenthesis) is encountered. An E, D, +, or - defines the exponent field, and of course the exponent must be integral. Note that E and D are interchangeable since all conversion is done in double precision (only the store depends on the length of the variable). In general, the permissible forms are

±X.XXXXXXXXXXXXXXXYY

±X.XXXXXXXXXXXXXXX±YY

±X.XXXXXXXXXXXXXXXYY

±X.XXXXXXXXXXXXXXX, where there can be one or more X's, there may be either one or two Y's, the decimal point may appear anywhere to the left of the tenth X (measured from the left), and if no sign appears, it is assumed to be positive.

 research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER
		PAGE 7 OF 10
PROGRAMMER L. Atkinson, R430	MACHINE ØS/360	DATE 8 September 1966
		BCD NAME 1) WHERE, 2) SETUP 3) READIN, 4) HEDING

Language: ØS/360 Fortran
IV and PAL

c) Hexadecimal numbers are input in a manner analogous to the way octal numbers are input by way of 7094/SYMBOLS. A % (or X) in column one of a card signifies to READIN that the card contains only hexadecimal data and nothing else. The data may be read into length 4 or length 8 variables only. A maximum of 8 digits may be read into length 4 variables, and a maximum of 16 digits may be read into length 8 variables. If more than this maximum is given (before a blank), the first 8 (or 16) will be stored correctly, and the digits left over will be right adjusted and stored in the next core location as defined by the last core defining symbol. If less than the maximum is given, the field is right adjusted and stored, and if other than 0 - 9 or A - F appears in the field, execution is terminated with an error message.

d) Alphanumeric data may be defined in three different ways:


- 1) Hollerith card for HEDING. This type of card has an H in column one. Columns 2 - 80 of this card (not 2 - 72 as with 7094/SYMBOLS) will be read into a specially set aside block of storage as Hollerith characters. Nothing else can be done with the card.
- ii) Alphanumeric data intended for arrays may be read in using an A-card. There are two cards associated with this option. The first card must have an A in column one, a core location defining symbol and a length of field count anywhere in columns 2 - 72. This count is in units of the length of the variable. For instance, to read 24 characters the length would be 6 for a Real*4 variable and 3 for a Real*8 variable. The second card, which must follow immediately, will contain the alphanumeric data to be stored. The characters on this card, starting in column one will be stored until the length of field is exhausted. The rest of the card is then ignored. If the number of characters as obtained from the A card exceeds 72, execution is terminated with an error message.
Example of A-cards:

on the 7094, A TITLE 12 would cause columns 1 - 72 of the next card to be read in. On the 360,

A TITLE 9 (TITLE Real*8)

or
A TITLE 18 (TITLE Real*4)

would also cause columns 1 - 72 of the next card to be read in.

		 research and advanced development division PROGRAMER'S HANDBOOK		WRITE-UP NUMBER	
				PAGE 8 OF 10	
PROGRAMER	MACHINE	DATE	BCD NAME 1) WHERE, 2) SETUP 3) READIN, 4) HEDING		
L. Atkinson, R430	ØS/360	8 September 1966	Language: ØS/360 Fortran IV and BAL		

111) Alphanumeric data intended for single storage. This type of data has a slash at the beginning of the field, and the first blank encountered terminates the processing of the field. The maximum number of characters is 8 for a length 8 variable and 4 for a length 4 variable. If more than the maximum are given on the card, execution is terminated with an error message. If less than the max are given, the characters defined are left adjusted and the remainder of the word (or double word if Real*8) is filled with blanks. Also, to fill X with blanks, one can use

X /

e) Logical data: To input logical data, all that is necessary is to use a decimal point followed by T or F; all characters after this are ignored until a blank is encountered. For instance, the following data cards all will produce the same result

(LOGIC1 = .TRUE.)

```

LOGIC1      .T
LOGIC1      .T.
LOGIC1      .TRUE.
LOGIC1      .TRUEBUTREALLYFALSE

```

One must be careful when reading a logical constant into a length 8 variable, since the second word of the double word will NOT be zeroed.

f) Table generating data. This feature is exactly the same as it is in 7094/SYMBOLS. I. e., data may be generated with a minimum of keypunching when numbers are desired ranging from a lower limit to an upper limit in steps of certain given deltas. The format is as follows:

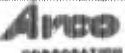
$X_1(X_2)X_3(X_4)X_5$, etc.

This will generate from X_1 to X_3 in steps of X_2 ,

X_3 to X_5 in steps of X_4 , etc.

where the X_i may be floating point or integers, but not mixed. There may be as many deltas as desired and they may be positive or negative. There should be no imbedded blanks in the expression, but they are permitted

RAD 1-1101
10-65

		 research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER
				PAGE 9 OF 10
PROGRAMMER L. Atkinson, R430	MACHINE OS/360	DATE 8 September 1966	BCD NAME 1) WHERE, 2) SETUP 3) READIN, 4) HEDING	
				Language: OS/360 Fortran IV and BAL

before a number and after a parenthesis

i.e., $X_1(X_2)X_3$ is permitted.

3. Transfer card. This card will cause return to the calling program after the remainder of the card has been processed. It is signified by a 1 - 9 punch in column 1. Also, this value is stored in INCØLL.
4. End of job card. This card signifies to READIN that there is no more data to be processed, and that the caller wishes to terminate his run. READIN will then transfer control to EXIT.

Internal and External Controls.

- A) A * or \$ in column 1 will cause the entire card to be ignored (it is printed on the output tape, but NOT on line).
- B) There is no *PRINT or *NØPRI option as there is in 7094/SYMBOLS. All data cards are automatically printed. There is also no ~~COMMON~~ option.
- C) There are no external controls.

ERRORS: Execution is terminated immediately with an error message, if

- i) SETUP has not been called before READIN.
- ii) A name symbol on a data card is not found in the table generated by SETUP.
- iii) A field contains an illegal character (such as a decimal point in a subscript field).
- iv) A variable is given more subscripts than allowed according to the table. Also, if an error is found, processing is terminated immediately. The rest of the data cards up to the transfer card are not scanned for errors, as is the case with 7094/SYMBOLS.

D. HEDING

This routine will write (on NTAPE) the hollerith card read by READIN and eject a page. If an H card was not read in by READIN, the printer will merely restore a page.

PROGRAMMER
L. Atkinson, R430

MACHINE
OS/360

DATE
8 September 1966

SCD NAME 1) WHERE, 2) SETUP
3) READIN, 4) HEDING

Language: OS/360 Fortran
IV and BAL

The only error is in the case that the Fortran data set reference number (NTAPE) is not defined on a DD card, in which case the result is the same as if one tried to execute a Fortran WRITE with the same number.

Summary

1. No complex variables allowed.
2. Either keypunch may be used.
3. Core storage: $292A_{16}$ bytes or 2635_{10} words.
4. External subroutines used:

IBCØM#

FDKPI#

EXIT

		 research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER F4-005-1
PROGRAMMER A. Kinison R430	MACHINE OS 360	DATE October 11, 1966	PAGE 1 OF 1 RCD NAME WHERE SETUP <u>READIN</u> HEDING	

IDENTIFICATION

Title Variable Field Input Subroutines (Addenda)

USAGE

This write-up is an attempt to clarify some points in the previous write-up (F4-005) and also a discussion of two modifications recently made to READIN

A CLARIFICATIONS Referring to page one of the previous write-up in the discussion of WHERE ND must be greater than or equal to seven times the number of input variable names or greater than or equal to seven times the number of calls to SETUP

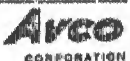
The first sentence in the fourth paragraph on page two should read "The remaining arguments are the dimensions".

in 7094/SYMBOLS a dollar sign anywhere on a data card means that the rest of the card was a comment This is NOT true for READIN A dollar sign is treated as a letter, except when it appears in column one, in which case the entire card is treated as a comment and ignored

Care should be exercised when calling these routines in a source module which has been compiled under OPT=2 The compiler will invariably compile the module incorrectly unless

- 1 All input variables are in COMMON
- 2 The module does no arithmetic calculations but instead calls other subroutines which perform the calculations

Therefore unless one (or both) of the above two conditions is satisfied, the programmer should not use OPT=2 for the module which calls SETUP and READIN

 Avco <small>CORPORATION</small> research and advanced development division PROGRAMMER'S HANDBOOK		WRITE-UP NUMBER F4-005-1
		PAGE 2 OF 3
PROGRAMMER I Atkinson. R430	MACHINE OS 360	DATE October 11, 1966
		BCD NAME WHERE SETUP READIN. HEDING

B MODIFICATIONS

1 READIN has been modified so that it will treat a comma and an equal sign (from either keypunch) as a field terminator for variable names (symbols), subscript fields, and numerical or logical data. However, only a blank (or end of card) terminates alphanumeric and hexadecimal data. This was done to enable READIN to process NAMELIST data cards. The only changes that need be made to the data cards are:

1) The asterisk (for repeated fields) must be removed since READIN will treat the field as illegal. In lieu of the asterisk the dash may be used in the subscript field as follows

A = 20*0 0 becomes A(1-20)= 0 0.

2) The quotes must be removed from alphanumeric data and a slash (or A-card) used instead, as NAME= 'ALPHA' becomes NAME=/ALPHA

3) Logical data must have a decimal point in front of the T or F.

2 An option of returning control to the calling program after encountering the END-OF-JOB card has been added. This has been done so that the programmer can regain control in order to close any data sets (such as the plots, by calling PLTND) which would otherwise be closed abnormally by the system. However, the programmer must be extremely careful not to try to read "tape" 5 afterwards. After calling PLTND (or whatever) he merely needs to CALL EXIT.

This option can be exercised if the programmer adds a second argument in the CALL to READIN, which is a non-standard return, and sets INCOL1 (the first argument) to the value 8888 upon first entry to READIN. This is a flag to inform READIN that there is a non-standard return. An example follows:

```
DATA INCOL1/8888/
```

```
CALL SETUP (etc )
```

```
etc
```

Avco

research and advanced development division

PROGRAMMER'S HANDBOOK

WRITE-UP NUMBER

F4-005-1

PAGE 3 OF 3

PROGRAMER

L. Atkinson R430

MACHINE

OS 360

DATE

October 11, 1966

BCD NAME WHERE SETUP,
READIN, HEDING

1 CALL READIN (INCØL1, \$99) [or&99 if EBCDIC]

[Calculations]

GO TO 1


99 CALL PLTND

CALL EXIT

RETURN

END

It should also be mentioned that no changes are necessary to existing programs which do not use this option as long as they do not set INCØL1 to 8888 initially.

		 Arco <small>CORPORATION</small>		research and advanced development division PROGRAMER'S HANDBOOK		WRITE-UP NUMBER F4-005-2	
PROGRAMER L. Atkinson, R430		MACHINE OS/360		DATE 29 May 1967		PAGE 1 OF 1 RCD NAME WHERE, SETUP, READIN, HEDING	

IDENTIFICATION:

Title: Variable Field Input Routines (Addenda)

USAGE:

READIN has been completely rewritten in BAL in order to increase efficiency and to save core storage. In the process several modifications have been made.

(1) Special characters are now permitted in the names of variables; only the first character need be alphabetic or a dollar sign. Any special characters, except an equals sign or left parenthesis (either keypunch), may be used.

(2) A blank is now permitted between the E or D and the exponent in floating point data fields. READIN treats the blank as a plus sign. Thus READIN is now able to read cards written by a FORTRAN program.

(3) In logical data fields, .Y and .N (for true and false) are now permitted in addition to .T and .F. Therefore the user can have logical print, punch, plot flags in his program to control the output, as follows:

PRINT .YES PUNCH .NØ PLOT .YES

As with .T and .F, all characters after the Y or N are ignored until a blank or comma (or the end of the card) is encountered.

(4) The suppression of printing of the data cards is now possible by punching NØPRINT in columns 1 through 7. To restore the printing mode, the user punches PRINT in columns 1 through 5. Anything else punched on these cards is ignored. The user does not need to start with a PRINT card since READIN is normally in the printing mode.

(5) READIN normally ejects a page for each case. This may be changed to a line skip for each case by punching NØFLIP in columns 1 through 6. To restore the page ejection mode, the user punches FLIP in columns 1 through 4. The remainder of these cards is ignored by READIN.

(6) The core storage requirement of SETUP/READIN has been reduced to 11BB₁₆ bytes or 1135 single precision words.

8.7 The dummy subroutines EZPLOT, ENDJOB, FRAMEV, IDFRMV,
and PLTND replace the AVCO plotting routines of the same
names and must be removed when the program is used at
AVCO.

APPENDIX I - Program Listing and Preset Data

CHAIN	MAIN	1
IMPLICIT-REAL*8-(A-H,O-Z)	MAIN	2
C	MAIN	3
C	MAIN	4
C	MAIN	5
C	MAIN	6
C PURPOSE	MAIN	7
C	MAIN	8
C MAIN DIRECTS THE CALCULATION OF THE OPTIMUM VALUES FOR THE	MAIN	9
C DESIGN PARAMETERS OF A DECOY. THE TECHNIQUE USED IS TO	MAIN	10
C MINIMIZE THE DIFFERENCES BETWEEN CERTAIN CHARACTERISTICS OF A	MAIN	11
C REENTRY VEHICLE AND THE SAME CHARACTERISTICS FOR A DECOY.	MAIN	12
C THE FOLLOWING NINE CHARACTERISTICS MAY BE USED - VELOCITY,	MAIN	13
C DECELERATION, BALLISTIC COEFFICIENT, WAKE LENGTH AT EACH OF	MAIN	14
C THREE RADAR FREQUENCIES AND WAKE CROSS SECTION AT EACH OF THESE	MAIN	15
C FREQUENCIES. FOUR METHODS FOR FINDING THESE MINIMA ARE	MAIN	16
C (1) A FIBONACCI ONE VARIABLE SEARCH (2) A FIBONACCI TWO	MAIN	17
C VARIABLE SEARCH (3) THE DAVIDON METHOD (4) THE ROSENBRUCK	MAIN	18
C ROTATING COORDINATE METHOD) A QUANTITY PD, THE PROBABILI-	MAIN	19
C TY THAT A DECOY WILL BE DISCRIMINATED AS BEING A DECOY, CAN	MAIN	20
C ALSO BE DETERMINED.	MAIN	21
C	MAIN	22
C	MAIN	23
C	MAIN	24
C	MAIN	25
REAL*4 TAB99(2800)	MAIN	26
COMMON /XXSAVE/ X1SAVE(40), X2SAVE(40), KOUNT	MAIN	27
COMMON /IOPT/IPROC,IN,NCONS,IPNT,TEX,LIMIT,IRAND	MAIN	28
COMMON /FOPT/ERR,PRAND, FAC, DELTA	MAIN	29
COMMON /MIN/ALOW(20),UP(20),ANULT(20),CALOW(20),CTP(20)	MAIN	30
COMMON /OCCUR, NOCCUR	MAIN	31
COMMON /CCRN/O	MAIN	32
COMMON /IDNOS/ ID1(50), ID2(50)	MAIN	33
COMMON /DWL/OBJECT(20),WRF,LRED	MAIN	34
COMMON /BLKJ/ HH(1600),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)	MAIN	35
1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FD,E, P,TD,RS,SL, Z,Q,A,EL,	MAIN	36
ZDELZZ,FAK,C(40,10)	MAIN	37
COMMON /SENSE/ISEN1,ISEN2	MAIN	38
COMMON /NALTFG/FGSN,NALT	MAIN	39
COMMON /CICCUR/ IOCCUR(320)	MAIN	40
COMMON /END/ ITERM	MAIN	41
COMMON /MINSK/N,K, ALB(20), UB(20),XX(20),F1	MAIN	42
COMMON /DOPT/DELX(20)	MAIN	43
C	MAIN	44
DIMENSION IOP(90)	MAIN	45
DIMENSION CUP(20)	MAIN	46
DIMENSION OCCUR(4000), NOCCUR(30)	MAIN	47
C	MAIN	48
EQUIVALENCE (NOCCUR(14),NPRINT)	MAIN	49
EQUIVALENCE (OCCUR(4000),LPI)	MAIN	50
EQUIVALENCE (IOCCUR(00001), IOP)	MAIN	51
EQUIVALENCE (IOCCUR(00309), IMPL0T)	MAIN	52
EQUIVALENCE (IOCCUR(00301), IREF)	MAIN	53
EQUIVALENCE (IOCCUR(00303), MUDE)	MAIN	54
C	MAIN	55
C	MAIN	56
C ITERM IS DEFINED IN FEV	MAIN	57
C IF ITERM = -1, FUNCTION = ZERO	MAIN	58
C IF ITERM = 0, FUNCTION IS NOT ZERO	MAIN	59

C	IF ITEM = 1, FUNCTION IS UNDEFINED.	MAIN	60
C	CALL WHERE(TA899)	MAIN	61
	OCCUR(42) = 3.141592653589793	MAIN	62
	CALL SR2490	MAIN	63
	CALL ZPRM	MAIN	64
	MODE = 3	MAIN	65
	P = 0.00	MAIN	66
	D = 0.00	MAIN	67
	DELTA = 1.000	MAIN	68
	FAC = 1.000	MAIN	69
	K = 0	MAIN	70
	ILL111 = 1	MAIN	71
	IMPL0T = 0	MAIN	72
	LRED = 0	MAIN	73
	DO 707 I = 1, 50	MAIN	74
	ID1(I) = 0	MAIN	75
707	ID2(I) = 0	MAIN	76
C		MAIN	77
C		MAIN	78
2	CALL READIT(ILL111, IMPL0T)	MAIN	79
	ILL111 = 2	MAIN	80
	IF(MODE - 2) 100, 110, 120	MAIN	81
C		MAIN	82
100	IREFS = IREF	MAIN	83
	IREF = 1	MAIN	84
	IOP(74) = 0	MAIN	85
	CALL F123(VAL)	MAIN	86
	IREF = IREFS	MAIN	87
	GO TO 2	MAIN	88
110	CALL F123(VAL)	MAIN	89
	GO TO 2	MAIN	90
C		MAIN	91
C		MAIN	92
120	IF(IREF.EQ.2) GO TO 5	MAIN	93
	CALL F123(VAL)	MAIN	94
	GO TO 2	MAIN	95
C		MAIN	96
5	IRED = -1	MAIN	97
C	SAVE QUANTITIES WHICH CAN BE REDEFINED IN REDUCE	MAIN	98
	CTPSAV = CTP(1)	MAIN	99
	IY = ID2(1)	MAIN	100
	YSAV = OCCUR(IY)	MAIN	101
C		MAIN	102
C		MAIN	103
C	SETUP INITIAL STATE	MAIN	104
	DO 101 I = 1, IN	MAIN	105
	I2 = ID1(I)	MAIN	106
	OCCUR(I2) = OVECT(I)	MAIN	107
	X(I) = OVECT(I)	MAIN	108
10	XX(I) = OVECT(I)	MAIN	109
C		MAIN	110
20	GO TO (500, 600, 700, 800, 850), IPROC	MAIN	111
C		MAIN	112
C	SETUP FOR FIBONACCI SEARCH	MAIN	113
500	CONTINUE	MAIN	114
	I2MIN = ID1(1)	MAIN	115
	CALL MIMAX(ALOW(1), UP(1), 1, -1, ERR, LIMIT, XMIN, YMIN)	MAIN	116
	IF(ITEM.EQ.1) GO TO 200	MAIN	117
	X(1) = XMIN	MAIN	118
		MAIN	119

NPSAVE = NPRINT	MAIN 120
NPRINT = 1	MAIN 121
CALL FEV(IN,X,VAL)	MAIN 122
NPRINT = NPSAVE	MAIN 123
GO TO 900	MAIN 124
C	MAIN 125
C	MAIN 126
600 CONTINUE	MAIN 127
CALL FEV(IN,XX,VAL)	MAIN 128
GO TO 900	MAIN 129
C	MAIN 130
C SETUP FOR DAVIDON	MAIN 131
700 CONTINUE	MAIN 132
CALL DAVIDON	MAIN 133
IF(ITERM.EQ.1) GO TO 200	MAIN 134
GO TO 900	MAIN 135
800 CONTINUE	MAIN 136
C	MAIN 137
C SET UP FOR ROSBRK	MAIN 138
CALL ROSBRK	MAIN 139
IF(ITERM.EQ.1) GO TO 200	MAIN 140
NPSAVE = NPRINT	MAIN 141
NPRINT = 1	MAIN 142
CALL FEV(IN,X,VAL)	MAIN 143
NPRINT = NPSAVE	MAIN 144
GO TO 900	MAIN 145
C	MAIN 146
C	MAIN 147
850 DO 860 I = 1, IN	MAIN 148
ALB(I) = ALDW(I)	MAIN 149
UB(I) = UP(I)	MAIN 150
860 CONTINUE	MAIN 151
KOUNT = 0	MAIN 152
CALL GIMAX(ALB(2),UB(2),2,-1,ERR,LIMIT,XX(2),VAL)	MAIN 153
IF(ITERM.EQ.1) GO TO 200	MAIN 154
DO 666 I = 1, 40	MAIN 155
IF(XX(2).NE.X2SAVE(I)) GO TO 666	MAIN 156
XX(I) = X1SAVE(I)	MAIN 157
GO TO 870	MAIN 158
666 CONTINUE	MAIN 159
WRITE(6,867)	MAIN 160
867 FORMAT(1H0 50HERROR IN MAIN. XX(2) UNEQUAL X2SAVE(I) FOR ANY I)	MAIN 161
870 NPSAVE = NPRINT	MAIN 162
NPRINT = 1	MAIN 163
CALL FEV(IN,XX,VAL)	MAIN 164
NPKINT = NPSAVE	MAIN 165
GO TO 900	MAIN 166
C	MAIN 167
C	MAIN 168
C PREPARE TO TIGHTEN CONSTRAINTS.	MAIN 169
900 CALL REDUCE(CTP,KRED,LRED,WRF)	MAIN 170
WRITE(6,1100)KRED, ITERM	MAIN 171
1100 FORMAT(1H0 KRED = '16, ' ITERM = ' 16)	MAIN 172
C	MAIN 173
C	MAIN 174
IF(KRED.EQ.0) GO TO 20	MAIN 175
C	MAIN 176
C RESTORE SAVED QUANTITIES	MAIN 177
200 CTP(I) = CTPSAV	MAIN 178
OCCUR(I) = YSAV	MAIN 179

C

GO TO 2

MAIN 180

MAIN 181

C

END

MAIN 182

MAIN 183

CA00		ADD	1
	IMPLICIT REAL*8(A-M,D-Z)	ADD	2
	FUNCTION ADD(L,R,S,A,B,C,D,E)	ADD	3
	DIMENSION E(9)	ADD	4
C		ADD	5
C		ADD	6
C	FUNCTION ADD EVALUATES THE INTEGRAL OF F(T) DT FROM R TO S	ADD	7
C	OVER A TRAPEZOIDAL REGION IN THE T,Y PLANE DEFINED AS THE AREA	ADD	8
C	ENCLOSED BY THE FOUR LINES - T=R, T=S, Y=A+T*B, Y=C+T*D.	ADD	9
C	F(T) = E(L)*T**2&E(L&1)+T&E(L&2)	ADD	10
C		ADD	11
C		ADD	12
	BD = D-B	ADD	13
	CA = C-A	ADD	14
	C1 = BD*E(L&2)	ADD	15
	C2 = (CA*E(L&2)+BD*E(L&1))/2.00	ADD	16
	C3 = (CA*E(L&1)+BD*E(L))/3.00	ADD	17
	C4 = CA*E(L)/4.00	ADD	18
	ADD = C1*(S-R)+C2*(S*S-R*R)+C3*(S**3-R**3)+C4*(S**4-R**4)	ADD	19
	RETURN	ADD	20
	END	ADD	21

CADMKRK		ADM4R 1
IMPLICIT REAL*8 (A-H,O-Z)		ADM4R 2
SUBROUTINE ADM4RK(NZ,ZDEL,VALUE,DERN,UPBND,DNBND,FACTOR,FREQ,		ADM4R 3
1 HLIMIT,LZ,ZXINDE,DELIMIT)		ADM4R 4
DIMENSION ZBAR(70)		ADM4R 5
2 DIF(70), VALUE(70), DERNN(70),DERN(70),		ADM4R 6
3DERNM1(70),DERNM2(70),DERNM3(70),		ADM4R 7
4VALUEN(70)		ADM4R 8
DIMENSION BO(70),B1(70)		ADM4R 9
DIMENSION UPBND(70),DNBND(70)		ADM4R 10
DIMENSION SAVD(70),SAVE(70)		ADM4R 11
C		ADM4R 12
C		ADM4R 13
C		ADM4R 14
C	SUBROUTINE ADM4RK PERFORMS THE INTEGRATION OF NZ FIRST ORDER	ADM4R 15
C	DIFFERENTIAL EQUATIONS OF THE FORM DY(I)/DX EQUALS THE ITH	ADM4R 16
C	FUNCTION OF (X,Y(1),...,Y(NZ)) FOR VALUES OF I FROM 1 TO NZ BY	ADM4R 17
C	A FOUR POINT PREDICTOR CORRECTOR METHOD WHICH WILL ALTER THE	ADM4R 18
C	INTEGRATION INTERVAL TO MAINTAIN A REQUIRED ACCURACY.	ADM4R 19
C		ADM4R 20
	N=NZ	ADM4R 21
	L=LZ	ADM4R 22
	DEL=ZDEL	ADM4R 23
	XINDEP=ZXINDE	ADM4R 24
	IF (DELMIT) 9002, 9001, 9002	ADM4R 25
9001	DELIMIT = DEL / 1000.000	ADM4R 26
9002	CONTINUE	ADM4R 27
C		ADM4R 28
	IF (L) 1210, 201, 1211	ADM4R 29
1210	L = ABS(L)	ADM4R 30
	GO TO 23	ADM4R 31
1211	GO TO (23,204,204,204,24),L	ADM4R 32
C		ADM4R 33
204	DO 205 JJ = 1,N	ADM4R 34
	DERN (JJ) = SAVD (JJ)	ADM4R 35
	VALUE (JJ) = SAVE (JJ)	ADM4R 36
205	CONTINUE	ADM4R 37
	XINDEP = SINDEP	ADM4R 38
	GO TO 24	ADM4R 39
C		ADM4R 40
23	XFREQ = XINDEP & FREQ	ADM4R 41
	LL = 2	ADM4R 42
	SAVDEL = DEL	ADM4R 43
	SINDEP = XINDEP	ADM4R 44
	ASSIGN 41 TO M	ADM4R 45
	CALL DEREQ (VALUE , XINDEP , DERN , L)	ADM4R 46
	IF(L.EQ.6)GO TO 220	ADM4R 47
	GO TO 40	ADM4R 48
C		ADM4R 49
24	GO TO M, (61, 62)	ADM4R 50
41	DO 62 KK = 1,N	ADM4R 51
	DERNM3 (KK) = DERN (KK)	ADM4R 52
62	CONTINUE	ADM4R 53
C		ADM4R 54
	DO 43 NN = 1,3	ADM4R 55
	W1 = DEL / 2.000	ADM4R 56
	DO 44 I = 1,N	ADM4R 57
	BO(I) = 0.000	ADM4R 58
44	CONTINUE	ADM4R 59

C			ADM4R 60
	DO 50	J = 1,4	ADM4R 61
	GO TO	(45, 46, 48, 49), J	ADM4R 62
45	DIV1	= 6.000	ADM4R 63
47	DIV2	= 2.000	ADM4R 64
	H	= W1	ADM4R 65
	GO TO	60	ADM4R 66
46	DIV1	= 3.000	ADM4R 67
	GO TO	47	ADM4R 68
48	DIV1	= 3.000	ADM4R 69
	DIV2	= 1.000	ADM4R 70
	H	= DEL	ADM4R 71
	GO TO	60	ADM4R 72
49	DIV1	= 6.000	ADM4R 73
C			ADM4R 74
60	DO 51	I = 1, N	ADM4R 75
	B1(I)	= DERN(I) * DEL	ADM4R 76
	B0(I)	= B0(I) & B1(I) / DIV1	ADM4R 77
	GO TO	(52, 52, 52, 53), J	ADM4R 78
52	VALUE(I)	= VALUE(I) & B1(I) / DIV2	ADM4R 79
	TINDEP	= XINDEP & H	ADM4R 80
	GO TO	51	ADM4R 81
53	VALUE(I)	= VALUE(I) & B0(I)	ADM4R 82
51	CONTINUE		ADM4R 83
54	CALL DREQ	(VALUE, TINDEP, DERN, L)	ADM4R 84
	IF(L.EQ.6)GO TO 220		ADM4R 85
50	CONTINUE		ADM4R 86
C			ADM4R 87
	DO 61	II = 1, N	ADM4R 88
	GO TO	(63, 64, 65), NN	ADM4R 89
63	DERNM2(II)	= DERN(II)	ADM4R 90
	GO TO	65	ADM4R 91
64	DERNM1(II)	= DERN(II)	ADM4R 92
65	VALUE(II)	= VALUE(II)	ADM4R 93
61	CONTINUE		ADM4R 94
	XINDEP	= TINDEP	ADM4R 95
C			ADM4R 96
43	CONTINUE		ADM4R 97
	H1	= DEL	ADM4R 98
	H2	= DEL	ADM4R 99
	H3	= DEL	ADM4R100
	ASSIGN	42 TO M	ADM4R101
	GO TO	9042	ADM4R102
C			ADM4R103
42	CONTINUE		ADM4R104
	XFREQ	= XFREQ & FREQ	ADM4R105
9042	CONTINUE		ADM4R106
C			ADM4R107
	GO TO	(1406, 1407, 1407, 1407, 1207), L	ADM4R108
1407	IF	(ABS(XINDEP - XFREQ) - ABS(DEL)) 1403, 1408, 14060	ADM4R109
14060	IF	(ABS(XINDEP - HL(M,I)) - ABS(DEL)) 1408, 1408, 1406	ADM4R110
1408	CONTINUE		ADM4R111
	DO 1410	IJK = 1, N	ADM4R112
	ZBAK(IJK)	= VALUE(IJK)	ADM4R113
	DERNN(IJK)	= DERN(IJK)	ADM4R114
	DERN (IJK)	= DERNM1 (IJK)	ADM4R115
	DERNM1 (IJK)	= DERNM2 (IJK)	ADM4R116
	DERNM2 (IJK)	= DERNM3 (IJK)	ADM4R117
1410	CONTINUE		ADM4R118
C			ADM4R119

LL = 1	ADM4R120
11 TEMP = ABS(XINDEP - XFREQ)	ADM4R121
IF (TEMP - ABS(DEL)) 16, 16, 15	ADM4R122
16 L = 2	ADM4R123
SINDEP = XINDEP	ADM4R124
SAVDEL = DEL	ADM4R125
TEMP1 = ABS(DEL) / DEL	ADM4R126
DEL = TEMP * TEMP1	ADM4R127
15 TEMP = ABS(XINDEP - HLIMIT)	ADM4R128
IF (TEMP - ABS(DEL)) 18, 18, 17	ADM4R129
118 L = 1	ADM4R130
18 L = L & 2	ADM4R131
SAVDEL = DEL	ADM4R132
TEMP1 = ABS(DEL) / DEL	ADM4R133
DEL = TEMP * TEMP1	ADM4R134
17 GO TO (22, 6, 6, 6, 22) , L	ADM4R135
C	ADM4R136
1406 CONTINUE	ADM4R137
LL = 1	ADM4R138
L = 1	ADM4R139
C	ADM4R140
1207 CONTINUE	ADM4R141
12 W1 = H1 & H2	ADM4R142
W2 = W1 * W1	ADM4R143
W3 = W1 & H1	ADM4R144
W4 = W1 & H3	ADM4R145
W5 = W4 & W1	ADM4R146
W6 = W4 & W3	ADM4R147
W7 = W3 & H3	ADM4R148
W8 = W1 * H1	ADM4R149
W9 = W1 * H3	ADM4R150
W10 = H2 & H3	ADM4R151
W11 = H2 * H3	ADM4R152
W12 = H1 * H2	ADM4R153
W16 = W2 & W9	ADM4R154
C	ADM4R155
GO TO (14, 202, 202, 202, 14) , L	ADM4R156
14 TEMP = XINDEP & DEL	ADM4R157
DEL = TEMP - XINDEP	ADM4R158
IF (ABS(DEL) - ABS(DELMIT)) 201, 202, 202	ADM4R159
201 CONTINUE	ADM4R160
L = 6	ADM4R161
GO TO 4	ADM4R162
C	ADM4R163
202 XINDEP = XINDEP & DEL	ADM4R164
W13 = DEL / 2.000	ADM4R165
W14 = W13 * W13 * DEL	ADM4R166
W15 = DEL * DEL / 3.000	ADM4R167
W17 = H1 & DEL	ADM4R168
W18 = W1 & DEL	ADM4R169
W19 = W1 * DEL	ADM4R170
W29 = W3 * DEL	ADM4R171
B0(1) = DEL / W4 * (W14 & W15 * W6 & W13 * (W2 & 2.000 * H1 * W4 & W16) / W8	ADM4R172
X W	ADM4R173
H1(1) = -DEL / W10 * (W14 & W15 * W5 & W13 * W16) / W12	ADM4R174
B2 = DEL / W1 * (W14 & W15 * W7 & W13 * H1 * W4)	ADM4R175
1/W11	ADM4R176
B3 = -DEL / W4 * (W14 & W15 * W3 & W13 * W8) / W10	ADM4R177
1/H3	ADM4R178
	ADM4R179

C		ADM4R180
	DO 1 I = 1,N	ADM4R181
	ZBAR(I) = VALUE(I) & B0(1) * DERN(I) & B1(1) * DERNM1(I) & B2 * DERNM2(I) & B3	ADM4R182
	1 * DERNM3(I)	ADM4R183
1	CONTINUE	ADM4R184
	CALL UEREQ (ZBAR , XINDEP , DERNN , L)	ADM4R185
	IF(L.EQ.6)GO TO 220	ADM4R186
	A = (W14 & W15 * W3 & W13 * W8) / W17 / W18	ADM4R187
	AO = -(W14 & W15 * (W3 - DEL) & W13 * (W8 - W29) - DEL * W8)	ADM4R188
	1/W8	ADM4R189
	A1 = DEL/W17 * (W14 & W15 * (W1 - DEL) - W13 * W19) / W12	ADM4R190
	A2 = DEL / W1 * (W14/3.000 & W1 * W15 / 2.000) / W2 / W 18	ADM4R191
	DO 13 I = 1,N	ADM4R192
	VALUE(I) = VALUE(I) & A * DERNN(I) & AO * DERN(I) & A1 *	ADM4R193
	1 DERNM1(I) & A2 * DERNM2(I)	ADM4R194
	DIF(I) = ABS(VALUE(I) - ZBAR(I))	ADM4R195
C		ADM4R196
	IF (DIF(I) - UPBND(I)) 13 , 13 , 2	ADM4R197
2	XINDEP = XINDEP - DEL	ADM4R198
	DEL = DEL - FACTOR * DEL	ADM4R199
	IF(L-5)7000,14,220	ADM4R200
	7000 CONTINUE	ADM4R201
	L = 1	ADM4R202
	LL = 1	ADM4R203
	GO TO 14	ADM4R204
C		ADM4R205
13	CONTINUE	ADM4R206
	IF(L-2)500,5,500	ADM4R207
500	CONTINUE	ADM4R208
	H3 = H2	ADM4R209
	H2 = H1	ADM4R210
	H1 = DEL	ADM4R211
	GO TO (-11,5,-5,5,22) , L	ADM4R212
200	SAVDEL = DEL	ADM4R213
	GO TO 21	ADM4R214
22	DO 19 K = 1 ,N	ADM4R215
	IF (DIF(K) - DNBND(K)) 19 , 19 , 7010	ADM4R216
19	CONTINUE	ADM4R217
	DEL = DEL & DEL * FACTOR	ADM4R218
7010	IF(L-5)16,200,220	ADM4R219
C		ADM4R220
5	CONTINUE	ADM4R221
	DO 1040 LLL = 1,N	ADM4R222
	VALUE (LLL) = ZBAR(LLL)	ADM4R223
	DERN(LLL) = DERNN(LLL)	ADM4R224
1040	CONTINUE	ADM4R225
21	LL = 2	ADM4R226
C		ADM4R227
	GO TO (6,40,40,40,5) , L	ADM4R228
6	DO 8 J = 1,N	ADM4R229
	DERNM3(J) = DERNM2(J)	ADM4R230
	DERNM2(J) = DERNM1(J)	ADM4R231
	DERNM1(J) = DERN (J)	ADM4R232
	DERN(J) = DERNN(J)	ADM4R233
	VALUE(J) = ZBAR(J)	ADM4R234
8	CONTINUE	ADM4R235
40	CONTINUE	ADM4R236
C		ADM4R237
	GO TO (212,208,208,208,209) , L	ADM4R238
212	CONTINUE	ADM4R239

	GO TO (1407, 1206) , LL	ADM4R240
1206	L = 2	ADM4R241
	GO TO 206	ADM4R242
208	CONTINUE	ADM4R243
	GO TO (206, 4) , LL	ADM4R244
206	CONTINUE	ADM4R245
	DO 203 JJJ = 1,N	ADM4R246
	SAVD (JJJ) = DERN (JJJ)	ADM4R247
	SAVE (JJJ) = VALUE (JJJ)	ADM4R248
209	CONTINUE	ADM4R249
209	CONTINUE	ADM4R250
C		ADM4R251
	GO TO (12 , 4) ,LL	ADM4R252
4	DEL = SAVDEL	ADM4R253
220	NZ = N	ADM4R254
	LZ = L	ADM4R255
	ZDEL = DEL	ADM4R256
	ZXINDE = XINDEP	ADM4R257
	RETURN	ADM4R258
	END	ADM4R259

CAERO	IMPLICIT REAL*8 (A-H,O-Z)	AERO	1
	SUBROUTINE AERODY	AERO	2
C		AERO	3
C	REAL*8 K1L,K2L,K1T,K2T,K3T,K4T,LA,LA1,LA2,LAMDA,LAPS,ME,MINF	AERO	4
C		AERO	5
C	DIMENSION A(514),XLA(8),QDOT(4,8),PEPSR(8),CCUR(4000),NUCCUR(30)	AERO	6
C		AERO	7
C	EQUIVALENCE	AERO	8
	1(OCCUR(010),CAPL),(OCCUR(029),HSRT),(OCCUR(032),LAMDA),	AERO	9
	2(OCCUR(033),LA),(OCCUR(035),MINF),(OCCUR(046),PINF),	AERO	10
	3(OCCUR(047),PS),(OCCUR(052),RN),(OCCUR(055),RHCINF),	AERO	11
	4(OCCUR(064),SINT),(OCCUR(069),THETA), (OCCUR(070),TANT),	AERO	12
	5(OCCUR(076),THETA),(OCCUR(082),V),(OCCUR(091),Z),	AERO	13
	6(OCCUR(092),ZTR),(OCCUR(138),LA1),(OCCUR(144),LA2),	AERO	14
	7(OCCUR(145),ZTURN),(OCCUR(189),FACTR1),	AERO	15
	8(OCCUR(0301), A(1)),(OCCUR(0815), XLA(1)),	AERO	16
	9(OCCUR(2676),QDOT(1,1)),(OCCUR(2801),PEPSR(1))	AERO	17
C		AERO	18
C	COMMON OCCUR,NUCCUR	AERO	19
C		AERO	20
C		AERO	21
C		AERO	22
C		AERO	23
C	SUBROUTINE AERODY CALCULATES HEATING RATES FOR APPROPRIATE	AERO	24
C	BODY STATIONS. FOR A SHARP CONE, THESE STATIONS ARE THE	AERO	25
C	STAGNATION POINT AND THE MAXIMUM DIAMETER POINT. FOR THE	AERO	26
C	BLUNT CONE, THE STAGNATION POINT, SONIC POINT IN TURBULENT	AERO	27
C	FLOW ONLY, TANGENT POINT, THE 20,40,60,75,90 PER CENT STATIONS	AERO	28
C	BASED ON THE INITIAL AXIAL LENGTH, AND THE MAXIMUM DIAMETER	AERO	29
C	POINT - IN ADDITION, SHARP CONE VALUES ARE CALCULATED AT THE	AERO	30
C	MAX. DIAMETER POINT FOR COMPARISON PURPOSES.	AERO	31
C	QDOT(1,1)=STAG. POINT	AERO	32
C	QDOT(2,J)=LAMINAR	AERO	33
C	QDOT(3,J)=TURBULENT	AERO	34
C	QDOT(4,1)=SONIC POINT	AERO	35
C	J=1 TANGENT POINT ON BLUNT CONE	AERO	36
C	J=2 THROUGH 7 CORRESPOND TO THE 20,40,60,75,90 PER CENT	AERO	37
C	STATIONS AND THE MAXIMUM DIAMETER POINT ON THE BLUNT CONE.	AERO	38
C	J=8 MAXIMUM DIAMETER POINT ON THE SHARP CONE OF ANGLE THETA.	AERO	39
C		AERO	40
C		AERO	41
C	CALCULATION OF STAGNATION POINT HEATING RATE	AERO	42
	IF(LAMDA.LT.1.0D-3)RS=1.0D0	AERO	43
	IF(LAMDA.GE.1.0D-3)RS=RN	AERO	44
	QDOTST=1.78D4*SQR(TIRHCINF/(.002375D0*RS))*(V/2.604)**3.15D0	AERO	45
	QDOT(1,1)=QDOTST	AERO	46
C		AERO	47
C	SHARP CONE PE/PS AT MAX DIAMETER	AERO	48
	DO 20 J=1,7	AERO	49
	20 PEPSR(J)=0.0D0	AERO	50
	XLA(8)=1.0D0	AERO	51
	PEPSR(8)=0.0331D0*EXP(.0064D0*THETA-0.33D0*(PIAF-5.0D0)**0.85D0)	AERO	52
	PEPSR(8)=PEPSR(8)&0.468D-3*THETA**1.88D32D0	AERO	53
C		AERO	54
C	BLUNT CONE STATION LOCATIONS	AERO	55
	IF(LAMDA.LT.1.0D-3)GO TO 1	AERO	56
	XLA(1)=RN*(1.0D-SINT)/LA	AERO	57
	IF(Z.GT.ZTURN)DLA=LA1/LA	AERO	58
	IF(Z.LE.ZTURN)DLA=LA2/LA	AERO	59

	XLA(2)=1.000-C.8DC*DLA	AERO	60
	XLA(3)=1.000-C.6DC*DLA	AERO	61
	XLA(4)=1.000-C.4DC*DLA	AERO	62
	XLA(5)=1.000-C.25DC*DLA	AERO	63
	XLA(6)=1.000-C.1DC*DLA	AERO	64
	XLA(7)=1.000	AERO	65
C		AERO	66
C	QUANTITY TEST IS A NON-DIMENSIONAL AXIAL DISTANCE USED IN	AERO	67
C	DETERMINING WHEN FLOW IS CONSIDERED CONICAL	AERO	68
	TEST=1.13DC*RN/(LA*TANT*TANT)	AERO	69
	DO 2 I=1,7	AERO	70
	IF(I.EQ.1)GO TO 16	AERO	71
	IF(XLA(I).LE.XLA(1))GO TO 21	AERO	72
16	CONTINUE	AERO	73
	IF(XLA(I).GE.TEST)GO TO 3	AERO	74
C		AERO	75
	TEM=LA*TANT*TANT*1.004/RN	AERO	76
	XP=C.17400*THETAD	AERO	77
	YP=1C.DC/M INF	AERO	78
	ZP=C.2DC*DLG(XLA(I))*TEM	AERO	79
	WP=C.00	AERO	80
	IF(THETAD.LT.2C.DC)GO TO 13	AERO	81
C	CALCULATION OF BLUNT CONE PRESSURE DISTRIBUTION FOR CONE HALF	AERO	82
C	ANGLE GREATER THAN OR EQUAL TO 20 DEGREES.	AERO	83
	DO 4 N=1,2	AERO	84
	NN=N-1	AERO	85
	ISUB1=416NN	AERO	86
	TEMN=XP**NN	AERO	87
	DO 4 J=1,3	AERO	88
	JJ=J-1	AERO	89
	ISUB2=ISUB1&2*JJ	AERO	90
	TEMJ=YP**JJ	AERO	91
	DO 4 K=1,3	AERO	92
	KK=K-1	AERO	93
	ISUB=ISUB2&6*KK	AERO	94
4	WP=WP&A(ISUB)*TEMN*TEMJ*ZP**KK	AERO	95
	GO TO 15	AERO	96
C	CALCULATION OF BLUNT CONE PRESSURE DISTRIBUTION FOR CONE HALF	AERO	97
C	ANGLE LESS THAN 20 DEGREES.	AERO	98
13	DO 14 N=1,3	AERO	99
	NN=N-1	AERO	100
	ISUB1=1256NN	AERO	101
	TEMN=XP**NN	AERO	102
	DO 14 J=1,3	AERO	103
	JJ=J-1	AERO	104
	ISUB2=ISUB1&3*JJ	AERO	105
	TEMJ=YP**JJ	AERO	106
	DO 14 K=1,3	AERO	107
	KK=K-1	AERO	108
	ISUB=ISUB2&9*KK	AERO	109
14	WP=WP&A(ISUB)*TEMN*TEMJ*ZP**KK	AERO	110
15	PEPSB(I)=WP*TANT*TANT&PI*NF'S	AERO	111
	GO TO 2	AERO	112
21	XLA(I)=XLA(1)	AERO	113
	PEPSB(I)=C.000	AERO	114
	GO TO 2	AERO	115
3	PEPSB(I)=PEPSB(9)	AERO	116
2	CONTINUE	AERO	117
C		AERO	118
C		AERO	119

```

C LAMINAR HEATING RATE CALCULATIONS AERO 120
1 IF(Z.LT.ZTR)GO TO 5 AFRO 121
QBL=10.07*(C.51420C*DLOG(H.SRTD)**0.9736D0) AFRO 122
K1L=0.9664D06THE TAD*(C.00528D060.288D-3*THE TAD) AFRO 123
K2L=1.0D06LAMDA*(1.782D0-2.008D0*LAMDA) AFRO 124
LAPS=L A/P S AFRO 125
IF(LAMDA.GT.1.0D-3)GO TO 6 AFRO 126
C AFRO 127
C SHARP CONE AFRO 128
QDOT(2,8)=QBL*SQRT(PEPSB(8)/(2116.0D*LAPS*XLA(8)))/(K1L*K2L) AFRO 129
RETURN AFRO 130
C AFRO 131
C BLUNT CONE AFRO 132
C LAMINAR TANGENT POINT HEATING AFRO 133
6 QDOT(2,1)=1.732D0*QDOT(1,1)*(0.007789D06PEPSB(1)*(1.849D06PEPSB(1) AFRO 134
1*(-1.6832D060.841D06PEPSB(1)))) AFRO 135
C LAMINAR HEATING ON CONICAL FRUSTUM AFRO 136
DO 7 I=2,8 AFRO 137
7 QDOT(2,I)=QBL*SQRT(PEPSB(I)/(2116.0D*LAPS*XLA(I)))/(K1L*K2L) AFRO 138
RETURN AFRO 139
C AFRO 140
C AFRO 141
C TURBULENT HEATING RATE CALCULATIONS AFRO 142
5 QBTC=10.0D0*(C.7450D*DLOG(H.SRTD)**0.3122D0) AFRO 143
K1T=C.9D060.02D0C*THE TAD AFRO 144
K2T=C.6D0 AFRO 145
K3T=C.65D06LA*(C.0318D0-0.65D-3*LA) AFRO 146
K4T=1.0D0 AFRO 147
IF(LAMDA.GT.1.0D-3)GO TO 8 AFRO 148
C AFRO 149
C SHARP CONE AFRO 150
TM=PS/2116.0D AFRO 151
QDOT(3,8)=QBTC*(PEPSB(8)*TM)**0.8D0/(K1T*K2T*K3T*K4T*XLA(8)**0.165 AFRO 152
XDO) AFRO 153
RETURN AFRO 154
C AFRO 155
C BLUNT CONE AFRO 156
C TURBULENT SONIC POINT HEATING AFRO 157
8 IF(Z.LE.1150C.DC)MB=3.45DC AFRO 158
IF(Z.GT.1150C.DC)MB=2.254D06Z*(2.245D-5&Z*(-146.9D-12&Z*367.1D-18 AFRO 159
X)) AFRO 160
QDOT(4,1)=376C.DC*(RHOINF**C.8D0)*V**MB AFRO 161
QDOT(4,1)=QDOT(4,1)/(RS**0.2D0)*(FACTR AFRO 162
XMB))*(10.0D0**4.0D0)*AFRO 163
TM=PS/2116.DC AFRO 164
QDOT(3,8)=QBTC*(PEPSB(8)*TM)**0.8D0/(K1T*K2T*K3T*K4T*XLA(8)**0.165 AFRO 165
XDO) AFRO 166
C AFRO 167
C TURBULENT TANGENT POINT HEATING AFRO 167
FITANG=(PEPSB(1)**(4.1D0/6.D0))*(1.D0-(PEPSB(1))**(1.0D0/6.D0))**0. AFRO 168
X4D0 AFRO 169
QDOT(3,1)=1.346D4/((RS*(1.0D0-THE TAI)**0.2D0) AFRO 170
QDOT(3,1)=QDOT(3,1)*((RHOINF/2.375D-3)**0.8D0)*FITANG*(0.0001D0*V) AFRO 171
X**MB AFRO 172
C AFRO 173
C ASSIGNMENT OF APPROPRIATE DENOMINATOR FOR HEATING ON STATIONS AFRO 173
AFT OF TANGENT POINT. AFRO 174
TEST=1.13D0*RN/(3.D0*LA*TANT*TANT) AFRO 175
K2T=1.0D0 AFRO 176
IF(LAMDA.LE.C.2D0)K2T=C.6D062.0D0*LAMDA AFRO 177
K3T=C.65D06 LA *(C.0318D0-0.65D-3* LA ) AFRO 178
K4T=1.0D0 AFRO 179

```

C	CFNOM1=K1T*K2T*K3T*K4T	AERO 180
	CFNOM1 FOR XLA .LE. TEST	AERO 181
	K3T=C.65DC&LA*(0.0318DC-C.65D-3*LA)	AERO 182
	DENOM2=K1T*K2T*K3T	AERO 183
C	CFNOM2 FOR XLA .LF. 3.*TEST	AERO 184
	K2T=C.6DC	AERO 185
	K4T=1.0DC	AERO 186
	TEM=LA-RN&RN/SINT	AERO 187
	K3T=C.65DC&TEM*(C.(318DC-0.65D-3*TEM)	AERO 188
	DENOM3=K1T*K2T*K3T*K4T	AERO 189
C	DENOM3 FOR XLA .GT. 3.*TEST	AERO 190
	CO 9 I=2,7	AERO 191
	IF(XLA(I).LE.TEST)GO TO 10	AERO 192
	IF(XLA(I).LE.(3.0C*TEST))GO TO 11	AERO 193
	TM=PS/2116.DC	AERO 194
C	TURBULENT HEATING ON CONICAL FRUSTUM.	AERO 195
	QDOT(3,I)=QBTC*(PEPSB(I)*TM)**0.8DD/(CFNOM3 *XLA(I)**0.165	AERO 196
	XCC)	AERO 197
	GO TO 9	AERO 198
10	DENOM=DENOM1	AERO 199
	GO TO 12	AERO 200
11	TEM=XLA(I)-TEST	AERO 201
	K4T=C.901DD&TEM*(-C.867DD&0.966DD*TEM)	AERO 202
	DENOM=DENOM2*K4T	AERO 203
	TM=PS/2116.DC	AERO 204
12	QDOT(3,I)=QBTC*(PEPSB(I)*TM)**0.8DD/(DENOM)	AERO 205
9	CONTINUE	AERO 206
	RETURN	AERO 207
	END	AERO 208

C	AIM	1
-----	-----	-----
-----	SUBROUTINE- AIM	AIM 2
-----	IMPLICIT REAL*8 (A-H,O-Z)	AIM 3
-----	COMMON/BLK0/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)	AIM 4
-----	1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FC,E, P,TO,RS,SL, Z,Q,A,EL,	AIM 5
-----	2DELTA,FAC,C(40,10)	AIM 6
-----	COMMON/BLK1/M,N,L,LS,M1,MS,NS,IT,K,NC	AIM 7
-----	COMMON /END/ ITEM	AIM 8
C	AIM	9
-----	-----	-----
C	AIM	10
-----	SUBROUTINE AIM, PART OF THE DAVIDON MINIMIZATION METHOD,	AIM 11
-----	ESTIMATES THE LOCATION OF THE RELATIVE MINIMUM WITHIN THE	AIM 12
-----	INTERVAL SELECTED BY SUBROUTINE READY. THIS LOCATION IS	AIM 13
-----	COMPARED WITH THE VALUE THAT WOULD BE EXPECTED FROM A *PERPEN-	AIM 14
-----	DICULAR STEP*.	AIM 15
-----	-----	AIM 16
-----	-----	AIM 17
-----	300-Z=GS&GSP&3.0*(F-FP)/EL	AIM 18
-----	301 TO=GS/Z	AIM 19
-----	TI=GSP/Z	AIM 20
-----	302 O = DSQRT(1.000-10*TI)	AIM 21
-----	O = DABS(Q*Z)	AIM 22
-----	A=(GSP&Q-Z)/(GSP-GS&2.0*Q)	AIM 23
-----	303-TU=(EL*(GSP&Z&2.0*Q)*A**2)/3.0	AIM 24
-----	304 FO=FP-TO	AIM 25
-----	305-CALL MATMP (N,N,H,GP,T)	AIM 26
-----	306 TP1=GSP/SL	AIM 27
-----	307 DO 308 I=1,N	AIM 28
-----	308 T(I)=-T(I)&TP1*S(I)	AIM 29
-----	309 M=1	AIM 30
-----	310 CALL MATMP (M,N,T,GP,GTP)	AIM 31
-----	311-IF-(2.0*TO>P)317,312,312	AIM 32
-----	312 TP1=1.0-A	AIM 33
-----	313 DO 314 I=1,N	AIM 34
-----	314 T(I)=A*X(I)&TP1*XP(I)	AIM 35
-----	315 L=1	AIM 36
-----	316 RETURN	AIM 37
-----	-----	AIM 38
C	AIM	39
-----	-----	-----
C	AIM	40
-----	317 IF-(F>P/2.0)312,318,318	AIM 40
-----	318 DO 319 I=1,N	AIM 41
-----	319 T(I)=T(I)&XP(I)	AIM 42
-----	320 M1=2	AIM 43
-----	-----	AIM 44
C	AIM	45
-----	-----	-----
C	AIM	46
-----	-----	-----
C	AIM	47
-----	-----	-----
C	AIM	48
-----	-----	-----
10	AIM	49
-----	-----	-----
10	AIM	50
-----	-----	-----
10	AIM	51
-----	-----	-----
10	AIM	52
-----	-----	-----
10	AIM	53
-----	-----	-----
10	AIM	54
-----	-----	-----
10	AIM	55
-----	-----	-----
10	AIM	56
-----	-----	-----
10	AIM	57
-----	-----	-----
10	AIM	58
-----	-----	-----
10	AIM	59
-----	-----	-----

330 GSS=GTT	AIM	60
331 SL=GTP	AIM	61
332 EL=1.0	AIM	62
333 L=2	AIM	63
334 GO TO 316	AIM	64
C	AIM	65
335 L=3	AIM	66
336 GO TO 316	AIM	67
1-FORMAT. (9HORICOCHET)	AIM	68
END	AIM	69

CARFDT2	ARFDT 1
IMPLICIT-REAL*8 (A-H,O-Z)	ARFDT 2
SUBROUTINE ARFDT2(H,ASOUND,RHO)	ARFDT 3
C	ARFDT 4
C SUBROUTINE ARFDT2 CALLS IN SUBROUTINE COMP62 AND CONVERTS THE	ARFDT 5
C NON-DIMENSIONAL-DENSITY-OBTAINED FROM IT INTO UNITS OF SLUGS	ARFDT 6
C PER CUBIC FOOT.	ARFDT 7
C	ARFDT 8
RHO = 0.002376900	ARFDT 9
CALL COMP62(H, RHO, ASOUND, T, P)	ARFDT 10
RHO = RHO * RHO	ARFDT 11
RETURN	ARFDT 12
END	ARFDT 13

CAR2DIM		AR2DI 1
	IMPLICIT REAL * 8 (A-H, O-Z)	AR2DI 2
	SUBROUTINE AR2DIM(N, M, X, XT8, Y, YTB, Z, ZTB)	AR2DI 3
C	AR2DIM - NOTE THIS HAS ZTB(J,I) - THE LIBRARY HAS Z(I,J)	AR2DI 4
	DIMENSION XT8(N), YTB(M), ZTB(M, N)	AR2DI 5
C		AR2DI 6
C		AR2DI 7
C	SUBROUTINE AR2DIM PERFORMS A LINEAR INTERPOLATION IN A TWO-	AR2DI 8
C	DIMENSIONAL TABLE FOR Z AS A FUNCTION OF X AND Y.	AR2DI 9
C		AR2DI 10
C		AR2DI 11
	NN = N-1	AR2DI 12
	MM = M-1	AR2DI 13
	DO 5 I = 1, NN	AR2DI 14
	I = I	AR2DI 15
	IF(XT8(I&1) - X) 5, 6, 6	AR2DI 16
6	P2 = (X - XT8(I)) / (XT8(I&1) - XT8(I))	AR2DI 17
	GO TO 7	AR2DI 18
5	CONTINUE	AR2DI 19
C		AR2DI 20
7	DO 9 J = 1, MM	AR2DI 21
	J = J	AR2DI 22
	IF(YTB(J&1) - Y) 9, 10, 10	AR2DI 23
10	P1 = (Y - YTB(J)) / (YTB(J&1) - YTB(J))	AR2DI 24
	GO TO 11	AR2DI 25
9	CONTINUE	AR2DI 26
C		AR2DI 27
11	Z = (1.00 - P1 - P2 - & P1 * P2) * ZTB(J,I) & (P2 - P2 * P1) * ZTB(JAR2DI 28	
	1,I&1) & (P1 - P1 * P2) * ZTB(J&1,I) & P1 * P2 * ZTB(J&1,I&1)	AR2DI 29
C		AR2DI 30
13	RETURN	AR2DI 31
	END	AR2DI 32

```

CAR3DIM                                AR3DI  1
IMPLICIT REAL * 8 (A-H, O-Z)          AR3DI  2
FUNCTION AR3DIM( D , N20, M20, N10 , ROW, H , EM , XYZTBL ) AR3DI  3
DIMENSION D(N20, M20, N10 ) , XYZTBL( 11,3) AR3DI  4
C                                       AR3DI  5
C                                       AR3DI  6
C      FUNCTION AR3DIM IS THE RESULT OF A TABLE LOOK-UP EMPLOYING AR3DI  7
C      LINEAR INTERPOLATION TO DETERMINE THE VALUE OF A DEPENDENT AR3DI  8
C      VARIABLE WHICH IS A FUNCTION OF THREE INDEPENDENT VARIABLES. AR3DI  9
C                                       AR3DI 10
C                                       AR3DI 11
DO 1 I=2, N10                          AR3DI 12
  I=1                                    AR3DI 13
  IF( EM .LE. XYZTBL( I,3) ) GO TO 2    AR3DI 14
1 CONTINUE                              AR3DI 15
C                                       AR3DI 16
  I= N10                                 AR3DI 17
  WRITE(6,1000) XYZTBL(1,3)            AR3DI 18
1000  FORMAT( ' THE VALUE OF M-RAT IS GREATER THAN THE M-RAT VALUE AR3DI 19
      10F THE LARGEST D-TABLE WHICH IS'ID18.6 / ) AR3DI 20
C                                       AR3DI 21
2 CONTINUE                              AR3DI 22
  I= I-1                                 AR3DI 23
  CALL AR2DIM( N20, M20, ROW, XYZTBL(1,1), H, XYZTBL(1,2), AR3DI 24
1 ANS1 , D(1,1,I) ) AR3DI 25
  CALL AR2DIM( N20, M20, ROW, XYZTBL(1,1), H, XYZTBL(1,2), AR3DI 26
1 ANS2 , D(1,1,I) ) AR3DI 27
  AR3DIM = (ANS1 *(XYZTBL( I,3) - EM) + ANS2 * (EM - XYZTBL(11,3 AR3DI 28
1) ) ) / ( XYZTBL( I,3) -XYZTBL(11,3) ) AR3DI 29
C                                       AR3DI 30
RETURN                                  AR3DI 31
END                                     AR3DI 32

```

CAVPLT PROGRAM 2542	AVPLT 1
CAVPLT WITH CORE REMOVED SEPT. 11, 1968	AVPLT 2
CAVPLT	AVPLT 3
SUBROUTINE AVPLT(XQ,IZ,X,Y,Z,KY,IC)	AVPLT 4
IMPLICIT REAL*8(A-H,O-Z)	AVPLT 5
COMMON OCCUR, NOCCUR	AVPLT 6
COMMON /CPCCUR/ PCCUR(11770)	AVPLT 7
COMMON /CICCUR/ IOCCUR(320)	AVPLT 8
REAL*4 B,B1,B2,B3, XQ	AVPLT 9
DIMENSION XQ(40), IOP(90)	AVPLT 10
DIMENSION B1(18),B2(10),B3(10),IC(1), HOLD(3,50), IZ(1)	AVPLT 11
DIMENSION NOCCUR(30),NPV(160),OCCUR(4000),R(3),SAVE(9,160,3),	AVPLT 12
1 W(160)	AVPLT 13
DIMENSION X(1),Y(1),Z(1),ZPLOT(160)	AVPLT 14
DIMENSION T(200)	AVPLT 15
EQUIVALENCE (PCCUR(07101), HOLD)	AVPLT 16
EQUIVALENCE (PCCUR(07251), SAVE)	AVPLT 17
EQUIVALENCE (PCCUR(05561), T)	AVPLT 18
EQUIVALENCE (PCCUR(00161), ZPLOT)	AVPLT 19
EQUIVALENCE (IOCCUR(00308), II)	AVPLT 20
EQUIVALENCE (IOCCUR(00309), IMPL0T)	AVPLT 21
EQUIVALENCE (IOCCUR(00001), IOP)	AVPLT 22
EQUIVALENCE (IOCCUR(00310), JJ)	AVPLT 23
EQUIVALENCE (IOCCUR(000513), KX)	AVPLT 24
EQUIVALENCE (IOCCUR(00307), LPL0T)	AVPLT 25
EQUIVALENCE (IOCCUR(00312), NC0N)	AVPLT 26
EQUIVALENCE (IOCCUR(00305), NDEC0Y)	AVPLT 27
EQUIVALENCE (IOCCUR(00307), NPA)	AVPLT 28
EQUIVALENCE (IOCCUR(00141), NPV)	AVPLT 29
DATA B/4H /	AVPLT 30
C	AVPLT 31
C	AVPLT 32
C SUBROUTINE AVPLT CAN PRODUCE FOUR KINDS OF PLOTS, DEPENDING	AVPLT 33
C ON THE VALUE OF KY. IF KY=1, INFLUENCE COEFFICIENT PLOTS WILL	AVPLT 34
C BE PREPARED AT A SPECIFIED NUMBER OF ALTITUDES, NPA. IF KY=2,	AVPLT 35
C ONE CORRIDOR PLOT IS PRODUCED. IF KY=3, ONE PLOT OF Y VS. X	AVPLT 36
C IS PRODUCED. IF KY=4, ONE PLOT OF LOG(Y) TO THE BASE 10 VS.	AVPLT 37
C X WILL BE PRODUCED. IF KY=5, THE PLOT FILE WILL BE CLOSED AND	AVPLT 38
C THE COMPUTER RUN WILL BE TERMINATED.	AVPLT 39
C	AVPLT 40
C	AVPLT 41
REWIND 9	AVPLT 42
IF(IMPL0T.NE.0) GO TO 10	AVPLT 43
CALL IDFRMV(12HL0RAYNE K210,2H ,0H2542F)	AVPLT 44
IMPL0T = 1	AVPLT 45
C	AVPLT 46
C	AVPLT 47
10 GO TO (20,20,110,110,230), KY	AVPLT 48
20 CONTINUE	AVPLT 49
WRITE(9,2000) IC(1), IC(2)	AVPLT 50
DO 33 I = 1, 19	AVPLT 51
JP = I	AVPLT 52
IF(IZ(I).EQ.NC0N) GO TO 40	AVPLT 53
33 CONTINUE	AVPLT 54
JP = 0	AVPLT 55
GO TO 45	AVPLT 56
40 JS = JP + JP	AVPLT 57
JR = JS - 1	AVPLT 58
45 IF(KY.EQ.2) GO TO 70	AVPLT 59
C	AVPLT 60

C		AVPLT 61
	W(1)=HOLD(1,II)	AVPLT 62
	W(2)=HOLD(2,II)	AVPLT 63
	W(3)=HOLD(3,II)	AVPLT 64
	D0 66 I=1,NPA	AVPLT 65
	J = NPV(I)	AVPLT 66
	R(1) = SAVE(JJ,J,1)	AVPLT 67
	R(2) = SAVE(JJ,J,2)	AVPLT 68
	R(3) = SAVE(JJ,J,3)	AVPLT 69
	WRITE(9,2100)(OCCUR(L),L=127,129),ZPL0T(J)	AVPLT 70
	IF(JP.EQ.0) G0 T0 50	AVPLT 71
	WRITE(9,2200) XQ(JR), XQ(JS)	AVPLT 72
	G0 T0 55	AVPLT 73
50	WRITE(9,2300) NCON	AVPLT 74
55	ENDFILE 9	AVPLT 75
	REWIND 9	AVPLT 76
	READ(9, 3100) B2	AVPLT 77
	READ(9, 2900) B1	AVPLT 78
	READ(9, 3100) B3	AVPLT 79
	REWIND 9	AVPLT 80
	IF(IOP(76).EQ.1) G0 T0 60	AVPLT 81
C		AVPLT 82
	XL = DMIN1(W(1),W(2),W(3))	AVPLT 83
	XU = DMAX1(W(1),W(2),W(3))	AVPLT 84
	YL = DMIN1(R(1),R(2),R(3))	AVPLT 85
	YU = DMAX1(R(1),R(2),R(3))	AVPLT 86
	XU = (XU -XL)/8.00	AVPLT 87
	XL = XL -XU	AVPLT 88
	YU = (YU -YL)/8.00	AVPLT 89
	YL = YL -YU	AVPLT 90
	CALL PLT(1,0,XL,XU,W,0,25,YL,YU,7,0,R,70,B1,40,B3,40,B2)	AVPLT 91
	CALL PLT(2,3,1)	AVPLT 92
	CALL PLT(5,12,PLTIM)	AVPLT 93
	WRITE(6,3000) PLTIM	AVPLT 94
	G0 T0 66	AVPLT 95
C		AVPLT 96
60	CALL EZPL0T(9.,1,1,W,R,3,1,1,55,70,B1,40,B3,40,B2,1,1,XL,XU,1, 1YL,YU,1,0,1)	AVPLT 97
		AVPLT 98
C		AVPLT 99
66	CONTINUE	AVPLT100
	RETURN	AVPLT101
C		AVPLT102
C		AVPLT103
C		AVPLT104
C		AVPLT105
70	XU = ZPL0T(1)	AVPLT106
	XL = ZPL0T(LPL0T)	AVPLT107
	CALL MAXMIN(KX,KX,X ,XMIN,XMAX)	AVPLT108
	CALL MAXMIN(KX,KX,Y ,YMIN,YMAX)	AVPLT109
	CALL MAXMIN(KX,KX,Z ,ZMIN,ZMAX)	AVPLT110
	YL = DMIN1(XMIN,YMIN,ZMIN)	AVPLT111
	YU = DMAX1(XMAX,YMAX,ZMAX)	AVPLT112
	WRITE(9,2500)	AVPLT113
	IF(NDEC0Y.GT.1) G0 T0 80	AVPLT114
	WRITE(9,2400)(OCCUR(L),L=127,129)	AVPLT115
	G0 T0 95	AVPLT116
80	IF(JP.EQ.0) G0 T0 90	AVPLT117
	JT = IZ(JP)	AVPLT118
	WRITE(9,2600)(OCCUR(L),L=127,129),XQ(JR),XQ(JS),OCCUR(JT)	AVPLT119

	GO TO 95	AVPLT120
90	WRITE(9,2700)(@CCUR(L),L=127,129),NC@N,@CCUR(NC@N)	AVPLT121
95	ENDFILE 9	AVPLT122
	REWIND 9	AVPLT123
	READ(9,3100) B2	AVPLT124
	READ(9,3100) B3	AVPLT125
	READ(9,2900) B1	AVPLT126
	REWIND 9	AVPLT127
	IF(10P(76).EQ.1) GO TO 100	AVPLT128
C		AVPLT129
	XU = (XU-XL)/8.0D0	AVPLT130
	XL = XL -XU	AVPLT131
	YU = (YU -YL)/8.0D0	AVPLT132
	YL = YL -YU	AVPLT133
	NF = 123	AVPLT134
	CALL PLT(1,0,XL,XU,T,0,NF,YL,YU,4,1,X,YL,YU,4,1,Y,YL,YU,4,1,Z,	AVPLT135
	1 70,B1,8,B3,8,B2,8,B3,8,B2,8,B3,8,B2)	AVPLT136
	CALL PLT(2,KX,1)	AVPLT137
	CALL PLT(5,12,PLTIM)	AVPLT138
	WRITE(6,3000) PLTIM	AVPLT139
	RETURN	AVPLT140
100	CALL EZPLOT(9,.1,1,T,X,KX,1,2,42,70,B1,40,B3,40,B2,1,-2,XL,XU,-2,	AVPLT141
	1YL,YU,1,0,1)	AVPLT142
	CALL EZPLOT(9,.1,1,T,Z,KX ,1,2,42,1,B,1,B,1,B,2,-2,XL,XU,-2;YL,YU,	AVPLT143
	1 1,0,1)	AVPLT144
	CALL EZPLOT(9,.1,1,T ,Y,KX ,1,2,42,1,B,1,B,1,B,2,-2,XL,	AVPLT145
	1 XU,-2,YL,YU,1,0,1)	AVPLT146
	RETURN	AVPLT147
C		AVPLT148
C		AVPLT149
C		AVPLT150
110	N = 1	AVPLT151
	IF(IC(1).EQ.0.AND.KY.EQ.4) N = 2	AVPLT152
	DO 111 J = 2, LPL0T	AVPLT153
	IF(Y(J)-1).NE.Y(J))GO TO 120	AVPLT154
111	CONTINUE	AVPLT155
	RETURN	AVPLT156
120	DO 121 I = 1, LPL0T	AVPLT157
	W(I) = Y(I)	AVPLT158
121	CONTINUE	AVPLT159
	IF(N.NE.2) GO TO 125	AVPLT160
	IF(KY.NE.4) GO TO 125	AVPLT161
	IF(IC(1).NE.0) GO TO 125	AVPLT162
	DO 122 I = 1, LPL0T	AVPLT163
	W(I) = DLOG10(W(I))	AVPLT164
122	CONTINUE	AVPLT165
125	CONTINUE	AVPLT166
	WRITE(9,2800)(@CCUR(L),L=127,129)	AVPLT167
	ENDFILE 9	AVPLT168
	REWIND 9	AVPLT169
	READ(9,2900) B1	AVPLT170
	REWIND 9	AVPLT171
	IF(10P(76).EQ.1) GO TO 130	AVPLT172
C		AVPLT173
	XU = (ZPLOT(1)-ZPLOT(LPL0T))/8.0D0	AVPLT174
	XL = ZPLOT(LPL0T) -XU	AVPLT175
	CALL MAXMIN(LPL0T,LPL0T,W,YMIN,YMAX)	AVPLT176
	YU = (YMAX-YMIN)/8.0D0	AVPLT177
	YL = YMIN -YU	AVPLT178

	CALL PLT(1,0,XL,XU,X,0,25,YL,YU,4,0,W,70,B1,40,B3,40,B2)	AVPLT179
	CALL PLT(2,LPL0T,1)	AVPLT180
	CALL PLT(5,12,PLTIM)	AVPLT181
	WRITE(6,3000) PLTIM	AVPLT182
	RETURN	AVPLT183
130	CALL EZPLOT(9,,1,N,X,W,LPL0T,1,2,28,70,B1,40,XQ,40,IZ,1,1,XL,XU,1,	AVPLT184
	YL,YU,1,0,1)	AVPLT185
	RETURN	AVPLT186
	C	AVPLT187
	C	AVPLT189
	C	AVPLT139
230	IF(I0P(76).EQ.1) GO TO 240	AVPLT190
	CALL PLT(6)	AVPLT191
	CALL EXIT	AVPLT192
	RETURN	AVPLT193
	C	AVPLT194
	C	AVPLT195
	C	AVPLT196
240	CALL FRAMEV	AVPLT197
	CALL ENDJOB	AVPLT198
	CALL PLTND	AVPLT199
	RETURN	AVPLT200
	C	AVPLT201
	C	AVPLT202
	C	AVPLT203
	C	AVPLT204
2000	FORMAT(1H 5HDELTA 2A4, 27X)	AVPLT205
2100	FORMAT(1H 5HDATE F9.3, 6H CASE F9.3, 6H MEMØ F9.3, 10H ALTITUDE	AVPLT206
	1 F9.1, 9X)	AVPLT207
2200	FORMAT(1H 2A4, 32X)	AVPLT208
2300	FORMAT(1H I12, 28X)	AVPLT209
2400	FORMAT(1H 5HDATE F9.3, 6H CASE F9.3, 6H MEMØ F9.3, 28X)	AVPLT210
2500	FORMAT(1H 8HALTITUDE 32X)	AVPLT211
2600	FORMAT(1H 5HDATE F9.3, 6H CASE F9.3, 6H MEMØ F9.3, 1X 2A4, 1X	AVPLT212
	1 F10.2, 8X)	AVPLT213
2700	FORMAT(1H 5HDATE F9.3, 6H CASE F9.3, 6H MEMØ F9.3, 1X I8, 1X	AVPLT214
	1 F10.2, 8X)	AVPLT215
2800	FORMAT(1H 30HRESULTS OF PROGRAM 2542F DATE F9.3, 6H CASE F9.3,	AVPLT216
	1 6H MEMØ F9.3, 3X)	AVPLT217
2900	FORMAT(1H 18A4)	AVPLT218
3000	FORMAT(1HØ 8HPL0TIM = F10.4)	AVPLT219
3100	FORMAT(1H 10A4)	AVPLT220
	END	AVPLT221

CHESSEL	BESSE 1
IMPLICIT REAL*8 (A-H,O-Z)	BESSE 2
SUBROUTINE BESSEL(XR,XI,C,N,ZNRO,ZNIO,	BESSE 3
IT)	BESSE 4
DIMENSION ZNRO(25,25),ZNIO(25,25)	BESSE 5
C	BESSE 6
C	BESSE 7
C SUBROUTINE BESSEL CALCULATES FOR COMPLEX ARGUMENTS THE BESSEL	BESSE 8
C FUNCTIONS OF INTEGRAL ORDER ZERO THROUGH TWENTY-FOUR. JNXBES	BESSE 9
C IS CALLED TO CALCULATE THE BESSEL FUNCTION OF MAXIMUM ORDER, N,	BESSE 10
C AND OF V-1. THEN A RECURSIVE FORMULA IS USED TO CALCULATE	BESSE 11
C EACH DIMINISHING ORDER UNTIL THE ZEROth ORDER IS DETERMINED.	BESSE 12
C	BESSE 13
C	BESSE 14
N1=NG1	BESSE 15
B=XR**2&XI**2	BESSE 16
IF (B)2,4,2	BESSE 17
C	BESSE 18
4 ZNRG(N1,IT)=0.000	BESSE 19
ZNIO(N1,IT)=0.000	BESSE 20
ZNRO(N,IT)=0.000	BESSE 21
ZNIO(N,IT)=0.000	BESSE 22
GO TO 5	BESSE 23
C	BESSE 24
2 CONTINUE	BESSE 25
CALL JNXBES(XR,XI,N,C,ZNRO(N1,IT),ZNIO(N1,IT))	BESSE 26
NN=N-1	BESSE 27
CALL JNXBES(XR,XI,NN,C,ZNRO(N,IT),ZNIO(N,IT))	BESSE 28
XRB=XR/B	BESSE 29
XIB=-XI/B	BESSE 30
C	BESSE 31
5 DO 1 I=2,N	BESSE 32
NN=N-1	BESSE 33
XNN=FLOAT(NNG1)	BESSE 34
NN1=NN&1	BESSE 35
NN2=NN&2	BESSE 36
NN3=NN&3	BESSE 37
ZNRO(NN1,IT)=2.00*XNN*(ZNRO(NN2,IT)*XRB-ZNIO(NN2,IT)*XIB)-ZNRO(NN3,	BESSE 38
X,	BESSE 39
IT)	BESSE 40
ZNIO(NN1,IT)=2.00*XNN*(ZNRO(NN2,IT)*XIB&ZNIO(NN2,IT)*XRB)-ZNIO(NN3,	BESSE 41
X,	BESSE 42
IT)	BESSE 43
RETURN	BESSE 44
END	

CCNTB		CHNTB	1
	IMPLICIT REAL*8 (A-H,O-Z)	CHNTB	2
	SUBROUTINE CHNTBL(DVALUE,ZTURNX,LCHNGE)	CHNTR	3
C		CHNTR	4
	REAL*8 LAMDA,LA,NSL,NGL,NST,NGT,LAMDA1,LAMDA2,LA1,LA2,LA1F,LAMC1F	CHNTR	5
C		CHNTR	6
	DIMENSION	CHNTR	7
1	TXCGD(50),TAB1(50),TABIX(50),TABZ(50),TW(4,8),	CHNTR	8
2	DOCCUR(4000),NOCCUR(30),DVALUE(16),TABIX1(50),TABIX2(50)	CHNTR	9
3	,TXCGD1(50),TXCGD2(50),TAB11(50),TAB12(50),TABZ1(50),	CHNTR	10
4	TABZ2(50)	CHNTR	11
C		CHNTR	12
	EQUIVALENCE	CHNTR	13
1	(DOCCUR(004),BETA1),(DOCCUR(005),BETA2),(DOCCUR(006),BETA3),	CHNTR	14
2	(DOCCUR(007),BETA4),(DOCCUR(008),COST),(DOCCUR(014),CP2),	CHNTR	15
3	(DOCCUR(015),CPG),(DOCCUR(022),DELRA0),(DOCCUR(023),DELHC),	CHNTR	16
4	(DOCCUR(024),EPSIL),(DOCCUR(025),F)	CHNTR	17
	EQUIVALENCE	CHNTR	18
1	(DOCCUR(030),HREF),(DOCCUR(032),LAMDA),(DOCCUR(033),LA),	CHNTR	19
2	(DOCCUR(038),NSL),(DOCCUR(039),NGL),(DOCCUR(040),NST),	CHNTR	20
3	(DOCCUR(041),NGT),(DOCCUR(042),PI),(DOCCUR(052),RN),	CHNTR	21
4	(DOCCUR(053),RB),(DOCCUR(054),RESINT),(DOCCUR(058),PHC2),	CHNTR	22
5	(DOCCUR(064),SINT),(DOCCUR(066),SQCOST),(DOCCUR(069),THETA),	CHNTR	23
6	(DOCCUR(070),TANT),(DOCCUR(073),TWSTAG),(DOCCUR(074),TW),	CHNTR	24
7	(DOCCUR(076),THETA),(DOCCUR(084),W),(DOCCUR(085),WF),	CHNTR	25
8	(DOCCUR(124),CMOIN1),(DOCCUR(125),CMOIN2)	CHNTR	26
	EQUIVALENCE	CHNTR	27
1	(DOCCUR(133),W1),(DOCCUR(134),THE TAI),(DOCCUR(135),RN1),	CHNTR	28
2	(DOCCUR(136),RR1),(DOCCUR(137),LAMDA1),(DOCCUR(138),LA1),	CHNTR	29
3	(DOCCUR(139),W2),(DOCCUR(140),THE TAZ),(DOCCUR(141),RN2),	CHNTR	30
4	(DOCCUR(142),RR2),(DOCCUR(143),LAMDA2),(DOCCUR(144),LA2),	CHNTR	31
5	(DOCCUR(145),ZTURN),(DOCCUR(146),LA1F),(DOCCUR(147),RR1F),	CHNTR	32
6	(DOCCUR(149),TW1),(DOCCUR(150),THE T1F),(DOCCUR(151),LAMC1F),	CHNTR	33
7	(DOCCUR(168),TW2),(DOCCUR(169),RN1F),(DOCCUR(170),W1F)	CHNTR	34
	EQUIVALENCE	CHNTR	35
1	(DOCCUR(190),FACTR2),(DOCCUR(191),FACTR3),	CHNTR	36
2	(DOCCUR(192),FACTR4),(DOCCUR(193),FACTR5),(DOCCUR(194),FACTR6),	CHNTR	37
3	(DOCCUR(195),FACTR7),(DOCCUR(196),CMOIN),(DOCCUR(204),WTH),	CHNTR	38
4	(DOCCUR(228),WTO TAL)	CHNTR	39
	EQUIVALENCE	CHNTR	40
1	(DOCCUR(0844),TXCGD(1)),(DOCCUR(2894),TAB1(1)),	CHNTR	41
2	(DOCCUR(0944),TABIX(1)),(DOCCUR(2994),TABZ(1)),	CHNTR	42
3	(DOCCUR(2644),TW(1,1)),	CHNTR	43
4	(DOCCUR(2833),TXCGD1(1)),(DOCCUR(2883),TXCGD2(1)),	CHNTR	44
5	(DOCCUR(2933),TAB11(1)),(DOCCUR(2983),TAB12(1)),	CHNTR	45
6	(DOCCUR(3033),TABIX1(1)),(DOCCUR(3083),TABIX2(1)),	CHNTR	46
7	(DOCCUR(3133),TABZ1(1)),(DOCCUR(3183),TABZ2(1))	CHNTR	47
	EQUIVALENCE	CHNTR	48
1	(NOCCUR(01),JJHOLD),(NOCCUR(04),MAXTAB),(NOCCUR(05),NOS EUP),	CHNTR	49
2	(NOCCUR(10),MHEAT),(NOCCUR(13),MATLNO),(NOCCUR(15),NGECM),	CHNTR	50
3	(NOCCUR(16),MXTAB1),(NOCCUR(17),MXTAB2),(NOCCUR(20),MATLNI),	CHNTR	51
4	(NOCCUR(21),MATLN2)	CHNTR	52
C		CHNTR	53
	COMMON OCCUR,NOCCUR	CHNTR	54
C		CHNTR	55
C		CHNTR	56
C		CHNTR	57
C		CHNTR	58
C	SUBROUTINE CHNTRL ASSIGNS THE APPROPRIATE GEOMETRIC AND MATERIAL PROPERTIES AS INITIAL VALUES AT THE REENTRY ALTITUDE	CHNTR	59

C	AND AT ALTITUDE ZTURN, WHERE THE INTEGRATION IS RESTARTED	CHNTR 60
C	AFTER DISCONTINUOUS CHANGE IN CONFIGURATION. THE TABLES WITH	CHNTR 61
C	SUBSCRIPT 1 ARE FOR THE INITIAL CONDITIONS AT REENTRY, THOSE	CHNTR 62
C	SUBSCRIPTED WITH A 2 ARE CONFIGURATION PARAMETERS AT ZTURN.	CHNTR 63
C		CHNTR 64
C		CHNTR 65
	JJHOLD=0	CHNTR 66
	GO TO (100, 200), LCHANGE	CHNTR 67
100	ZTURNX=ZTURN	CHNTR 68
C		CHNTR 69
C	GEOMETRY DESCRIPTION AT REENTRY ALTITUDE Z0	CHNTR 70
	DO 101 J=1, 50	CHNTR 71
	TXCGC(J)=TXCGC1(J)	CHNTR 72
	TAB1(J)=TAB11(J)	CHNTR 73
	TAB1X(J)=TAB1X1(J)	CHNTR 74
101	TABZ(J)=TABZ1(J)	CHNTR 75
	CMQIN=CMQIN1	CHNTR 76
	MAXTAB=MAXTAB1	CHNTR 77
	TW=TW1	CHNTR 78
	JJ=151	CHNTR 79
	MATLNO=MATLN1	CHNTR 80
C		CHNTR 81
	GO TO (102, 103, 104), NGEOM	CHNTR 82
C		CHNTR 83
C	NGEOM=1 INPUT WEIGHT, CONE HALF ANGLE, ACSE AND BASE RADIUS	CHNTR 84
102	W=W1	CHNTR 85
	THETA=THETA1	CHNTR 86
	RN=RN1	CHNTR 87
	RB=RB1	CHNTR 88
	LAMDA1=RN1/RB1	CHNTR 89
	SINTH1=DSIN(THETA1*0.01745329DD)	CHNTR 90
	COSTH1=DCOS(THETA1*0.01745329DD)	CHNTR 91
	LA1=RB1*(COSTH1-LAMDA1*(1.0DD-SINTH1))/SINTH1	CHNTR 92
	GO TO 300	CHNTR 93
C		CHNTR 94
C	NGEOM=2 INPUT WEIGHT, BASE RADIUS, CONE HALF ANGLE, AND	CHNTR 95
C	BLUNTNESS RATIO	CHNTR 96
103	W=W1	CHNTR 97
	THETA=THETA1	CHNTR 98
	RB=RB1	CHNTR 99
	RN=RB1*LAMDA1	CHNTR 100
	SINTH1=DSIN(THETA1*0.01745329DD)	CHNTR 101
	COSTH1=DCOS(THETA1*0.01745329DD)	CHNTR 102
	LA1=RB1*(COSTH1-LAMDA1*(1.0DD-SINTH1))/SINTH1	CHNTR 103
	GO TO 300	CHNTR 104
C		CHNTR 105
C	NGEOM=3 INPUT WEIGHT, NOSE RADIUS, BASE RADIUS, AXIAL LENGTH	CHNTR 106
104	W=W1	CHNTR 107
	RN=RN1	CHNTR 108
	RB=RB1	CHNTR 109
	TEM=LA1-RN1	CHNTR 110
	TEST=TEM*TEM-RN1*RN1&RB1*RB1	CHNTR 111
	IF(TEST.LT.0.0DD)GO TO 35	CHNTR 112
	THETA=180.0DD*ASIN((-LA1-RN1)*RN1&RB1*SQRT(TEST))/	CHNTR 113
	11(TEM&RN1*RN1)/PI	CHNTR 114
	GO TO 300	CHNTR 115
C		CHNTR 116
200	ZTURNX=-1.0D10	CHNTR 117
	LA1F=LA*12.0DD	CHNTR 118
	RB1F=RB*12.0DD	CHNTR 119

	THETA=THETA*57.26578DC	CHNTB120
	RN1F=RN*12.0DC	CHNTB121
	LAMD1F=LAMDA	CHNTB122
	W1F=WTOTAL	CHNTB123
C		CHNTB124
C	GEOMETRY DESCRIPTION AT ALTITUDE ZTURN	CHNTB125
	CD 201 J=1,50	CHNTB126
	TXCGC(J)=TXCGC2(J)	CHNTB127
	TAB1(J)=TAB12(J)	CHNTB128
	TAB1X(J)=TAB1X2(J)	CHNTB129
201	TABZ(J)=TABZ2(J)	CHNTB130
	CMQIN=CMQIN2	CHNTB131
	MAXTAB3=MAXTAB2	CHNTB132
	TWC=TW2	CHNTB133
	JJ=170	CHNTB134
	MATLNO=MATLN2	CHNTB135
C		CHNTB136
	CD TO(202,203,204),NGEOM	CHNTB137
C		CHNTB138
C	NGEOM=1 INPUT WEIGHT, CONE HALF ANGLE, NOSE AND BASE RACII	CHNTB139
202	W=W2	CHNTB140
	THETA=THETA2	CHNTB141
	RN=RN2	CHNTB142
	RB=RB2	CHNTB143
	LAMDA2=RN2/RB2	CHNTB144
	SINTH2=DSIN(THETA2*0.0174532900)	CHNTB145
	COSTH2=DCOS(THETA2*0.0174532900)	CHNTB146
	LA2=RB2*(COSTH2-LAMDA2*(1.0DC-SINTH2))/SINTH2	CHNTB147
	GO TO 300	CHNTB148
C		CHNTB149
C	NGEOM=2 INPUT WEIGHT, BASE RADIUS, CONE HALF ANGLE, AND	CHNTB150
C	BLUNTNESS RATIO	CHNTB151
203	W=W2	CHNTB152
	THETA=THETA2	CHNTB153
	RB=RB2	CHNTB154
	RN=RB2*LAMDA2	CHNTB155
	SINTH2=DSIN(THETA2*0.0174532900)	CHNTB156
	COSTH2=DCOS(THETA2*0.0174532900)	CHNTB157
	LA2=RB2*(COSTH2-LAMDA2*(1.0DC-SINTH2))/SINTH2	CHNTB158
	GO TO 300	CHNTB159
C		CHNTB160
C	NGEOM=3 INPUT WEIGHT, NOSE RADIUS, BASE RADIUS, AXIAL LENGTH	CHNTB161
204	W=W2	CHNTB162
	RN=RN2	CHNTB163
	RB=RB2	CHNTB164
	TEM=LA2-RN2	CHNTB165
	TEST=TEM*TEM-RN2*RN2&RB2*RB2	CHNTB166
	IF(TEST.LT.0.0DC)GO TO 35	CHNTB167
	THETA=180.0DC*ASIN((-(LA2-RN2)*RN2&RB2*SQRT(TEST))/ ((TEST&RN2*RN2))/PI	CHNTB168
	CHNTB169	
300	CONTINUE	CHNTB170
C		CHNTB171
C	MATLNO=1 IS TEFLON	CHNTB172
C	MATLNO=2 IS LTALPHA	CHNTB173
C	MATLNO=3 IS OTWR	CHNTB174
C	MATLNO=4 IS PHENOLIC NYLON	CHNTB175
C	MATLNO=5 IS CARBON PHENOLIC	CHNTB176
C	MATLNO=6 IS AN INPUT MATERIAL	CHNTB177
C		CHNTB178
	IF(MATLNO.GT.6)WRITE(6,1,31)MATLNO	CHNTB179

IF(MATLNO.GT.6)MATLNO=1
55 CO TO (21, 27, 32, 38, 39, 33), MATLNO

C
C

TEFLON PROPERTIES

31 BETA1=0.00
BETA2=2.16204
BETA3=0.5900
BETA4=3.64504
HPEF=0.00
F=485.00
RHJ2=136.00
DEL RHO=0.00
CP2=0.2600
CPC=0.00
NSL=0.5200
NST=0.28200
NGL=0.00
NGT=0.00
DEL HC=0.00
EPSIL=0.400
CO TO 34

C
C

LYALPHA PROPERTIES

37 BETA1=0.000
BETA2=0.541700
BETA3=0.500
BETA4=1.604
HPEF=0.000
F=207.000
RHJ2=66.400
DEL RHO=0.000
CP2=0.5200
CPC=0.000
NSL=0.5700
NST=0.45800
NGL=0.000
NGT=0.000
DEL HC=0.000
EPSIL=0.7500
CO TO 34

C
C

OTWR PROPERTIES

32 BETA1=0.00
BETA2=1.185010
BETA3=-0.5000
BETA4=1.1605
HPEF=0.00
F=2585.00
RHJ2=82.00
DEL RHO=20.00
CP2=0.2200
CPC=0.42300
NSL=0.60700
NST=0.30300
NGL=0.84700
NGT=0.37200
DEL HC=950.00
EPSIL=0.500
CO TO 34

C

CHNTR180
CHNTR181
CHNTR182
CHNTR183
CHNTR184
CHNTR185
CHNTR186
CHNTR187
CHNTR188
CHNTR189
CHNTR190
CHNTR191
CHNTR192
CHNTR193
CHNTR194
CHNTR195
CHNTR196
CHNTR197
CHNTR198
CHNTR199
CHNTR200
CHNTR201
CHNTR202
CHNTR203
CHNTR204
CHNTR205
CHNTR206
CHNTR207
CHNTR208
CHNTR209
CHNTR210
CHNTR211
CHNTR212
CHNTR213
CHNTR214
CHNTR215
CHNTR216
CHNTR217
CHNTR218
CHNTR219
CHNTR220
CHNTR221
CHNTR222
CHNTR223
CHNTR224
CHNTR225
CHNTR226
CHNTR227
CHNTR228
CHNTR229
CHNTR230
CHNTR231
CHNTR232
CHNTR233
CHNTR234
CHNTR235
CHNTR236
CHNTR237
CHNTR238
CHNTR239

NOT REPRODUCIBLE

C		PHENOLIC NYLON PROPERTIES	
38	BETA1=0.000		CHNTB240
	BETA2=0.000		CHNTB241
	BETA3=0.000		CHNTB242
	BETA4=0.000		CHNTB243
	HREF=0.000		CHNTB244
	F=400.000		CHNTB245
	RHO2=30.000		CHNTB246
	DEL RHO=45.000		CHNTB247
	CP2=0.3800		CHNTB248
	CPG=0.000		CHNTB249
	NSL=0.5200		CHNTB250
	NST=0.2000		CHNTB251
	NGL=0.5200		CHNTB252
	NGT=0.2000		CHNTB253
	DEL HC=0.000		CHNTB254
	EPSIL=0.8000		CHNTB255
	GO TO 34		CHNTB256
			CHNTB257
			CHNTB258
C			CHNTB259
C		CARBON PHENOLIC PROPERTIES	
39	BETA1=0.000		CHNTB260
	BETA2=5.100		CHNTB261
	BETA3=0.2900		CHNTB262
	BETA4=1.6105		CHNTB263
	HREF=0.000		CHNTB264
	F=11050.000		CHNTB265
	RHO2=85.700		CHNTB266
	DEL RHO=0.000		CHNTB267
	CP2=0.34800		CHNTB268
	CPG=0.000		CHNTB269
	NSL=0.000		CHNTB270
	NST=0.000		CHNTB271
	NGL=0.000		CHNTB272
	NGT=0.000		CHNTB273
	DEL HC=0.000		CHNTB274
	EPSIL=0.8000		CHNTB275
	GO TO 34		CHNTB276
			CHNTB277
			CHNTB278
C			CHNTB279
C		INPUT MATERIAL PROPERTIES	
33	BETA1=OCCUR(JJ&1)		CHNTB280
	BETA2=OCCUR(JJ&2)		CHNTB281
	BETA3=OCCUR(JJ&3)		CHNTB282
	BETA4=OCCUR(JJ&4)		CHNTB283
	HREF=OCCUR(JJ&5)		CHNTB284
	F=OCCUR(JJ&6)		CHNTB285
	RHO2=OCCUR(JJ&7)		CHNTB286
	DEL RHO=OCCUR(JJ&8)		CHNTB287
	CP2=OCCUR(JJ&9)		CHNTB288
	CPG=OCCUR(JJ&10)		CHNTB289
	NSL=OCCUR(JJ&11)		CHNTB290
	NST=OCCUR(JJ&12)		CHNTB291
	NGL=OCCUR(JJ&13)		CHNTB292
	NGT=OCCUR(JJ&14)		CHNTB293
	DEL HC=OCCUR(JJ&15)		CHNTB294
	EPSIL=OCCUR(JJ&16)		CHNTB295
34	CONTINUE		CHNTB296
			CHNTB297
C		TESTING CONE HALF ANGLE WITH RANGE OF APPLICABILITY	
	IF(THETA.LT.4.000)GO TO 57		CHNTB298
	IF(THETA.LT.40.000)GO TO 5		CHNTB299

57	WRITE(6,1050)THETA	CHNTR300
1058	FORMAT(1H0 10X,19HINPUT ERROR. THETA=F8.3,40H IS OUTSIDE THE RANGE	CHNTR301
	1 OF 4 TO 40 DEGREE(S)	CHNTR302
	JJHOLD=-1	CHNTR303
	RETURN	CHNTR304
5	CONTINUE	CHNTR305
C		CHNTR306
C	DEFINING TRIGONOMETRIC QUANTITIES	CHNTR307
	THETA=THETA	CHNTR308
	THETA=C.01745329DC*THE TA	CHNTR309
	SINT=SIN(THETA)	CHNTR310
	COST=COS(THETA)	CHNTR311
	TANT=SINT/COST	CHNTR312
	RESINT=1.0C/SINT	CHNTR313
	SQCCOST=COST*COST	CHNTR314
C		CHNTR315
C	DEFINING INITIAL VALUES OF DVALUE ASSOCIATED WITH GEOMETRY	CHNTR316
	DVALUE(5)=W	CHNTR317
	DVALUE(6)=RN	CHNTR318
	DVALUE(7)=PR	CHNTR319
	WO=W	CHNTR320
	WTH=W	CHNTR321
	DVALUE(16)=WTH	CHNTR322
C		CHNTR323
C	DEFINING INITIAL VALUES FOR WALL TEMPERATURE	CHNTR324
	TWSTAG=TWO	CHNTR325
	TW(1,1)=TWC	CHNTR326
	TW(4,1)=TWC	CHNTR327
	DO 36 J=1,8	CHNTR328
	TW(2,J)=TWC	CHNTR329
36	TW(3,J)=TWC	CHNTR330
C		CHNTR331
C	FOLLOWING FACTORS ARE USED IN EVAL	CHNTR332
	FACTR2=DLOG(1.11C507)	CHNTR333
	FACTR3=12.765700G2.55840C	CHNTR334
	FACTR4=RHO 2*CP 2&CPG*DELRHO	CHNTR335
	FACTR5=DELRHO*DFLHC	CHNTR336
	FACTR6=RHO 2*VSL&DFLRHO*NGL	CHNTR337
	FACTR7=RHO 2*NST&DELRHO*NGT	CHNTR338
C		CHNTR339
C	TESTS ON VALUES INPUT FOR OPTICS	CHNTR340
	IF(MATLNO.LT.1)WRITE(6,1031)MATLNO	CHNTR341
	IF(NOSEOP.GT.1)WRITE(6,1103)NOSEOP	CHNTR342
1103	FORMAT(1H0 10X,24H*****WARNING*****NOSEOP=13,43H IS NOT ALLOWED, N	CHNTR343
	OSEOP IS TAKEN TO BE = 0.)	CHNTR344
	IF(MHEAT.GT.1)WRITE(6,1104)MHEAT	CHNTR345
1104	FORMAT(1H0 10X,23H*****WARNING*****MHEAT=13,25H IS THE SAME AS MHE	CHNTR346
	AT = 1.)	CHNTR347
1071	FORMAT(1H0 10X,25H*****WARNING***** MATLNO=13,24H IS THE SAME AS M	CHNTR348
	ATLNO=1)	CHNTR349
	RETURN	CHNTR350
35	JJHOLD=-1	CHNTR351
	WRITE(6,1035)LCHNGE	CHNTR352
1035	FORMAT(1H0 10X,6HTABLE ,12,28H GIVES NEGATIVE SQUARE ROOT.)	CHNTR353
	RETURN	CHNTR354
	END	CHNTR355

CCLASSC		CLASS 1
IMPLICIT REAL*8 (A-H,O-Z)		CLASS 2
SUBROUTINE CLASSC(F321)		CLASS 3
COMMON OCCUR,NOCCUR		CLASS 4
COMMON/END/ITERM		CLASS 5
DIMENSION OCCUR(4000),NOCCUR(30), X(4000), A(514)		CLASS 6
EQUIVALENCE(OCCUR(1),X(1)),(OCCUR(301),A(1)),(NOCCUR(3),IOP)		CLASS 7
EQUIVALENCE (OCCUR(4000),LP)		CLASS 8
C		CLASS 9
C		CLASS 10
C	SUBROUTINE CLASSC CONTAINS CLASSIC FUNCTIONS WHICH ARE	CLASS 11
C	DESIGNED TO PROVIDE DIFFICULT TESTS OF HOW WELL A MINIMIZATION	CLASS 12
C	ROUTINE WORKS.	CLASS 13
C		CLASS 14
C		CLASS 15
	IF(IOP.EQ.3) GO TO 300	CLASS 16
	IF(IOP.EQ.4) GO TO 401	CLASS 17
	IF(IOP.EQ.5) GO TO 501	CLASS 18
	F= 100.000*(X(2)-X(1)**2)**2 & (1.000 - X(1))**2 & A(1)	CLASS 19
	X(100) =F	CLASS 20
	IF(IABS(IOP).EQ. 1) GO TO 999	CLASS 21
C		CLASS 22
	211 B= A(11) & (X(1)-A(7))*(A(10) &(X(1)-A(7))*(A(9) &(X(1)-A(7))*	CLASS 23
	1A(8)))	CLASS 24
	C= A(16) & (X(1)-A(12))*(A(15) &(X(1)-A(12))*(A(14) &(X(1)-A(12))*	CLASS 25
	1A(13)))	CLASS 26
	X(101)= A(2)*(X(2)- B)	CLASS 27
	X(102)= A(3)*(X(2)- C)	CLASS 28
	GO TO 999	CLASS 29
C		CLASS 30
	300 FA =0.000	CLASS 31
	FB =0.000	CLASS 32
	FC =0.000	CLASS 33
	FD =0.000	CLASS 34
	FA = X(1)**2 & X(2)**2 & 2.00*X(3)**2 & X(4)**2 -5.000*X(1)-5.000*X(2)-	CLASS 35
	121.000*X(3) & 7.000*X(4) & A(17)	CLASS 36
	X(200) =FA	CLASS 37
	301 FA2= 0.000 & X(4)-X(3) & X(2)-X(1)-X(4)**2-X(3)**2-X(2)**2-X(1)**2	CLASS 38
	X(201) =FA2	CLASS 39
	302 FA3=10.000 & X(4) & X(1) -2.000*X(4)**2 -X(3)**2- 2.000*X(2)**2	CLASS 40
	1-X(1)**2	CLASS 41
	X(202) =FA3	CLASS 42
	303 FA4= 5.000 & X(4) & X(2)-2.000*X(1) -2.000*X(1)**2-X(2)**2-X(3)**2	CLASS 43
	X(203) =FA4	CLASS 44
	GO TO 999	CLASS 45
C		CLASS 46
	401 F= 4.000*X(1) & X(2) & A(25)*(1.000/X(1) & 1.000/X(2)) & A(26)	CLASS 47
	X(300) =F	CLASS 48
	GO TO 999	CLASS 49
C		CLASS 50
	501 F=A(1)*X(1)**2 & A(2)*X(1)*X(2) & A(3)*X(2)**2	CLASS 51
	1&A(4)*X(1) & A(5)*X(2) & A(6)	CLASS 52
	X(400)= F	CLASS 53
	GO TO 999	CLASS 54
C		CLASS 55
	999 F321=0.0	CLASS 56
	LP = 1	CLASS 57
	RETURN	CLASS 58
	END	CLASS 59

```

CCOMP62                                COMP6 1
-----[IMPLICIT REAL*8 (A-H,O-Z)]----- COMP6 2
SUBROUTINE COMP62(H,RHO,ASOUND,T,P)     COMP6 3
DIMENSION TB(26),HB(26),BM(26),R(26),PB(26),TATM(26) COMP6 4
C                                         COMP6 5
C-----SUBROUTINE COMP62 COMPUTES THE NON-DIMENSIONAL DENSITY AND COMP6 6
C-----PRESSURE, TEMPERATURE IN DEGREES RANKINE, AND SPEED OF SOUND COMP6 7
C-----IN FEET PER SECOND OF THE 1962 STANDARD ATMOSPHERE. FROM COMP6 8
C-----Z=0.0 FEET TO Z=300000. FEET THE SUBROUTINE IS EXACT. FROM COMP6 9
C-----Z=300000. FEET TO 2,275,000. FEET. COMPARES TO LESS THAN 1/2 COMP6 10
C-----OF 1 PER CENT. ABOVE Z=2,275,000. FEET CONSTANT VALUES ARE COMP6 11
C-----USED. COMP6 12
C                                         COMP6 13
C-----DATA TB/ COMP6 14
1 320.6500, 288.1500, 216.6500, 216.6500, 228.6500, 270.6500, COMP6 15
X 270.6500, COMP6 16
2 252.6500, 180.6500, 180.6500, 210.6500, 235.6500, 260.6500, COMP6 17
X 360.6500, COMP6 18
3 560.6500, 960.6500, 1110.6500, 1210.6500, 1350.6500, 1550.6500 COMP6 19
X10, COMP6 20
4 1830.6500, 2160.6500, 2290.6500, 2420.6500, 2590.6500, 2730. COMP6 21
X6500/ COMP6 22
C----- COMP6 23
C-----DATA HB/ COMP6 24
1 -5000.000, 0.000, 11000.000, 20000.000, 32000.000, 47000.000 COMP6 25
X, 52000.000, COMP6 26
2 61000.000, 79000.000, 88743.000, 98451.000, 103294.000, 1081 COMP6 27
X29.000, COMP6 28
3 117776.000, 127374.000, 146541.000, 156071.000, 165571.000, COMP6 29
X144485.000, COMP6 30
4 221967.000, 286476.000, 376312.000, 420240.000, 463526.000, COMP6 31
X548250.000, COMP6 32
5 630530.000/ COMP6 33
C----- COMP6 34
C-----DATA BM/ COMP6 35
1 -0.006500, -0.006500, -0.000, 0.00100, 0.002800, 0.000, -0.00 COMP6 36
X0200, COMP6 37
2 -0.004000, 0.000, 0.003390200, 0.0051620900, 0.0051706310, COMP6 38
3 0.0103659100, 0.0207943400, 0.02089100, 0.01573900, 0.0105263 COMP6 39
X00, COMP6 40
4 0.0074019300, 0.0053359000, 0.0043404800, 0.0036733600, 0.002 COMP6 41
X95938800, COMP6 42
5 0.0030032800, 0.0020069900, 0.0013365700, 0.000/ COMP6 43
C----- COMP6 44
C-----DATA R/ COMP6 45
1 0.1930499970E1, 0.1225002630E1, 0.3639188560-0, 0.8803506570-1, COMP6 46
2 0.1322504580-1, 0.1427537430-2, 0.7594342830-3, 0.2510913390-3, COMP6 47
3 0.2001143100-4, 0.3170151710-5, 0.4974097760-5, 0.2115740120-6, COMP6 48
4 0.9829869270-7, 0.2436293590-7, 0.7591592450-8, 0.1836558950-8, COMP6 49
5 0.1159384650-8, 0.8040257920-9, 0.4349107670-9, 0.1564816980-9, COMP6 50
6 .3588817940-10, .6509508190-11, .3127898720-11, .1579707120-11, COMP6 51
7 0.4648827390-12, 0.1540552600-12/ COMP6 52
C----- COMP6 53
C-----DATA PB/ COMP6 54
1 0.1206762490E4, 0.4726812690E3, 0.1055786160E3, 0.2554036600E2, COMP6 55
2 0.4049310980E1, 0.5173783750E0, 0.2752396320-0, 0.8495007680-1, COMP6 56
3 0.4840930630-2, 0.7688859060-3, 0.1403097440-3, 0.6679547730-4, COMP6 57
4 0.3430974860-4, 0.1176590450-4, 0.5699498800-5, 0.2362555410-5, COMP6 58
5 0.1724303810-5, 0.1303459700-5, 0.7865965740-6, 0.3247275090-6, COMP6 59

```

```

6 0.8797634170-7, -0.1883396430-7, -0.9594460430-8, 0.5120562690-8, COMP6 60
7 0.1612728610-8, 0.5571266340-9/ COMP6 61
C COMP6 62
DATA TWTM/ COMP6 63
1 28.964400, 28.964400, 28.964400, 28.964400, 28.964400, 28.96 COMP6 64
X4400, COMP6 65
2 28.964400, 28.964400, 28.964400, 28.964400, 28.8800, 28.7500 COMP6 66
X, 28.5600, COMP6 67
3 28.0700, 27.5800, 26.9200, 26.6600, 26.4000, 25.8500, 24.7000 COMP6 68
X00, 22.6600, COMP6 69
4 19.9400, 18.8200, 17.9400, 16.8400, 16.1700/ COMP6 70
C COMP6 71
DATA AG /9.8066500/, AMO /28.964400/, ARR /8.31432083/, COMP6 72
2 AR /6356766.000/, CON1 /0.304800/, CON2 /1.8000 COMP6 73
X/, COMP6 74
3 CON3 /0.002115600/, CON4 /263.8305200/ COMP6 75
C COMP6 76
C COMP6 77
IF((H-2275000.000) -.00000100)2,2,3 COMP6 78
C COMP6 79
3 RHO =.134865142D-12 COMP6 80
ASOUND=3413.42933D0 COMP6 81
T =2713.95661D0 COMP6 82
P =.126068033D-11 COMP6 83
GO TO 100 COMP6 84
C COMP6 85
2 ZZ = H * CON1 COMP6 86
AH = AR * ZZ / (AR & ZZ) COMP6 87
DO 7 I = 1,26 COMP6 88
IF(AH - HB(I)) 7,8,9 COMP6 89
IF(AH - HB(I61)) 8,7,7 COMP6 90
8 K = I COMP6 91
GO TO 10 COMP6 92
C COMP6 93
7 CONTINUE COMP6 94
C COMP6 95
10 WS4 = (AH - HB(K)) COMP6 96
TM = TB(K) & (BM(K) * (WS4)) COMP6 97
KK=K COMP6 98
IF(KK.EQ.26)KK=25 COMP6 99
WTM=TWTH(KK)&(AH-HB(KK))*(TWTM(KK&1)-TWTM(KK))/(HB(KK&1)-HB(KK)) COMP6100
T = TM * CON2*(WTM / AMO) COMP6101
C COMP6102
12 IF(BM(K) - 0.00) 15,16,15 COMP6103
C COMP6104
15 WS5 = (TB(K) / TM ) COMP6105
WS6 = (.AG * AMO / (ARR * BM(K))) COMP6106
RHO = (R(K) * WS5 ** (1.00 & WS6)) / R(2) COMP6107
P = (PB(K) * WS5 ** (WS6)) * CON3 COMP6108
GO TO 17 COMP6109
C COMP6110
16 WS7 = EXP ( - ( AG * AMO * WS4) / ( ARR * TB(K) ) ) COMP6111
RHO = R(K) * WS7 / R(2) COMP6112
P = PB(K) * WS7 * CON3 COMP6113
C COMP6114
17 CONTINUE COMP6115
ASOUND = CON4 * SQRT ( T / WTM ) COMP6116
100 RETURN COMP6117
END COMP6118

```


CCAVDON	DAVDD 1
IMPLICIT REAL*8 (A-H,O-Z)	DAVDD 2
SUBROUTINE DAVDON	DAVDD 3
COMMON/IDPT/IPROC,IN,NCONS,IPNT,IEX,LIMIT,IRAND	DAVDD 4
COMMON/FDPT/EKR,PRAND, FAC1, DELTA1	DAVDD 5
COMMON /END/ITERM	DAVDD 6
COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),SP(40),T(40),GB(40)	DAVDD 7
1,GS, GSP, -GTP,GSS, -GTT, GSB,F,FP,FB,FC,E, P,TD,RS,SL, Z,Q,A,EL,	DAVDD 8
2DELTA,FAC,C(40,10)	DAVDD 9
COMMON/SENSE/ISEN1,ISEN2	DAVDD 10
COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC	DAVDD 11
C	DAVDD 12
C	DAVDD 13
C SUBROUTINE DAVDON IS THE CONTROLLING SUBROUTINE FOR THE	DAVDD 14
C CALCULATIONS OF THE DAVIDON VARIABLE METRIC METHOD OF MINIMI-	DAVDD 15
C ZATION, WHICH DETERMINES NUMERICALLY THE LOCAL MINIMA OF	DAVDD 16
C DIFFERENTIABLE FUNCTIONS OF SEVERAL VARIABLES. IN THE PROCESS	DAVDD 17
C OF LOCATING EACH MINIMUM, A MATRIX H WHICH CHARACTERIZES THE	DAVDD 18
C BEHAVIOR OF THE FUNCTION ABOUT THE MINIMUM IS DETERMINED.	DAVDD 19
C	DAVDD 20
C	DAVDD 21
FLIM = 0.000	DAVDD 22
N=IN	DAVDD 23
K=IRAND	DAVDD 24
FAC = FAC1	DAVDD 25
DELTA = DELTA1	DAVDD 26
NC=0	DAVDD 27
ITLIM= LIMIT	DAVDD 28
E = ERR	DAVDD 29
C VARIABLE METRIC MINIMIZATION	DAVDD 30
101 MS=0	DAVDD 31
IF(FAC)106,1066,106	DAVDD 32
106 DO 1064 I=1,N	DAVDD 33
DO 105 J=1,N	DAVDD 34
105 H(I,J) = 0.000	DAVDD 35
H(I,I)=FAC	DAVDD 36
1064 CONTINUE	DAVDD 37
GO TO 109	DAVDD 38
C	DAVDD 39
1066 DO 1068 I=1,N	DAVDD 40
1067 DO 1068 J=1,N	DAVDD 41
1068 H(J,I)=H(I,J)	DAVDD 42
C	DAVDD 43
109 WRITE(6,5)	DAVDD 44
WRITE(6,1)	DAVDD 45
111 WRITE(6,6) N, K, E, P, DELTA	DAVDD 46
112 WRITE(6,7) (X(I),I=1,N)	DAVDD 47
113 WRITE(6,8)	DAVDD 48
114 DO 115 I=1,N	DAVDD 49
115 WRITE (6,9) (H(I,J),J=1,N)	DAVDD 50
116 M1=1	DAVDD 51
1165 F=0.0	DAVDD 52
C	DAVDD 53
C	DAVDD 54
117 CALL FCN(N,G,F,X,M1)	DAVDD 55
IF(ITERM.NE.0) GO TO 149	DAVDD 56
C	DAVDD 57
1175 IF(F-FLIM) 1393,1393,118	DAVDD 58
118 L=1	DAVDD 59

1185 IT=0	DAVDD 60
119 WRITE (6,14)IT,MS,F	DAVDD 61
1195 WRITE (6,7)(X(I),I=1,N)	DAVDD 62
120 WRITE (6,4)	DAVDD 63
1201 IF (NC)121,121,1202	DAVDD 64
1202 DO 1209 J1=1,NC	DAVDD 65
1203 CALL MATMP (N,N,H,C(1,J1),T)	DAVDD 66
1204 CALL MATMP (1,N,T,C(1,J1),TO)	DAVDD 67
1205 IF (TO-E)1209,1209,1206	DAVDD 68
1206 DO 1208 I=1,N	DAVDD 69
1207 DO 1208 J=1,N	DAVDD 70
1208 H(I,J)=H(I,J)-T(I)*T(J)/TO	DAVDD 71
1209 CONTINUE	DAVDD 72
C	DAVDD 73
121 CALL READY	DAVDD 74
IF(ITERM.NE.0) GO TO 149	DAVDD 75
C	DAVDD 76
122 L=L	DAVDD 77
123 GO TO (139,159,137,126),L	DAVDD 78
124 L=2	DAVDD 79
125 GO TO 1201	DAVDD 80
C	DAVDD 81
126 CALL AIM	DAVDD 82
IF(ITERM.NE.0) GO TO 149	DAVDD 83
C	DAVDD 84
127 L=L	DAVDD 85
128 GO TO (129,135,137),L	DAVDD 86
C	DAVDD 87
129 CALL FIRE	DAVDD 88
IF(ITERM.NE.0) GO TO 149	DAVDD 89
C	DAVDD 90
130 L=L	DAVDD 91
131 GO TO (135,132,126),L	DAVDD 92
132 L=1	DAVDD 93
C	DAVDD 94
133 CALL DRESS	DAVDD 95
C	DAVDD 96
FD= F-FLIM	DAVDD 97
IF((FD.LE.0.000).OR.(IT.GT.ITLIN)) GO TO 142	DAVDD 98
1335 L=L	DAVDD 99
134 GO TO (124,162),L	DAVDD100
135 L=2	DAVDD101
136 GO TO 133	DAVDD102
137 L=3	DAVDD103
138 GO TO 133	DAVDD104
139 L=4	DAVDD105
140 GO TO 133	DAVDD106
139 IF(ISEN1) 1395,1395,1393	DAVDD107
1393 WRITE (6,7)(X(I),I=1,N)	DAVDD108
1395 CALL STUFF	DAVDD109
140 L=L	DAVDD110
141 GO TO (1165,142),L	DAVDD111
142 WRITE (6,10)	DAVDD112
IF(IT.LE.ITLIN) GO TO 143	DAVDD113
WRITE(6,900) IT	DAVDD114
900 FURMAT(1H /10X, 'PROGRAM HAS NOT CONVERGED AFTER', I3, ' ITERATIONS'	DAVDD115
1, 1H /)	DAVDD116
143 WRITE (6,11)	DAVDD117
144 DO 145 I=1,N	DAVDD118
145 WRITE (6,9)(H(I,J),J=1,N)	DAVDD119

146 WRITE (6,13)F,GS	DAVDJ120
147 WRITE (6,7)(X(I),I=1,N)	DAVDJ121
148 WRITE (6,12)(G(I),I=1,N)	DAVDJ122
C	DAVDJ123
C	DAVDJ124
149 M1=3	DAVDJ125
150 CALL FCN(N,G,F,X,M1)	DAVDJ126
C	DAVDJ127
C	DAVDJ128
151 IF(ISEN1) 152,152, 157	DAVDJ129
152 CONTINUE	DAVDJ130
156 GO TO 157	DAVDJ131
1 FORMAT (72H IDENTIFICATION TITLE	DAVDJ132
1)	DAVDJ133
2 FORMAT (316,F0.0)	DAVDJ134
3 FORMAT (6D12.5)	DAVDJ135
4 FORMAT (20H0- - - - -)	DAVDJ136
5 FORMAT (29H1VARIABLE METRIC MINIMIZATION)	DAVDJ137
6 FORMAT (3HON=12,4H K=12,4H E=1PE14.5,4H P=E14.5,8H DELTA=E14.5	DAVDJ138
1)	DAVDJ139
7 FORMAT (3HOX=1P8E14.5/(3HO 8E14.5))	DAVDJ140
8 FORMAT (2H0H)	DAVDJ141
9 FORMAT (1HO1P8E14.5/(1HO8E14.5))	DAVDJ142
10 FORMAT (13HOFINAL VALUES)	DAVDJ143
11 FORMAT (13HOERKOK MATRIX)	DAVDJ144
12 FORMAT (3HOG=1P8E14.5/(3HO 8E14.5))	DAVDJ145
13 FORMAT (3HOF=1PE14.5,5H GS=E14.5)	DAVDJ146
14 FORMAT (4HOIT 14,7H STEP 14,4H F=1PE14.5)	DAVDJ147
157 CONTINUE	DAVDJ148
158 CONTINUE	DAVDJ149
162 RETURN	DAVDJ150
END	DAVDJ151

CCREQ		DEREQ 1
	IMPLICIT REAL*8 (A-H,O-Z)	DEREQ 2
	SUBROUTINE DREQ(DVAL,ZUSE,DERIV,LL)	DEREQ 3
C		DEREQ 4
	REAL*8 MDOT,LAMDA,MINF	DEREQ 5
C		DEREQ 6
	DIMENSION ALPTAB(75),CDTAB(75),WHTAB(75),WCDTAB(75),HTAB(75)	DEREQ 7
	DIMENSION OCCUR(400),NOCCUR(30),MDOT(4,8)	DEREQ 8
	DIMENSION DVAL(16),DERIV(16)	DEREQ 9
C		DEREQ 10
	EQUIVALENCE	DEREQ 11
	1(OCCUR(001),ARFF),(OCCUR(002),ALPHA),(OCCUR(003),ALPHIM),	DEREQ 12
	2(OCCUR(016),CD),(OCCUR(026),GAMF),(OCCUR(032),LAMDA),	DEREQ 13
	3(OCCUR(035),MINF),(OCCUR(043),P),(OCCUR(044),PHI),	DEREQ 14
	4(OCCUR(045),PSI),(OCCUR(051),Q),	DEREQ 15
	5(OCCUR(052),RN),(OCCUR(053),RH),(OCCUR(059),RNLET),	DEREQ 16
	6(OCCUR(067),RROUT),(OCCUR(065),SMR),(OCCUR(071),THEALP),	DEREQ 17
	7(OCCUR(073),TWSTAG),(OCCUR(074),TWO),(OCCUR(080),TIMER),	DEREQ 18
	8(OCCUR(082),V),(OCCUR(084),W),(OCCUR(086),WDCT),	DEREQ 19
	9(OCCUR(087),XR),(OCCUR(090),XBAR),(OCCUR(091),Z)	DEREQ 20
	EQUIVALENCE	DEREQ 21
	1(OCCUR(188),AWREF),(OCCUR(199),YR),(OCCUR(200),PSIALP),	DEREQ 22
	2(OCCUR(204),WTH),(OCCUR(237),XUP),	DEREQ 23
	3(OCCUR(270E),MDOT(1,1)),(OCCUR(3233),HTAB(1)),	DEREQ 24
	4(OCCUR(3308),WHTAB(1)),(OCCUR(3383),CDTAB(1)),	DEREQ 25
	5(OCCUR(3458),WCDTAB(1)),(OCCUR(3646),ALPTAB(1))	DEREQ 26
	EQUIVALENCE	DEREQ 27
	1(NOCCUR(03),MDOT),(NOCCUR(05),NOSEOP),(NOCCUR(10),MHEAT),	DEREQ 28
	2(NOCCUR(18),MAXCD),(NOCCUR(19),MAXWCD),(NOCCUR(30),INALPH)	DEREQ 29
C		DEREQ 30
	COMMON OCCUR,NOCCUR	DEREQ 31
C		DEREQ 32
C		DEREQ 33
C		DEREQ 34
C	SUBROUTINE DREQ CALLS IN SUBROUTINES WHICH CALCULATE THE	DEREQ 35
C	DERIVATIVES INTEGRATED BY SUBROUTINE ADM4RK.	DEREQ 36
C		DEREQ 37
C		DEREQ 38
C	INTERNAL PROGRAM VARIABLES ARE SET EQUAL TO CORRESPONDING	DEREQ 39
C	INTEGRATED VALUES FROM INTEGRATION SUBROUTINE ADM4RK.	DEREQ 40
	1 V=DVAL(1)	DEREQ 41
	GAMF=DVAL(2)	DEREQ 42
	TIME=DVAL(3)	DEREQ 43
	Z=ZUSE	DEREQ 44
	XR=DVAL(4)	DEREQ 45
	W=DVAL(5)	DEREQ 46
	RN=DVAL(6)	DEREQ 47
	RH=DVAL(7)	DEREQ 48
	PSI=DVAL(8)	DEREQ 49
	THEALP=DVAL(9)	DEREQ 50
	PHI=DVAL(10)	DEREQ 51
	Q=DVAL(11)	DEREQ 52
	SMR=DVAL(12)	DEREQ 53
	W=DVAL(13)	DEREQ 54
	YR=DVAL(14)	DEREQ 55
	PSIALP=DVAL(15)	DEREQ 56
	WTH=DVAL(16)	DEREQ 57
	TIMER=TIME	DEREQ 58
	IF(LT.LT.O.CDO)GO TO 2	DEREQ 59

C		DEREQ 60
C	SUBROUTINE PRELIM DOES PRELIMINARY CALCULATIONS OF GEOMETRIC,	DEREQ 61
C	FLOW FIELD, AND THRUSTING PARAMETERS.	DEREQ 62
10	CONTINUE	DEREQ 63
	CALL PRELIM(LL)	DEREQ 64
	IF(LL.EQ.6)RETURN	DEREQ 65
C		DEREQ 66
C	PROGRAM WILL NOT RUN WHEN THE FREE STREAM MACH NUMBER IS	DEREQ 67
C	LESS THAN 5.	DEREQ 68
	IF(MINF.GE.5)DC1GO TO 20	DEREQ 69
	WRITE(6,1001)	DEREQ 70
1001	FORMAT(1H0 1CX,15MINF IS LESS THAN 5)	DEREQ 71
	LL=6	DEREQ 72
	RETURN	DEREQ 73
C		DEREQ 74
C	PROGRAM WILL NOT RUN WHEN THE BLUNTNES RATIO IS GREATER	DEREQ 75
C	THAN 0.6.	DEREQ 76
20	IF(LAMDA.LE.0.6)DC1GO TO 22	DEREQ 77
	WRITE(6,1002)	DEREQ 78
1002	FORMAT(1H0 10X,25HLAMDA IS GREATER THAN 0.6)	DEREQ 79
	LL=6	DEREQ 80
	RETURN	DEREQ 81
22	CONTINUE	DEREQ 82
C		DEREQ 83
C	NO HEATING OR MASS LOSS CALCULATIONS IN RAREFIED FLOW.	DEREQ 84
C	HEATING AND MASS LOSS CALCULATIONS ARE ONLY PERFORMED IN	DEREQ 85
C	CONTINUUM FLOW AND IN THE FAIRING REGIME BETWEEN STRUNG	DEREQ 86
C	INTERACTION AND CONTINUUM WHERE XBAR IS LESS THAN XUP.	DEREQ 87
	IF(XBAR.LT.XUP)GO TO 3	DEREQ 88
8	WSTAG=TWO	DEREQ 89
	DERIV(5)=0.00	DEREQ 90
	DERIV(6)=0.00	DEREQ 91
	DERIV(7)=0.00	DEREQ 92
	MDOT(2,8)=0.00	DEREQ 93
	MDOT(3,8)=0.00	DEREQ 94
	GO TO 4	DEREQ 95
C		DEREQ 96
C	IF MOPT=0 THE ENTIRE MASS LOSS AND HEATING BLOCK OF EQUATIONS	DEREQ 97
C	IS BYPASSED.	DEREQ 98
3	IF(MOPT.EQ.0)GO TO 6	DEREQ 99
C		DEREQ 100
C	SUBROUTINE AERODY PERFORMS AERO. HEATING CALCULATIONS WHEN	DEREQ 101
C	MOPT = 1.	DEREQ 102
	CALL AERODY	DEREQ 103
	IF(MHEAT.EQ.0)GO TO 8	DEREQ 104
C		DEREQ 105
C	SUBROUTINE MASSLO CONTROLS THE CALLING OF SUBROUTINE EVIL	DEREQ 106
C	WHICH CALCULATES MASS LOSS RATES, RECESSION RATES, AND WALL	DEREQ 107
C	TEMPERATURE. TOMALO INTEGRATES THE MASS LOSS RATES OVER THE	DEREQ 108
C	VEHICLE TO OBTAIN THE RATE OF CHANGE IN WEIGHT DUE TO ABLATION.	DEREQ 109
C	THE VALUE OF THE FOLLOWING DERIVATIVE IS CALCULATED IN TOMALO	DEREQ 110
C	DERIV(5)=W DOT	DEREQ 111
	CALL MASSLO	DEREQ 112
5	CALL TOMALO	DEREQ 113
	DERIV(5)=WDOT	DEREQ 114
C		DEREQ 115
C	SUBROUTINE NOSEBL CALCULATES VEHICLE SHAPE CHANGE WHEN	DEREQ 116
C	NOSTOP = 1. THE VALUES OF THE FOLLOWING DERIVATIVES ARE	DEREQ 117
C	CALCULATED IN NOSEBL.	DEREQ 118
6	IF(NOSFUP.EQ.1)GO TO 7	DEREQ 119

DERIV(2)=C.D0	DEREQ120
DERIV(7)=C.D0	DEREQ121
GO TO 4	DEREQ122
7 CALL NOSEBL	DEREQ123
DERIV(6)=RNDOT	DEREQ124
DERIV(7)=RBDOT	DEREQ125
C	DEREQ126
C	INALPH= THE NUMBER OF VALUES IN THE INPUT ANGLE OF ATTACK
C	TABLE -- WHEN GREATER THAN ZERO, CAUSES THE BYPASSING OF SUB-
C	ROUTINE ROTATE.
C	SUBROUTINE ROTATE CALCULATES THE ANGLE OF ATTACK BY ONE OF
C	TWO METHODS DEPENDING ON THE VALUE OF THE INPUT LCPT.
C	LCPT = 0 IS AN UNCOUPLED THREE DEGREE OF FREEDOM CALCULATION.
C	LCPT = 2 IS A SIMPLIFIED ANGLE OF ATTACK SOLUTION REQUIRING NO
C	INTEGRATION.
4 CONTINUE	DEREQ135
IF(INALPH.GT.0.CDC) GO TO 17	DEREQ136
C	DEREQ137
C	ANGLE OF ATTACK IS CALCULATED BY SUBROUTINE ROTATE.
CALL ROTATE(DERIV,LL)	DEREQ138
IF(ILL.EQ.6)RETURN	DEREQ139
GO TO 15	DEREQ141
17 CONTINUE	DEREQ142
C	THE ANGLE OF ATTACK IS A TABULAR INPUT
LLL=-1	DEREQ144
ALPHA=TABLE(Z,HTAB,ALPTAB,INALPH,LLL)/57.2957RDO	DEREQ145
ALPRIM=ALPHA	DEREQ146
DO 18 IJ = 8,13	DEREQ147
DERIV(IJ) = 0.000	DEREQ148
18 CONTINUE	DEREQ149
19 CONTINUE	DEREQ150
C	DEREQ151
C	SUBROUTINE DRAGCO PERFORMS CALCULATIONS FOR DRAG COEFFICIENTS
C	MAXCD=NUMBER OF VALUES IN THE INPUT DRAG COEFFICIENT TABLE,
C	CDTAB, IF GREATER THAN ZERO, THE DRAG COEFFICIENT IS READ IN
C	AND SUBROUTINE DRAGCO IS BYPASSED.
IF(MAXCD.NE.0)GO TO 13	DEREQ156
CALL DRAGCO	DEREQ157
GO TO 14	DEREQ158
13 LLL=-1	DEREQ159
CD=TABLE(Z,HTAB,CDTAB,MAXCD,LLL)	DEREQ160
14 CONTINUE	DEREQ161
C	DEREQ162
C	MAXMCD=NUMBER OF VALUES IN TABLE OF DRAG COEFF. INCREMENT,
C	MCDTAB, WHICH IS ADDED TO THE CD OBTAINED FROM EITHER DRAGCO OR
C	INPUT CDTAB. MCDTAB IS BASED ON THE INPUT CONSTANT REFERENCE
C	AREA AREF, BUT IS MULTIPLIED BY THE RATIO AAREF/AREF BEFORE
C	BEING ADDED TO CD.
IF(MAXMCD.EQ.0)GO TO 15	DEREQ168
LLL=-1	DEREQ169
MIRECD=TABLE(Z,MHTAB,MCDTAB,MAXMCD,LLL)	DEREQ170
CD=CDCMIRECD*AREF/AREF	DEREQ171
C	DEREQ172
C	SUBROUTINE TEQUAT CALCULATES THE TRANSLATIONAL TRAJECTORY
C	DERIVATIVES.
15 CONTINUE	DEREQ173
CALL TEQUAT(DERIV)	DEREQ176
C	DEREQ177
C	THE TIME DEPENDENT DERIVATIVES ARE BEING CHANGED TO ALTITUDE
C	DEPENDENT DERIVATIVES; DERIV(3) = D(TIME)/D(ALTITUDE).
	DEREQ178
	DEREQ179

```
DERIV(1)=DERIV(1)*DERIV(3) DEREQ180
DERIV(2)=DERIV(2)*DERIV(3) DEREQ181
DO 16 J=4,16 DEREQ182
16 DERIV(J)=DERIV(J)*DERIV(3) DEREQ183
RETURN DEREQ184
2 LL=6 DEREQ185
WRITE(6,1000)7 DEREQ186
1000 FORMAT(1H0,10X,14HZ IS NEGATIVE (E12.5,1H)) DEREQ187
RETURN DEREQ188
END DEREQ189
```

Code	Statement	Line Number
	CDIVML	
	IMPLICIT REAL*8 (A-H,O-Z)	DIVML 1
	SUBROUTINE DIVMLT(A,B,C,D,E,F)	DIVML 2
C		DIVML 3
C	SUBROUTINE DIVMLT CALCULATES THE REAL AND IMAGINARY COMP-	DIVML 4
C	ONENTS OF THE FIRST ORDER NEUMANN FUNCTION USING THE ZEROth	DIVML 5
C	ORDER NEUMANN FUNCTION COMPONENTS.	DIVML 6
C		DIVML 7
	R=C**2&D**2	DIVML 8
	IF (R)1,2,1	DIVML 9
2	E=0.000	DIVML 10
	F=0.000	DIVML 11
	GO TO 3	DIVML 12
C		DIVML 13
1	CONTINUE	DIVML 14
	G=C/R	DIVML 15
	H=-D/R	DIVML 16
	E=A*G-B*H	DIVML 17
	F=A*H&B*G	DIVML 18
3	RETURN	DIVML 19
	END	DIVML 20
		DIVML 21

CORAGCO	DRAGC 1
IMPLICIT REAL*8 (A-H,O-Z)	DRAGC 2
SUBROUTINE DRAGCO	DRAGC 3
C	DRAGC 4
REAL*8 LAMDA,LAMND,LA,MINF,MDOT,ME,MLSTAR,LAMW	DRAGC 5
C	DRAGC 6
DIMENSION A(514),B(21),CDFINF(2,2,2),TW(4,8),MDOT(4,8),	DRAGC 7
1 OCCUR(4000),NOCCUR(30)	DRAGC 8
C	DRAGC 9
EQUIVALENCE	DRAGC 10
1(OCCUR(001),AREF),(OCCUR(002),ALPHA),(OCCUR(003),ALPRIM),	DRAGC 11
2(OCCUR(009),COURAG),(OCCUR(010),CAPL),(OCCUR(011),COSLAM),	DRAGC 12
3(OCCUR(016),CD),(OCCUR(017),CPE),(OCCUR(018),CPW),	DRAGC 13
4(OCCUR(021),D),(OCCUR(026),GAMF),(OCCUR(028),GAMMA),	DRAGC 14
5(OCCUR(029),HSRTO),(OCCUR(032),LAMDA),(OCCUR(033),LA),	DRAGC 15
6(OCCUR(035),MINF),(OCCUR(036),ME),(OCCUR(042),PI),	DRAGC 16
7(OCCUR(048),PE),(OCCUR(049),PINF)	DRAGC 17
EQUIVALENCE	DRAGC 18
1(OCCUR(055),RHOINF),(OCCUR(056),RHOINI),(OCCUR(061),RHOE),	DRAGC 19
2(OCCUR(062),REYL),(OCCUR(064),SINT),(OCCUR(067),SINTM),	DRAGC 20
3(OCCUR(069),THETAD),(OCCUR(070),TANT),(OCCUR(072),TINF),	DRAGC 21
4(OCCUR(074),TWO),(OCCUR(075),T),(OCCUR(077),TCRIT),	DRAGC 22
5(OCCUR(078),TECON),(OCCUR(079),TE),(OCCUR(080),TIMER)	DRAGC 23
EQUIVALENCE	DRAGC 24
1(OCCUR(082),V),(OCCUR(083),VE),(OCCUR(086),WDOT),	DRAGC 25
2(OCCUR(089),XBARST),(OCCUR(090),XBAR),(OCCUR(091),Z),	DRAGC 26
3(OCCUR(092),ZTR),(OCCUR(093),ZETA),(OCCUR(095),GRATE),	DRAGC 27
4(OCCUR(098),CDP),(OCCUR(099),COB),(OCCUR(100),COI),	DRAGC 28
5(OCCUR(101),CPO),(OCCUR(126),XBAR1),(OCCUR(148),TWST),	DRAGC 29
6(OCCUR(197),FACTR9),(OCCUR(225),HWBAR)	DRAGC 30
EQUIVALENCE	DRAGC 31
1(OCCUR(234),DELCDP),(OCCUR(235),DECDFP),(OCCUR(236),DECFTC),	DRAGC 32
2(OCCUR(237),XUP),(OCCUR(238),XLOW),(OCCUR(239),X1UP),	DRAGC 33
3(OCCUR(240),X1LOW),(OCCUR(246),TZTEST)	DRAGC 34
EQUIVALENCE	DRAGC 35
1(OCCUR(0301), A(1)),(OCCUR(0823), B(1)),	DRAGC 36
2(OCCUR(2644), TW(1,1)),(OCCUR(2708),MDOT(1,1)),	DRAGC 37
3(OCCUR(2793),CDFINF(1,1,1))	DRAGC 38
EQUIVALENCE	DRAGC 39
1 (NOCCUR(03), MUPT),(NOCCUR(07), LCPT),(NOCCUR(10), MHEAT)	DRAGC 40
C	DRAGC 41
COMMON OCCUR,NOCCUR	DRAGC 42
C	DRAGC 43
C	DRAGC 44
C	DRAGC 45
C SUBROUTINE DRAGCO CALCULATES THE DRAG COEFFICIENT WHEN INPUT	DRAGC 46
C DRAG COEFFICIENT OPTION IS NOT USED.	DRAGC 47
C	DRAGC 48
C FREE MOLECULE FLOW REGIME. XBARI IS GREATER THAN OR EQUAL	DRAGC 49
C TO X1UP.	DRAGC 50
C STRONG INTERACTION FLW REGIME. XBARI IS LESS THAN X1LOW	DRAGC 51
C AND XBAR GREATER THAN OR EQUAL TO XUP.	DRAGC 52
C FAIRING REGIME BETWEEN FREE MOLECULE AND STRONG INTERACTION.	DRAGC 53
C XBARI IS LESS THAN X1UP AND GREATER THAN OR EQUAL TO X1LOW.	DRAGC 54
C	DRAGC 55
C	DRAGC 56
PEPINF=PE/PINF	DRAGC 57
SURT3=1.73205080756887700	DRAGC 58
TEMLAM=1.000-COSLAM*COSLAM	DRAGC 59

.....	FIK=.9D06SINTM*(-.119D06.0108D0*SINTM)	DRAGC 60
.....	ALPHA0=180.000*ALPHA/PI	DRAGC 61
C	DRAGC 62
C	DRAGC 63
C	DRAGC 64
C	DRAGC 65
.....	N=312	DRAGC 66
.....	IF(THETAD.GT.20.000)GO TO 200	DRAGC 67
.....	N=247	DRAGC 68
.....	IF(THETAD.GT.10.000)GO TO 200	DRAGC 69
.....	N=211	DRAGC 70
.....	200 X=1.000/MINF	DRAGC 71
.....	ZZ=1.000/THETAD	DRAGC 72
.....	Y=LAMDA	DRAGC 73
.....	CDPO=0.000	DRAGC 74
.....	DO 201 I=1,3	DRAGC 75
.....	II=I-1	DRAGC 76
.....	ISUB1=NGII	DRAGC 77
.....	TEMI=X**II	DRAGC 78
.....	DO 201 J=1,3	DRAGC 79
.....	JJ=J-1	DRAGC 80
.....	ISUB2=ISUB1&3*JJ	DRAGC 81
.....	TEMJ=1.000	DRAGC 82
.....	IF(JJ.NE.0)TEMJ=Y**JJ	DRAGC 83
.....	DO 201 K=1,4	DRAGC 84
.....	KK=K-1	DRAGC 85
.....	201 CDPO=CDPO&A((ISUB2&9*KK)*TEMJ*TEMI*ZZ**KK	DRAGC 86
C	DRAGC 87
C	DRAGC 88
C	DRAGC 89
.....	CDPLEO=2.000*PE/(RHOINF*V*V)	DRAGC 90
C	DRAGC 91
C	DRAGC 92
C	DRAGC 93
C	DRAGC 94
.....	IF(DABS(ALPHA0).GT.4.000)GO TO 220	DRAGC 95
.....	COLCDO=1.000	DRAGC 96
.....	IF(ALPHA0.EQ.0.000)GO TO 230	DRAGC 97
.....	X=THETAD	DRAGC 98
.....	Y=DABS(ALPHA0)	DRAGC 99
.....	ZZ=LAMDA	DRAGC 100
.....	COLCDO=0.000	DRAGC 101
.....	DO 202 I=1,4	DRAGC 102
.....	II=I-1	DRAGC 103
.....	ISUB1=42I&II	DRAGC 104
.....	TEMI=X**II	DRAGC 105
.....	DO 202 J=1,3	DRAGC 106
.....	JJ=J-1	DRAGC 107
.....	ISUB2=ISUB1&4*JJ	DRAGC 108
.....	TEMJ=Y**JJ	DRAGC 109
.....	DO 202 K=1,3	DRAGC 110
.....	KK=K-1	DRAGC 111
.....	TEMK=1.000	DRAGC 112
.....	IF(KK.NE.0)TEMK=ZZ**KK	DRAGC 113
.....	202 COLCDO=COLCDO&A((ISUB2&12*KK)*TEMI*TEMJ*TEMK	DRAGC 114
.....	IF(COLCDO.LT.1.000)COLCDO=1.000	DRAGC 115
.....	GO TO 230	DRAGC 116
C	DRAGC 117
C	DRAGC 118
C	DRAGC 119

```

220-TEML=DABS(ALPHAD) DRAGC120
C IF ANGLE OF ATTACK IS GREATER THAN 40 DEGREES THEN ALPHA OF DRAGC121
C 40 DEGREES IS USED IN CURVE FIT. DRAGC122
IF(TEML.GT.40.000)TEML=40.000 DRAGC123
X=DLOG10(THETAD) DRAGC124
Y=DLOG10(TEML) DRAGC125
ZZ=LAMDA DRAGC126
FUN=0.000 DRAGC127
DO 222 I=1,4 DRAGC128
  II=I-1 DRAGC129
  ISUB1=348&II DRAGC130
  TEMI=X**II DRAGC131
  DO 222 J=1,3 DRAGC132
    JJ=J-1 DRAGC133
    ISUB2=ISUB1&4*JJ DRAGC134
    TEMJ=Y**JJ DRAGC135
    DO 222 K=1,3 DRAGC136
      KK=K-1 DRAGC137
      TEMK=1.000 DRAGC138
      IF(KK.NE.0)TEMK=ZZ**KK DRAGC139
222 FUN=FUN&A((ISUB2&12*KK)*TEMI*TEMJ*TEMK DRAGC140
      IF(FUN.LT.0.000)FUN=0.000 DRAGC141
      CDLCDO=10.000**FUN DRAGC142
230 CONTINUE DRAGC143
C DRAGC144
C CDP=CDPO*CDLCDO DRAGC145
C DRAGC146
C IF(XBAR.LT.XLOW)GO TO 1 DRAGC147
C WHEN XBAR HAS VALUE LESS THAN XLOW, ALL RAREFIED AIR DRAGC148
C SOLUTIONS ARE BYPASSED AND ONLY CONTINUUM CALCULATIONS ARE DRAGC149
C PERFORMED. DRAGC150
C DRAGC151
C TEMTSG=T(INF*(1.00&0.500*(GAMMA-1.00)*M(INF*M(INF) DRAGC152
  TEM = TWST/TEMTSG DRAGC153
  TEM3=DLOG(XBARST) DRAGC154
  SUM=0.000 DRAGC155
  IF(THETAD.LT.15.000)GO TO 204 DRAGC156
C DRAGC157
C SHARP CONE STRONG INTERACTION DRAG COEFF. FOR CONE HALF ANGLE DRAGC158
C GREATER THAN OR EQUAL TO 15 DEGREES. DRAGC159
DO 203 I=1,2 DRAGC160
  II=I-1 DRAGC161
  ISUB1=384&II DRAGC162
  TEMI=TEM3**II DRAGC163
  DO 203 J=1,4 DRAGC164
    JJ=J-1 DRAGC165
    ISUB2=ISUB1&2*JJ DRAGC166
    TEM2=THETAD**JJ DRAGC167
    DO 203 K=1,2 DRAGC168
      KK=K-1 DRAGC169
      ISUB=ISUB2&8*KK DRAGC170
203 SUM=SUM&A((ISUB)*TEMI*TEM2*TEM**KK DRAGC171
      COST=DEXP(SUM) DRAGC172
      GO TO 3 DRAGC173
C DRAGC174
C SHARP CONE STRONG INTERACTION DRAG COEFF. FOR CONE HALF ANGLE DRAGC175
C LESS THAN 15 DEGREES. DRAGC176
204 CONTINUE DRAGC177
DO 2 I=1,2 DRAGC178
  II=I-1 DRAGC179

```

```

ISUB1=200&I1-----DRAGC180
TEM1=TEM3**I1-----DRAGC181
DO 2 J=1,2-----DRAGC182
JJ=J-1-----DRAGC183
ISUB2=ISUB1&2*JJ-----DRAGC184
TEM2=THETAD**JJ-----DRAGC185
DO 2 K=1,2-----DRAGC186
KK=K-1-----DRAGC187
ISUB=ISUB2&4*KK-----DRAGC188
2 SUM=SUMGA(ISUB)*TEM1*TEM2*TEM**KK-----DRAGC189
CDST=EXP(SUM)-----DRAGC190
C-----DRAGC191
C FREE MOLECULE DRAG COEFFICIENT WITH ACCOMMODATION COEFFICIENT-----DRAGC192
C ZETA-----DRAGC193
C-----DRAGC194
3 CDF=EXP(-SINTM*SINTM)-----DRAGC195
CDF=CDF*(1.00/(SINTM*SQRT(PI)))&0.500*SQRT(TWST/TINF)/(MINF*MINF)-----DRAGC196
CDF=CDF&2.00&1.00/(MINF*MINF)&SINT*SQRT(PI*TWST/TINF)/MINF-----DRAGC197
CDF=ZETA*CDF-----DRAGC198
C-----DRAGC199
C NEWTONIAN DRAG COEFFICIENT-----DRAGC200
CDN=2.00*SINT*SINT-----DRAGC201
C-----DRAGC202
C CAPP IS THE SHARP CONE PROBABILITY DISTRIBUTION AS A FUNCTION-----DRAGC202
C OF KNUDSEN NUMBER-----DRAGC203
LAMW=CDRAG*SQRT(2.2500*PI*TWST/(GAMMA*TINF))/MINF-----DRAGC204
LAMWD=LAMW/D-----DRAGC205
IF(LAMWD.LT.0.0400)GO TO 19-----DRAGC206
CAPP=0.00-----DRAGC207
TEM=DLOG(LAMW/D)-----DRAGC208
DO 18 I=1,21-----DRAGC209
18 CAPP=CAPP&0(I)*TEM**(I-1)-----DRAGC210
GO TO 23-----DRAGC211
19 CAPP=0.50600-0.14700*(DLOG10(0.0400/LAMWD))-----DRAGC212
C-----DRAGC213
23 CONTINUE-----DRAGC214
C SHARP CONE FREE MOLECULE DRAG COEFFICIENT-----DRAGC215
CDTRFM=CAPP*(CDF-CDN)&CDN-----DRAGC216
IF(LAMDA.LT.1.00-3160 TO 30-----DRAGC217
C-----DRAGC218
C FREE MOLECULE DRAG COEFF. ON SPHERICAL NOSE-----DRAGC219
CDFMS=0.000-----DRAGC220
TEM=DLOG(REYL*D/(CAPL*MINF*FACTR9))-----DRAGC221
IF(DABS(TEM).LE.1.00-4)TEM=1.00-4-----DRAGC222
IF(TEM.GT.5.29800)TEM=5.29800-----DRAGC223
DO 31 J=1,20-----DRAGC224
31 CDFMS=CDFMS&A(J&173)*TEM**(J-1)-----DRAGC225
CDFMS=2.000*CDFMS-----DRAGC226
C-----DRAGC227
C BLUNT CONE FREE MOLECULE TOTAL DRAG COEFFICIENT-----DRAGC228
CDFN=CDFMS*COSLAM*COSLAM&CDTRFM*(1.000-LAMDA*LAMDA)-----DRAGC229
GO TO 32-----DRAGC230
C-----DRAGC231
C SHARP CONE FREE MOLECULE TOTAL DRAG COEFFICIENT-----DRAGC232
30 CDFM=CDTRFM-----DRAGC233
C-----DRAGC234
C BLUNT CONE STRONG INTERACTION DRAG COEFFICIENT-----DRAGC235
32 CDS=(CDST-COPEO)*(1.000-LAMDA*LAMDA)*CUSI(ALPRI4)&CDP-----DRAGC236
IF(XBARI.LT.XIUPI)GO TO 34-----DRAGC237
C-----DRAGC238
C FREE MOLECULE TOTAL DRAG COEFFICIENT-----DRAGC239

```

CD=CDFM	DRAGC240
RETURN	DRAGC241
C	DRAGC242
34 IF(XBAR1.GE.X1LOW)GO TO 35	DRAGC243
IF(XBAR.LE.XUP)GO TO-5	DRAGC244
C	DRAGC245
STRONG INTERACTION TOTAL DRAG COEFFICIENT.	DRAGC246
CD=CDS	DRAGC247
RETURN	DRAGC248
C	DRAGC249
FAIRING REGION BETWEEN STRONG INTERACTION AND FREE MOLECULE	DRAGC250
C	DRAGC251
FLOW	DRAGC252
35 W1=(XBAR1-X1LOW)/(X1UP-X1LOW)	DRAGC253
CD=CDFM*W1&CDS*(1.000-W1)	DRAGC254
IF(XBAR.GT.XUP)RETURN	DRAGC255
XUP=XBAR	DRAGC256
XLOW=XBAR-2.000	DRAGC257
IF(XBAR.LE.2.000)XLOW=0.500	DRAGC258
RETURN	DRAGC259
C	DRAGC260
C	DRAGC261
CONTINUUM FLOW DRAG CALCULATIONS.	DRAGC262
C	DRAGC263
C	DRAGC264
KEY FOR SKIN FRICTION DRAG COEFF. DESCRIPTION	DRAGC265
C SHARP=1 BLUNT=2	DRAGC266
C TURBULENT=1 LAMINAR=2	DRAGC267
C WITH BLOWING=1 NO BLOWING=2	DRAGC268
C CDFINF(SHARP OR BLUNT, LAM. OR TURB., BLOWING OR NO BLOWING)	DRAGC269
C	DRAGC270
BASE DRAG COEFFICIENT.	DRAGC271
1 CONTINUE	DRAGC272
5 Y=OABS(WOOT)/(RHUIN1*V*AREF)	DRAGC273
IF(Y.LT.0.000)Y=0.000	DRAGC274
IF(Y.GT.0.0600)Y=0.0600	DRAGC275
X=1.000/MINF	DRAGC276
IF(MINF.GT.24.000)X=1.000/24.000	DRAGC277
N=487	DRAGC278
IF(MINF.GE.7.000)N=499	DRAGC279
PBPINF=0.000	DRAGC280
DO 8 I=1,4	DRAGC281
II=I-1	DRAGC282
ISUBI=N&II	DRAGC283
TEMI=X**II	DRAGC284
DO 8 J=1,3	DRAGC285
JJ=J-1	DRAGC286
ISUB=ISUBI&4*JJ	DRAGC287
TEMJ=1.000	DRAGC288
IF(JJ.NE.0)TEMJ=Y**JJ	DRAGC289
8 PBPINF=PBPINF&A(ISUB)*TEMI*TEMJ	DRAGC290
CDB=(1.000-PBPINF)*X**2/0.700	DRAGC291
C	DRAGC292
9 CONTINUE	DRAGC293
IF(XBAR.GE.XLOW)GO TO 10	DRAGC294
IF(Z.GE.ZTR)GO TO 10	DRAGC295
C	DRAGC296
C	DRAGC297
TURBULENT DRAG CALCULATIONS.	DRAGC298
C	DRAGC299
C	DRAGC296
SCHULTZ GRUNOW SKIN FRICTION DRAG COEFFICIENT.	DRAGC297
HE=CPE*TE	DRAGC298

.....	HSTAHE=0.50000.500*CPW*TW(3,8)/HELD.09900*ME*ME*(GAMMA=1.000)	DRAGC300
.....	HSTAR=HSTAHE*HE	DRAGC301
.....	IF(HSTAR.LE.1110.000)GO TO 21	DRAGC302
.....	TEM=PE/2116.000	DRAGC303
.....	IF(TEM.GE.1.001)TEM=1.001	DRAGC304
.....	TSTAR=0.000	DRAGC305
.....	DO 20 I=1,4	DRAGC306
.....	II=I-1	DRAGC307
.....	ISUBI=162&II	DRAGC308
.....	TEMI=HSTAR**II	DRAGC309
.....	DO 20 J=1,3	DRAGC310
.....	JJ=J-1	DRAGC311
.....	ISUD=ISUBI&4*JJ	DRAGC312
.....	20 TSTAR=TSTAR&A(ISUB)*TEMI*(TEM**JJ)	DRAGC313
.....	GO TO 22	DRAGC314
.....	21 TSTAR=3.596400*HSTAR	DRAGC315
.....	22 CONTINUE	DRAGC316
.....	IF(TSTAR.LT.0.00)CALL EXIT	DRAGC317
.....	MUSTAR=32.200*2.270-8*TSTAR**1.500/(TSTAR&198.600)	DRAGC318
.....	TAU=TSTAR*(1.000-0.12500*LOG10(PE/2116.000))/1.800	DRAGC319
.....	ZTAU=2.500&0.100*DTANH(TAU/500.000-7.00)&0.400*DTANH(TAU/1000.00-	DRAGC320
.....	17.00)&DTANH(TAU/2500.00-5.800)	DRAGC321
.....	RHOSTA=(39.6500*PE)/(2116.00*ZTAU*TSTAR)	DRAGC322
.....	C	DRAGC323
.....	IF(CAPL.GE.2.00)GO TO 12	DRAGC324
.....	A(118)=-4.486600	DRAGC325
.....	A(119)=156.00	DRAGC326
.....	A(120)=-665.00	DRAGC327
.....	GO TO 13	DRAGC328
.....	12 A(118)=0.00	DRAGC329
.....	A(119)=0.00	DRAGC330
.....	A(120)=0.00	DRAGC331
.....	DO 14 I=1,7	DRAGC332
.....	II=I-1	DRAGC333
.....	TEMI=CAPL**II	DRAGC334
.....	A(118)=A(118)&A(116400)*TEMI	DRAGC335
.....	A(119)=A(119)&A(116407)*TEMI	DRAGC336
.....	14 A(120)=A(120)&A(116414)*TEMI	DRAGC337
.....	13 TEMI=LOG10(RHOSTA*VE/MUSTAR)	DRAGC338
.....	TEM2=A(120)&TEMI*(A(119)&A(118)*TEMI)	DRAGC339
.....	REYSTA=RHOSTA*VE*CAPL/MUSTAR	DRAGC340
.....	C	DRAGC341
.....	C	DRAGC342
.....	C	DRAGC343
.....	CF0=NO BLOWING SKIN FRICTION COEFF. FOR USE IN MASS LOSS	DRAGC344
.....	CORRECTION	DRAGC345
.....	CF0=.3700*1.1500*RHOSTA*VE*VE/LV*V*RHOIN1*LOG10(REYSTA)**2.58400	DRAGC346
.....	CF0=CF0*RHOIN1*V*V/(RHOE*VE*VE)	DRAGC347
.....	CFINF(1,1,2)=0.500*.85200*VE*VE*RHOSTA/(RHOIN1*TANT*V*V*TEM2)	DRAGC348
.....	C	DRAGC349
.....	C	DRAGC350
.....	BLOWING CORRECTION TO SKIN FRICTION DRAG COEFF.	DRAGC351
.....	COFINF(1,1,1)=COFINF(1,1,2)/(1.00&1.200*MDOT(3,8)/(RHOE*VE*CF0))	DRAGC352
.....	C	DRAGC353
.....	C	DRAGC354
.....	BLUNTNES CORRECTION TO SKIN FRICTION DRAG COEFF.	DRAGC355
.....	IF(LAMDA.LT.0.3200)COFINF(2,1,1)=COFINF(1,1,1)*(1.000-LAMDA*	DRAGC356
.....	1(0.8000&0.05200*MINF))	DRAGC357
.....	IF(LAMDA.GE.0.3200)COFINF(2,1,1)=COFINF(1,1,1)*(3.74400-0.0166400*	DRAGC358
.....	MINF)	DRAGC359
.....	COFINF(2,1,2)=COFINF(2,1,1)*COFINF(1,1,2)/COFINF(1,1,1)	DRAGC357
.....	C	DRAGC358
.....	C	DRAGC359
.....	INDUCED PRESSURE DRAG COEFFICIENT.	DRAGC358
.....	MW=HWBAR	DRAGC359

```

----- IF((MOPT*MHEAT).EQ.0)HW=0.2400*TWO/31.8600 ----- DRAGC350
HR=0.900*HSRTO ----- DRAGC351
NWHK=HW/HR ----- DRAGC352
TEM =RHOE*VE*CF0 ----- DRAGC353
DTHE1=CF0*(2.000*MOOT(3,8)/TEM -&1.000/(1.000&1.200*MOOT(3,8)/TEM)) DRAGC354
1/3.800 ----- DRAGC355
DTHE2=DTHE1*(0.54700*HWHR&ME*(0.5300&0.6800*HWHR)&ME*ME*(0.08300&
10.10&00*HWHR)) ----- DRAGC357
DTHE=DTHE2&1.000*MOOT(3,8)/13.600*RHCE*VE) ----- DRAGC368
CDLCP=1.1100*TEMLAM*CDPLEO*DTHE*FIK*SQRT3*ME ----- DRAGC369
CDI=CDLCP ----- DRAGC370
C ----- DRAGC371
C ----- VALUE OF CDSUM FOR FULLY TURBULENT FLOW ----- DRAGC372
CDSUM=CDFINF(2,1,1)&CDLCP ----- DRAGC373
CDSUM=CDSUM ----- DRAGC374
IF(LOPT.GE.3)GO TO 36 ----- DRAGC375
IF(TIMER.GE.TZTEST)GO TO 36 ----- DRAGC376
C ----- DRAGC377
C ----- FAIRED VALUE FOR CDSUM BETWEEN LAMINAR AND TURBULENT FLOW ----- DRAGC378
TEM1 =CFOZTR*LA/1.500 ----- DRAGC379
TEM2=CFO*LA**1.600/1.800 ----- DRAGC380
TEM3=1.000-(TIMER-TZTR)/TF ----- DRAGC381
PBAR=(TEM1*TEM3**1.500-TEM2*TEM3**1.800)/(TEM1-TEM2) ----- DRAGC382
CDI=CDILAM*PBAR&(1.000-PBAR)*CDI ----- DRAGC383
CDFINF(2,1,1)=CDFINF(2,2,1)*PBAR&(1.000-PBAR)*CDFINF(2,1,1) ----- DRAGC384
CDFINF(2,1,2)=CDFINF(2,2,2)*PBAR&(1.000-PBAR)*CDFINF(2,1,2) ----- DRAGC385
CDSUM=CDI&CDFINF(2,1,1) ----- DRAGC386
C ----- DRAGC387
C ----- AVERAGE ANGLE OF ATTACK OVER CYCLE USED IN CORRECTING CDP. ----- DRAGC388
36 CONTINUE ----- DRAGC389
IF(TCRIT.LE.T)GO TO 15 ----- DRAGC390
IF(LT.TECON)GO TO 15 ----- DRAGC391
CO=CDSUM&CDB&CDPO*(1.000&GRATE/T) ----- DRAGC392
RETURN ----- DRAGC393
C ----- DRAGC394
C ----- TOTAL TURBULENT DRAG COEFFICIENT ----- DRAGC395
15 CONTINUE ----- DRAGC396
CD=CDSUM&CDB&CDP ----- DRAGC397
RETURN ----- DRAGC398
C ----- DRAGC399
C ----- DRAGC400
C ----- LAMINAR DRAG CALCULATIONS. ----- DRAGC401
10-HSTAHE=0.500&0.500*CPW*TW(2,8)/(CPE*TE)&0.093500*(GAMMA-1.000)*ME*UAGL402
IME ----- DRAGC403
REYSTA=REYL ----- DRAGC404
CDFINF(1,2,2)=1.5300*SQRT(PEPINF/REYL)*(VE/V)**1.500/TANT ----- DRAGC405
CDFINF(1,2,2)=CDFINF(1,2,2)*(HSTAHE*CPE*TE/(.239800*TINF))**(-.185)DRAGC406
XDO) ----- DRAGC407
C ----- CFO= NO BLOWING SKIN FRICTION COEFF. FOR USE IN MASS LOSS ----- DRAGC408
C ----- CORRECTION. ----- DRAGC409
CFO=1.1500*TANT*CDFINF(1,2,2)*RHQIN1*V*V/(1.5300*RHOE*VE*VE) ----- DRAGC410
C ----- DRAGC411
C ----- BLOWING CORRECTION TO SKIN FRICTION DRAG COEFF. ----- DRAGC412
CDFINF(1,2,1)=CDFINF(1,2,2)/(1.000&2.0000*MOOT(2,8)/(RHOE*VE*CF0)) DRAGC413
C ----- DRAGC414
C ----- BLUNTNESS CORRECTION TO SKIN FRICTION DRAG COEFF. ----- DRAGC415
TEM=OLOG10(REYL) ----- DRAGC416
IF(LAMDA.GE.0.200)N=472 ----- DRAGC417
IF(LAMDA.LT.0.200)N=457 ----- DRAGC418
SUM=0.00 ----- DRAGC419

```

DO 16 J=1,3	DRAGC420
JJ=J-1	DRAGC421
ISUB1=N&5*JJ	DRAGC422
TEMJ=1.00	DRAGC423
IF(JJ.NE.0)TEMJ=LAMDA**JJ	DRAGC424
DO 16 I=1,5	DRAGC425
II=I-1	DRAGC426
ISUB=ISUB1&II	DRAGC427
16 SUM=SUM&A(ISUB)*TEMJ*TEM**II	DRAGC428
CDFINF(2,2,1)=CDFINF(1,2,1)*SUM	DRAGC429
CDFINF(2,2,2)=CDFINF(2,2,1)*CDFINF(1,2,2)/CDFINF(1,2,1)	DRAGC430
C	DRAGC431
C DELCDP,DECDFF, AND DECFYC ARE RESPECTIVELY THE LAMINAR	DRAGC432
C INDUCED PRESSURE DRAG COEFFICIENT, PRESSURE INDUCED AND TRANS-	DRAGC433
C VERSE CURVATURE INDUCED SKIN FRICTION DRAG COEFFICIENTS.	DRAGC434
C	DRAGC435
HW=HWBAR	DRAGC436
IF(MOPT*MHEAT).EQ.0)HW=0.2400*TW(2,8)/33.8600	DRAGC437
HR=0.900*HSRTO	DRAGC438
TEM =RHDE*VE*CF0	DRAGC439
DTHE1=CF0*(2.000*MOOT(2,8)/TEM&1.000/(1.000&1.2500*MOOT(2,8)/TEM)	DRAGC440
1/(2.000*SQRT3)	DRAGC441
FME2=ME*ME	DRAGC442
HWHR=HW/HR	DRAGC443
DTHE2=DTHE1*(A(283)&A(284)*FME2&HWHR*(A(285)&A(286)*FME2&HWHR*(DRAGC444
1A(287)&A(288)*FME2&HWHR*(A(289)&A(290)*FME2)))	DRAGC445
DTHE=DTHE2&MOOT(2,8)/(3.000*RHDE*VE)	DRAGC446
DELCDP=1.3300*TEMLAM*CDPLEO*DTHE*F1K*SQRT3*ME	DRAGC447
UECDFF=1.500*DTHE*TEMLAM *CDFINF(1,2,2)*F1K	DRAGC448
1*(-0.82300&0.52400*TW(2,8)/TE&0.438*FME2)/(ME*SQRT(CAPL))	DRAGC449
DINF=D.05800&0.968*TW(2,8)/(TINF*MINF*MINF)	DRAGC450
DECFYC=1.500*(0.51700&0.91300*TW(2,8)/TE&0.048400*FME2)	DRAGC451
1*TEMLAM*DTHE*CDFINF(1,2,2)/(FME2*TANT*DINF*SQRT(3.000*CAPL))	DRAGC452
CDI=DELCDP&UECDFF&DECFYC	DRAGC453
C	DRAGC454
IF(LDPT.GE.3)GO TO 4	DRAGC455
C	DRAGC456
C DEFINING QUANTITIES USED IN FAIRING BETWEEN LAMINAR AND	DRAGC457
C TURBULENT FLOW REGIMES.	DRAGC458
C	DRAGC459
CF0ZTR=CF0	DRAGC460
TZTR=TIMER	DRAGC461
TF=1.000/(0.486670-4*V*DABS(SIN(GAMF)))	DRAGC462
CDILAM=CDI	DRAGC463
TZTEST=TZTR&TF	DRAGC464
4 CONTINUE	DRAGC465
C	DRAGC466
IF(XBAR.GE.XLOW)GO TO 33	DRAGC467
IF(TGRIT.LE.T)GO TO 17	DRAGC468
IF(T.LT.TECON)GO TO 17	DRAGC469
C	DRAGC470
C AVERAGE ANGLE OF ATTACK OVER CYCLE USED IN CORRECTING CDP.	DRAGC471
C CD=COB&CDI&CDFINF(2,2,1)&COPO*(1.00&GRATE/T)	DRAGC472
RETURN	DRAGC473
C	DRAGC474
C FULLY LAMINAR FLOW.	DRAGC475
C	DRAGC476
17 CD=COB&CDI&CDFINF(2,2,1)&CDP	DRAGC477
RETURN	DRAGC478
C	DRAGC479
C FAIRING BETWEEN STRONG INTERACTION AND LAMINAR FLOW.	
C	
33 CDLAM=COB&CDI&CDP&CDFINF(2,2,1)	
W2=(XBAR-XLOW)/(XUP-XLOW)	

CD=CDS*W2&CDLAW*(L,ODO-W2)
RETURN
END

DRAGC480
DRAGC481
DRAGC482

C	DRESS		DRESS	1
		SUBROUTINE DRESS	DRESS	2
		IMPLICIT REAL*8 (A-H,O-Z)	DRESS	3
		COMMON/BLK0/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)	DRESS	4
		1,G,S,GSP,GTP,GSS,GTI,GSB,F,FP,FB,FG,E,P,TD,RS,SL,Z,Q,A,EL,	DRESS	5
		ZDELTA,FAC,C(40,10)	DRESS	6
		COMMON/SENSE/ISEN1,ISEN2	DRESS	7
		COMMON/BLK1/M,N,L,LS,ML,MS,NS,IT,K,NC	DRESS	8
C			DRESS	9
C			DRESS	10
C		SUBROUTINE DRESS, PART OF THE DAVIDON MINIMIZATION METHOD,	DRESS	11
C		MODIFIES THE METRIC MATRIX H ON THE BASIS OF INFORMATION	DRESS	12
C		OBTAINED ABOUT THE FUNCTION ALONG THE DIRECTION UNDER CONSIDER-	DRESS	13
C		ATION, S. THE NEW H MUST RETAIN THE INFORMATION WHICH THE	DRESS	14
C		PRECEDING ITERATIONS HAVE GIVEN ABOUT THE FUNCTION.	DRESS	15
C			DRESS	16
C			DRESS	17
		L=L	DRESS	18
		GO TO (500,525,519,510),L	DRESS	19
	500	CALL MATMP (N,N,H,G,X)	DRESS	20
	501	M=1	DRESS	21
	502	CALL MATMP (M,N,X,G,TO)	DRESS	22
	503	TP1=SL-GSS*2/TO-E	DRESS	23
	504	IF (TP1)524,505,505	DRESS	24
	505	DO 507 I=1,N	DRESS	25
	506	DO 507 J=1,N	DRESS	26
	507	H(I,J)=H(I,J)-X(I)*X(J)/TO	DRESS	27
	508	DELTA=DELTA*(EL+GSS/TO)	DRESS	28
	509	TO=EL/GSS	DRESS	29
			DRESS	30
C			DRESS	31
C		510 DO 512 I=1,N	DRESS	32
	511	DO 512 J=1,N	DRESS	33
	512	H(I,J)=H(I,J)&TO*S(I)*S(J)	DRESS	34
			DRESS	35
C			DRESS	36
C			DRESS	37
	519	F=FB	DRESS	38
	520	DO 522 I=1,N	DRESS	39
	521	G(I)=GB(I)	DRESS	40
	522	X(I)=T(I)	DRESS	41
	529	CALL OVERFL(LCP021)	DRESS	42
		GO TO (531,530,530),LCP021	DRESS	43
	530	CALL UVEKFL(LCP021)	DRESS	44
		GO TO (531,513,513),LCP021	DRESS	45
	513	WRITE (6,1)IT,MS,F,GS	DRESS	46
	514	IF(ISEN1) 513,515, 517	DRESS	47
	515	WRITE (6,2)(X(I),I=1,N)	DRESS	48
		WRITE(6,7)	DRESS	49
		DO 101 I = 1,N	DRESS	50
	101	WRITE(6,8) (H(I,J),J=1,N)	DRESS	51
	516	WRITE (6,3)DELTA	DRESS	52
	517	WRITE (6,4)	DRESS	53
	518	IT=IT&1	DRESS	54
	5165	L=L	DRESS	55
C			DRESS	56
C		523 RETURN	DRESS	57
C			DRESS	58
C		524 WRITE (6,5)	DRESS	59
C			DRESS	59

525	TP1=EL*SL/GSS	DRESS 60
526	DELTA=DELTA*TP1	DRESS 61
527	TU=(TP1-1.0)/SL	DRESS 62
528	GO TO 510	DRESS 63
C		DRESS 64
531	WRITE (6,6)	DRESS 65
532	WRITE (6,7)	DRESS 66
533	DO 534 I=1,N	DRESS 67
534	WRITE (6,8)(H(I,J),J=1,N)	DRESS 68
535	WRITE (6,9)F	DRESS 69
536	WRITE (6,2)(X(I),I=1,N)	DRESS 70
537	WRITE (6,10)(G(I),I=1,N)	DRESS 71
	IF(ISEN1) 539,539, 543	DRESS 72
539	CONTINUE	DRESS 73
543	L=2	DRESS 74
544	GO TO 523	DRESS 75
C		DRESS 76
1	FORMAT (4HOIT 14,7H STEP 14,4H F=1PE14.5,5H GS=E14.5)	DRESS 77
2	FORMAT (3HOX=1P8E14.5/(3HO 8E14.5))	DRESS 78
3	FORMAT (7HODELTA=1PE14.5)	DRESS 79
4	FORMAT (20HO- - - - -)	DRESS 80
5	FORMAT (9HOCOLINEAR)	DRESS 81
6	FORMAT (9HOOVERFLOW)	DRESS 82
7	FORMAT (13HOERROR MATRIX)	DRESS 83
8	FORMAT (1HO1P8E14.5/(1HO8E14.5))	DRESS 84
9	FORMAT (3HOF=1PE14.5)	DRESS 85
10	FORMAT (3HOG=1P8E14.5/(3HO 8E14.5))	DRESS 86
11	FORMAT (6E12.5)	DRESS 87
	END	DRESS 88

CDRLIM	REAL * 8 R(2) A(2)	DRLIM 1
	COMPLEX *16 C, C2	DRLIM 2
	SUBROUTINE DRLIM(C, R)	DRLIM 3
	EQUIVALENCE (C2, A)	DRLIM 4
C		DRLIM 5
C	SUBROUTINE DRLIM TAKES THE REAL AND IMAGINARY PARTS OF A	DRLIM 6
C	COMPLEX NUMBER AND SETS THEM EQUAL TO TWO REAL NUMBERS.	DRLIM 7
C		DRLIM 8
	C2 = C	DRLIM 9
	R (1) = A(1)	DRLIM 10
	R (2) = A(2)	DRLIM 11
	RETURN	DRLIM 12
	END	DRLIM 13
		DRLIM 14

CEFFECT	EFFEC	1
CEFFECT JUNE-3	EFFEC	2
SUBROUTINE EFFECT	EFFEC	3
IMPLICIT REAL*8(A-H,O-Z)	EFFEC	4
COMMON OCCUR,NOCCUR	EFFEC	5
COMMON /CICCUR/ IOCCUR(320)	EFFEC	6
COMMON /CPCUR/ PCCUR(11770)	EFFEC	7
DIMENSION IOP(90), NOCCUR(30), OCCUR(4000)	EFFEC	8
DIMENSION SRS(9)	EFFEC	9
DIMENSION DUMMY(100)	EFFEC	10
EQUIVALENCE (IOCCUR(00001), IOP)	EFFEC	11
EQUIVALENCE (PCCUR(11757), PFD)	EFFEC	12
EQUIVALENCE (PCCUR(11758), HDIF)	EFFEC	13
EQUIVALENCE (OCCUR(3962), PD)	EFFEC	14
EQUIVALENCE (PCCUR(11598), SRS)	EFFEC	15
EQUIVALENCE (OCCUR(3901), DUMMY)	EFFEC	16
EQUIVALENCE (OCCUR(03965), SIGMA)	EFFEC	17
DATA A, CON, EPS, B / .7071067811865475, .0500, .0000100,	EFFEC	18
* .0227501319481791 /	EFFEC	19
C	EFFEC	20
C	EFFEC	21
C	EFFEC	22
C	EFFEC	23
C	EFFEC	24
C	EFFEC	25
C	EFFEC	26
PD = 0.00	EFFEC	27
C	EFFEC	28
DO 11 I = 1, 9	EFFEC	29
IF (I.GT.3) GO TO 5	EFFEC	30
IF(IOP(I&3).EQ.1)PD=PD&SRS(I)*OCCUR(I&3952)	EFFEC	31
GO TO 11	EFFEC	32
5 IF(IOP(I&24).EQ.1) PD = PD&SRS(I)*OCCUR(I&3952)	EFFEC	33
11 CONTINUE	EFFEC	34
C	EFFEC	35
PD = DSORT(PD/HDIF)	EFFEC	36
SIGMA = PD	EFFEC	37
C	EFFEC	38
C	EFFEC	39
C	EFFEC	40
TOLD = -2.00	EFFEC	41
Y = 0	EFFEC	42
K = 0	EFFEC	43
10 K = K & 1	EFFEC	44
TX = A*(TOLD & DABS(TOLD)*CON)	EFFEC	45
Z = .500*(1.00 & DERF(TX))	EFFEC	46
TNEW = TOLD -CON*DABS(TOLD)*(Y-PFD)/(Z-Y)	EFFEC	47
W = .500*(1.00&DERF(A*TNEW))	EFFEC	48
IF(DABS(W-Y).LE.EPS*DABS(W)) GO TO 99	EFFEC	49
IF(K.EQ.20) GO TO 90	EFFEC	50
TOLD = TNEW	EFFEC	51
Y = W	EFFEC	52
GO TO 10	EFFEC	53
90 PD = 1.000	EFFEC	54
WRITE(6,100)	EFFEC	55
RETURN	EFFEC	56
99 CONTINUE	EFFEC	57
C	EFFEC	58
C	EFFEC	59

C	PD = .500 * (1.000 & DERF(A*(TNEW & PO)))	EFFEC 60
C	RETURN	EFFEL 61
100	FORMAT(1MO, 16MERROR IN EFFECT)	EFFEC 62
	END	EFFEC 63
		EFFEC 64
		EFFEC 65

```

C     CLV IL      IMPLICIT REAL*P (A-H,O-Z)
C     SUPROUTINE EVIL(ZZ)
C     REAL*8 MDOOT,NST,MOTWR,K1,K2,LBAR
C     DIMENSION OCCUR(400),NOCCUR(30),TW(4,8),QDOT(4,8),MOCT(4,8)
C     EQUIVALENCE
C     1(OCCUR(04),BETA1 ),(OCCUR(005),BETA2 ),(OCCUR(006),PRTA3 ),
C     2(OCCUR(007),BETA4 ),(OCCUR(014),CP2 ),(OCCUR(022),DFLRHC),
C     3(OCCUR(024),CPSTL ),(OCCUR(025),F ),(OCCUR(029),HSKTJ ),
C     4(OCCUR(030),HREF ),(OCCUR(031),HLS ),
C     5(OCCUR(040),NST ),(OCCUR(047),PS ),(OCCUR(058),PHC2 ),
C     6(OCCUR(068),SPD ),(OCCUR(074),TW ),(OCCUR(091),Z ),
C     7(OCCUR(092),ZTP ),(OCCUR(102),TINIT ),
C     8(OCCUR(100),FACTR2),(OCCUR(101),FACTR3),(OCCUR(102),FACTR4),
C     9(OCCUR(103),FACTR5),(OCCUR(104),FACTR6),(OCCUR(105),FACTR7)
C     EQUIVALENCE
C     1(OCCUR(225),HMBAR ),(OCCUR(2644),TW(1,1)),
C     2(OCCUR(2676),QDOT(1,1)),(OCCUR(2708),MDOOT(1,1))
C     EQUIVALENCE
C     1(MOCCUR(01),JHOLD),(MOCCUR(02),KKHOLD),(MOCCUR(13),MATLNU)
C     COMMON OCCUR,NOCCUR
C
C
C
C
C     SUBROUTINE EVIL CALCULATES THE MASS LOSS RATE - MOCT, THE
C     WALL RECESSION RATE - SPD, AND THE WALL TEMPERATURE - TW FOR
C     THE BODY STATIONS AT WHICH AERODYNAMIC HEATING RATES WERE
C     OBTAINED IN SUBROUTINE AERODY. THE METHOD OF CALCULATION
C     IS DEPENDENT ON THE MATERIAL BEING CONSIDERED. FOR LOW IN
C     TURBULENT FLOW ONLY,LTALPHA,TEFLON,AND THE INPUT MATERIAL IN
C     LAMINAR AND TURBULENT FLOW AN ITERATIVE STEADY STATE SOLUTION
C     IS USED. OTWR IN LAMINAR FLOW, CARBON PHENOLIC, AND PHENOLIC
C     NYLON UTILIZE CURVE FITS OF MDOOT AND SPD AS A FUNCTION OF
C     COLD WALL HEATING FROM AERODY, AND TW AS A FUNCTION OF SUIT.
C
C
C     THE STEADY STATE ABLATION METHOD WHICH ENTAILS ITERATIVE
C     SOLUTIONS OF SIMULTANEOUS EQUATIONS FOR THE SURFACE TEMPERATURE
C     AND THE WALL RECESSION RATE. IT INCLUDES THE FOLLOWING ENERGY
C     CONSIDERATIONS - CONVECTIVE ENERGY, CONDUCTION FLUX, AND
C     SURFACE RADIATION LOSS, AND SUBLIMATION ENERGY.
C
C     J=JHOLD
C     K=KKHOLD
C     ICHUNT=C
C     IF(IJ.NE.2)GO TO 12
C     TW(1,K)=TW(2,K)
C     TW(4,1)=TW(2,1)
C 12 CONTINUE
C     IF(MATLNO.LT.3)GO TO 14
C     IF(MATLNO.EQ.4)GO TO 52
C     IF(MATLNO.GE.6)GO TO 14
C     IF(ZDOT.ATR100 TO 52
C 14 CONTINUE
C     ZZ=TW(IJ,K)
C     PS=0.01506*TW(IJ,K)E25000.0C/TW(IJ,K)

```

```

FV IL 1
FV IL 2
FV IL 3
FV IL 4
FV IL 5
FV IL 6
FV IL 7
FV IL 8
FV IL 9
FV IL 10
FV IL 11
FV IL 12
FV IL 13
FV IL 14
FV IL 15
FV IL 16
FV IL 17
FV IL 18
FV IL 19
FV IL 20
FV IL 21
FV IL 22
FV IL 23
FV IL 24
FV IL 25
FV IL 26
FV IL 27
FV IL 28
FV IL 29
FV IL 30
FV IL 31
FV IL 32
FV IL 33
FV IL 34
FV IL 35
FV IL 36
FV IL 37
FV IL 38
FV IL 39
FV IL 40
FV IL 41
FV IL 42
FV IL 43
FV IL 44
FV IL 45
FV IL 46
FV IL 47
FV IL 48
FV IL 49
FV IL 50
FV IL 51
FV IL 52
FV IL 53
FV IL 54
FV IL 55
FV IL 56
FV IL 57
FV IL 58
FV IL 59

```

1 MOL REBONDING

NOT REPRODUCIBLE

1	ICOUNT=ICOUNT&1	FV IL 60
	TEM=LOG(BETA2**ZZ**BETA3-BETA4/ZZ	EV IL 61
	SPD=BETA1*ZZ&EXP(TEM)	FV IL 62
	TK1=FACTR2-1.111205/ZZ	FV IL 63
	K1=EXP(TK1)	FV IL 64
	PBAR=PS/2117.00	FV IL 65
	IFIK1.LT.(1.00-36)IK1=C.C00	FV IL 66
	K2=SQR(12.765700*K1*K1&4.00*K1*2.993400*(K1&4.00*PBAR)*(FACTR3))	FV IL 67
	K2=K2-12.765700*K1	FV IL 68
	K2=1.500*K2/(K1&4.000*PBAR)	FV IL 69
	GAM1=17.654000&465.585000*K2	FV IL 70
	GAM2=0.25601200&C.555800-2*K2	FV IL 71
	GAM3=0.534500-5-0.427000-6*K2	FV IL 72
	HWRAR=CAM1&ZZ*(GAM2&GAM3*77)	FV IL 73
	HWRAR=HWRAR/37.8&00	FV IL 74
	E1BAR=HWRAR/35.8&00	FV IL 75
	XCOM=0.349000*HSR TO	EV IL 76
	E2BAR=(XCOM&C.500*HWRAR)/(XCOM&17.94500)	EV IL 77
	E3BAR=0.950000-(HWRAR-HREF)/HSR TO	FV IL 78
	IF(JJHOLD.EQ.2)GO TO 2	FV IL 79
	IF(JJHOLD.EQ.3)GO TO 3	FV IL 80
	E1=-0.03700	EV IL 81
	E2=0.00	FV IL 82
	GO TO 4	FV IL 83
2	E1=0.00	FV IL 84
	E2=-0.18500	FV IL 85
	GO TO 4	FV IL 86
3	E1=0.00	FV IL 87
	E2=-0.50200	FV IL 88
4	QDOTCF=QDOT(JJHOLD,KKHOLD)*(E1BAR**E1)*(E2BAR**E2)	EV IL 89
	IF(QDOTCF.LE.0.000)GO TO 77	EV IL 90
	TEM=ZZ-TINIT	FV IL 91
	IF(Z.LT.ZTR)FBAR=SPD*H*FACTR6/QDOTCF	FV IL 92
	IF(Z.LT.ZTR)FBAR=SPD*H*FACTR7/QDOTCF	FV IL 93
	PHIB=EXP(-FBAR*(1.00&C.618000*FBAR))	FV IL 94
	QDOTT=0.47&800-12*PHIB*ZZ**4	EV IL 95
	QDOTC=QDOTCF*FBAR*PHIB	EV IL 96
	IF(MATLNO.EQ.5)GO TO 55	FV IL 97
	QDOTS=SPD*RH12*F	EV IL 98
	LBAR=SPD*(TEM*FACTR4&FACTR5)	EV IL 99
	RBAR=QDOTC-QDOTT-QDOTS	EV IL 100
	ARGU=LBAR-RBAR	FV IL 101
51	CONTINUE	FV IL 102
	IF(ICOUNT.GT.100)GO TO 50	FV IL 103
5	IF(DABS(ARGU).LT.1.00)GO TO 50	FV IL 104
6	IF(DABS(ARGU).LT.DABS(C.C1000*LBAR))GO TO 50	FV IL 105
7	IF(ARGU)8, 50, 9	EV IL 106
8	IF(US)11, 50, 10	EV IL 107
9	IF(OS)10, 50, 11	FV IL 108
11	OS=-0.500*OS	FV IL 109
10	ZZ=ZZ&DS	FV IL 110
	GO TO 1	EV IL 111
50	ZZ=OMAX(ZZ,0.0001)	FV IL 112
77	IF(QDOTCF.LE.0.000)SPD=C.000	FV IL 113
	SPD=OMAX(SPD,C.C00)	EV IL 114
	HOIT(J,K)=SPD*(RHO2&DEL RHO)	FV IL 115
	RETURN	FV IL 116
		FV IL 117
		EV IL 118
		FV IL 119
52	IF(MATLNO.EQ.5)GO TO 54	


```

C
C
      OTWR CALCULATIONS FOR LAMINAR FLOW
      QDOTCW=QDOT(JJHOLD,KKHOLD)
      IF(QDOTCW.LT.1.701160)GO TO 58
      IF(QDOTCW.LT.1.5001160)GO TO 58
      IF(QDOTCW.LT.2.5002160)GO TO 59
      IF(QDOTCW.LT.3.0003160)GO TO 62
      MDOT(J,K)=QDOTCW*(1.000-1.503/HS)/(3.640368.100*HS*TO)
      GO TO 57
62 MDOT(J,K)=-1.056500250-367.611186990-5*QDOTCW&3.342517000-8*QDOTCW
1**7-6.916824220-12*QDOTCW**3
      GO TO 57
56 MDOT(J,K)=0.000
      GO TO 57
55 MDOT(J,K)=0.000210660.(001500*(QDOTCW-13.000))
      GO TO 57
59 MDOT(J,K)=-1.274243350-361.360716700-4*QDOTCW-1.090915160-6*QDOTCW
1**28.982757470-9*QDOTCW**3-1.652105790-11*QDOTCW**4
57 MOTWR=MDOT(J,K)
      IF(QDOTCW.LT.1.0003160)GO TO 75
      IF(QDOTCW.LT.3.0003160)GO TO 72
      SPD=MOTWR/(RHO2&DELTA)
      GO TO 72
75 SPD=1.61150-6*QDOTCW-5.237410-4
      GO TO 73
72 SPD=-1.111967600-464.033767100-7*QDOTCW&9.701312610-10*QDOTCW**2-
12.455275040-13*QDOTCW**3
73 IF(SPD.LT.1.00-13)SPD=1.00-13
      SOTWR=SPD
      IF(MATLNO.EQ.4)GO TO 53
      SPDL=LOG10(SPD)
      IF(SPDL.LT.-1.2583011)SPDL=-12.58300
      ZZ=6.3463451203&5.5062879602*SPDL&1.96585366011*SPDL**2
      IF(ZZ.GT.4.8503122)=4850.000
      RETURN
C
C
      PHENOLIC NYLON CALCULATIONS
53 CONTINUE
      QDOTCW=QDOT(JJHOLD,KKHOLD)
      QDOTCW=DMAX1(QDOTCW,C.CDC)
      IF(QDOTCW.LT.1.0002160)GO TO 60
      IF(QDOTCW.GT.3.0003160)GO TO 74
      MDOT(J,K)=-1.623676420-361.789227930-4*QDOTCW&1.321136960-8*
1QDOTCW**2-5.067475130-12*QDOTCW**3
      GO TO 61
60 MDOT(J,K)=10.000**(-2.522880667.37590-3*QDOTCW)
      GO TO 61
74 MDOT(J,K)=QDOTCW*(1.000-1.703/HS)/(1.84503&1.1101*HS*TO)
61 CONTINUE
      IF(MDOT(J,K).LT.1.00-4)MDOT(J,K)=0.000
      IF(MOTWR.LT.1.00-5)MOTWR=1.00-5
C
      PHENOLIC NYLON SDOT IS RATED FROM CTWR SDOT
      SPD=(SOTWR*MDOT(J,K))/MOTWR
      IF(SPD.LT.1.00-5)SPD=1.00-5
      SPDL=LOG10(SPD)
      IF(SPDL.LT.1.00-4)GO TO 76
      ZZ=6.9447403503&6.4536714602*SPDL-1.4858917302*SPDL**2
      IF(ZZ.GT.4.8503122)=4850.000
      RETURN
76 ZZ=250.000&1756.000*(SPDL&5.000)

```

```

EV IL 120
EV IL 121
EV IL 122
EV IL 123
EV IL 124
EV IL 125
EV IL 126
EV IL 127
EV IL 128
EV IL 129
EV IL 130
EV IL 131
EV IL 132
EV IL 133
EV IL 134
EV IL 135
EV IL 136
EV IL 137
EV IL 138
EV IL 139
EV IL 140
EV IL 141
EV IL 142
EV IL 143
EV IL 144
EV IL 145
EV IL 146
EV IL 147
EV IL 148
EV IL 149
EV IL 150
EV IL 151
EV IL 152
EV IL 153
EV IL 154
EV IL 155
EV IL 156
EV IL 157
EV IL 158
EV IL 159
EV IL 160
EV IL 161
EV IL 162
EV IL 163
EV IL 164
EV IL 165
EV IL 166
EV IL 167
EV IL 168
EV IL 169
EV IL 170
EV IL 171
EV IL 172
EV IL 173
EV IL 174
EV IL 175
EV IL 176
EV IL 177
EV IL 178
EV IL 179

```

	RETURN	EV IL 180
C		FV IL 181
C	CARBON PHENOLIC CALCULATIONS FOR LAMINAR FLOW	FV IL 182
54	CONTINUE	FV IL 183
	QDOTCW=QDOT(JJHOLD,KKHOLD)	FV IL 184
	IF(QDOTCW.LT.8.0D0)GO TO 64	FV IL 185
	IF(QDOTCW.LT.1.1D0)GO TO 65	FV IL 186
	IF(QDOTCW.LT.2.31D0)GO TO 66	FV IL 187
	IF(QDOTCW.LE.1.0D0)GO TO 67	FV IL 188
	MDOT(J,K)=1.2324402D-263.30858843D-5*QDOTCW64.60267808D-9*QDOTCW	FV IL 189
	1*2	FV IL 190
	GO TO 68	FV IL 191
64	MDOT(J,K)=C.0D0	FV IL 192
	GO TO 68	FV IL 193
65	MDOT(J,K)=6.5788D-4-1.55773D-4*QDOTCW69.9485D-6*QDOTCW**2	FV IL 194
	GO TO 68	FV IL 195
66	MDOT(J,K)=1.1D-465.76022D-5*(QDOTCW-1.13D1)	FV IL 196
	GO TO 68	FV IL 197
67	MDOT(J,K)=-1.73043255D-361.25965766D-4*QDOTCW-8.56939D32D-8*QDOTCW	FV IL 198
	1**264.65548312D-12*QDOTCW**3	FV IL 199
68	CONTINUE	FV IL 200
	IF(QDOTCW.LT.2.47D1)SPD=1.0D-18	FV IL 201
	IF(QDOTCW.LT.2.47D1)GO TO 69	FV IL 202
	IF(QDOTCW.LT.2.75D2)GO TO 70	FV IL 203
	IF(QDOTCW.LT.1.0D3)GO TO 71	FV IL 204
	SPD=-3.196484D-463.865D-7*QDOTCW62.171792D-10*QDOTCW**2	FV IL 205
	GO TO 69	FV IL 206
70	SPD=-2.122955D-261.18955951D-7*QDOTCW61.40098706D-10*QDOTCW**2	FV IL 207
	GO TO 69	FV IL 208
71	SPD=7.66751321D-5-4.5503115D-7*QDOTCW61.55981653D-9*QDOTCW**2	FV IL 209
	1-4.4421534D-13*QDOTCW**3	FV IL 210
69	CONTINUE	FV IL 211
61	SPDL=LOG10(SPD)	FV IL 212
	ZZ=0.68741134D364.46431845D2*SPDL61.20991423D1*5PDL**2	FV IL 213
	IF(ZZ.GT.0.16D3)ZZ=616C.LD0	FV IL 214
	RETURN	FV IL 215
C		FV IL 216
C	CARBON PHENOLIC CALCULATIONS FOR TURBULENT FLOW	FV IL 217
55	CONTINUE	FV IL 218
	QNET=QDOTCW-QDOTTT	FV IL 219
	ABAR=C22*(616C.CDC-TNG)6F	FV IL 220
	HBAR=NST	FV IL 221
	QSTAR=APAR*CBAR**HRTO	FV IL 222
	MDOT(J,K)=QNET/QSTAR	FV IL 223
	IF(QNET.LT.0.0D0)MDOT(J,K)=C.0D0	FV IL 224
	SPD=(J.2D0)*MDOT(J,K)/7RH02	FV IL 225
	IF(SPD.LE.1.0D-10)SPD=1.0D-10	FV IL 226
	SPDL=LOG10(SPD)	FV IL 227
	IF(SPDL.LT.-1.8058D1)SPDL=-18.0580D	FV IL 228
	ZZ=0.68741134D364.46431845D2*SPDL61.20991423D1*5PDL**2	FV IL 229
	RETURN	FV IL 230
	END	FV IL 231

NOT REPRODUCIBLE

C	FCN	SUBROUTINE FCN(N,G,F,X,M)	FCN	1
		IMPLICIT REAL*8 (A-H,O-Z)	FCN	2
		COMMON /CICUR/ IOCCUR(30)	FCN	3
		COMMON /DLK I/ IS(5),MS,IS(16)	FCN	4
		COMMON /END/ ITERM	FCN	5
		COMMON /DPT/ DELX(20)	FCN	6
		COMMON OCCUR, NOCCUR	FCN	7
		DIMENSION X(40),G(40),F(40)	FCN	8
		DIMENSION OCCUR(400), NOCCUR(30)	FCN	9
		DIMENSION TOP(20),NPLOT(5),NPSAVE(14)	FCN	10
		EQUIVALENCE (NOCCUR(14),NPRINT), (NOCCUR(24), NPLOT(1))	FCN	11
		EQUIVALENCE (IOCCUR(200), TOP)	FCN	12
C			FCN	13
C			FCN	14
C			FCN	15
C		SUBROUTINE FCN MUST COMPUTE THE FUNCTION F AND THE GRADIENT	FCN	16
C		VECTOR G (1,F,, THE VECTOP WHOSE ELEMENTS ARE THE FIRST	FCN	17
C		PARTIAL DERIVATIVES OF F WITH RESPECT TO THE PARAMETERS),	FCN	18
C		GIVEN THE N COORDINATES OF THE POINT X.	FCN	19
C			FCN	20
C			FCN	21
		DO 416 I = 1, 6	FCN	22
		NPSAVE(I) = TOP(IG63)	FCN	23
		IF(I.LT.6)NPSAVE(IG6) = NPLOT(I)	FCN	24
416		IF(I.LT.4)NPSAVE(IG11) = TOP(I)	FCN	25
C			FCN	26
		IF(M1.NE.3) GO TO 5	FCN	27
C			FCN	28
		NPRINT = 1	FCN	29
		CALL FFV(N,X,F)	FCN	30
		NPRINT = 0	FCN	31
		GO TO 106	FCN	32
C			FCN	33
5		CALL FEVIN,X,F)	FCN	34
		IF(ITERM.NE.0) GO TO 106	FCN	35
C			FCN	36
		DO 417 I = 1, 6	FCN	37
		TOP(IG63) = 0	FCN	38
		IF(I.LT.6) NPLOT(I) = 0	FCN	39
417		IF(I.LT.4) TOP(I) = 0	FCN	40
10		DO 105 I=1,N	FCN	41
		X(I)=X(I)+DELX(I)	FCN	42
		CALL FEVIN, X, FM)	FCN	43
		IF(ITERM.NE.0) GO TO 106	FCN	44
		G(I) = (FM - F)/DELX(I)	FCN	45
		X(I)=X(I)-DELX(I)	FCN	46
105		CONTINUE	FCN	47
C			FCN	48
106		DO 418 I = 1, 6	FCN	49
		TOP(IG63) = NPSAVE(I)	FCN	50
		IF(I.LT.6) NPLOT(I) = NPSAVE(IG6)	FCN	51
418		IF(I.LT.4) TOP(I) = NPSAVE(IG11)	FCN	52
C			FCN	53
		IF(M1.NE.1) GO TO 1000	FCN	54
		IPC=1	FCN	55
		IC=1	FCN	56
1000		WRITE(6,915) IC, MS, F, (X(I),I=1,N)	FCN	57
C			FCN	58
915		FORMAT(1H /10X, 'PRINT FCN',5X, 'PASS',15, 5X, 'RANDOM STEP',15,5X	FCN	59

	1*F=*,1P520.8/9X *X* 1P5F20.8/3 (10X,1P5F20.8/1)	FCN	60
	WR IT(6,916) (G(1),I=1,N)	FCN	61
916	FORMAT(1H0 9X *G* 1P5F20.8/(1H 10X 1P5F20.8))	FCN	62
	IC=IC&1	FCN	63
	RETURN	FCN	64
	END	FCN	65

CFEV		FFV	1
	SUBROUTINE FEV(N,X,VAL)	FFV	2
	IMPLICIT REAL*8 (A-H,O-Z)	FFV	3
	COMMON /IDNOS/ ID1(50), ID2(50)	FFV	4
	COMMON OCCUR, NOCCUR	FFV	5
	COMMON /IDPT/ IPROC, IN, NCONS, IPNT, IEX, LIMIT, IRAND	FFV	6
	COMMON /MIN/ALOW(20), UP(20), AMULT(20), CALOW(20), CUP(20)	FFV	7
	COMMON /CCRN/D	FFV	8
	COMMON /IXCOM/ XCOM(200), ICOM(200)	FFV	9
	COMMON /END/ ITERM	FFV	10
	DIMENSION X(20), PNLT(20)	FFV	11
	DIMENSION OCCUR(4000), NOCCUR(30)	FFV	12
	EQUIVALENCE(OCCUR(4000),IP)	FFV	13
	EQUIVALENCE(NOCCUR(14),NPINT)	FFV	14
C		FFV	15
C		FFV	16
C	SUBROUTINE FEV EVALUATES THE PENALTY FUNCTION WHICH IS BEING	FFV	17
C	OPTIMIZED FOR THE POINT DESCRIBED BY THE N VALUES OF THE X	FFV	18
C	ARRAY.	FFV	19
C		FFV	20
C		FFV	21
	ITERM=0	FFV	22
	VAL=0.000	FFV	23
C		FFV	24
	DO 10 I=1,N	FFV	25
	IZ=ID1(I)	FFV	26
	OCCUR(IZ)=X(I)	FFV	27
10	CONTINUE	FFV	28
C		FFV	29
	IF(ICOM(1).NE.1) GO TO 15	FFV	30
	CALL CLASSC(VAL)	FFV	31
	GO TO 22	FFV	32
C		FFV	33
15	CALL SCREEN(N,X,ISUC,VAL,D)	FFV	34
	IF(ISUC.LT.0) GO TO 100	FFV	35
C		FFV	36
C		FFV	37
	CALL F123(VAL)	FFV	38
C		FFV	39
C		FFV	40
	IF(ILP.LT.6) GO TO 20	FFV	41
	ITERM = 1	FFV	42
	GO TO 100	FFV	43
C		FFV	44
20	CALL MISC (N,X,VAL)	FFV	45
	CALL EFFECT	FFV	46
C		FFV	47
C		FFV	48
C		FFV	49
22	IF(NCONS.FO.0)GO TO 100	FFV	50
	WRITF(6,1000)	FFV	51
1000	FORMAT(11H* IZ*4X*LOWER BOUND*4X*UPPER BOUND*6X* OCCUR(IZ)*8X	FFV	52
	I*PENALTY*)	FFV	53
C		FFV	54
C		FFV	55
	DO 50 I=1,NCONS	FFV	56
	PNLT(I) = 0.000	FFV	57
	IZ = ID2(I)	FFV	58
	IF(OCCUR(IZ).GT.CALOW(IZ)) GO TO 25	FFV	59

	PNLTY(I) = AMULT(I)*DABS(OCCUR(IZ)-CALW(I))*IEX	FEV	60
	WRITE(6,1100) IZ,CALW(I),CUP(I),OCCUR(IZ),PNLTY(I)	FEV	61
	GO TO 50	FEV	62
25	IF(OCCUR(IZ),LT,CUP(I)) GO TO 27	FEV	63
	PNLTY(I) = AMULT(I)*DABS(OCCUR(IZ)-CUP(I))*IEX	FEV	64
27	WRITE(6,1100) IZ,CALW(I),CUP(I),OCCUR(IZ),PNLTY(I)	FEV	65
1100	FORMAT(3H 14, 1P4F15.7)	FEV	66
	50 VAL = VAL + PNLTY(I)	FEV	67
C		FEV	68
C		FEV	69
	E = VAL	FEV	70
	IF(VAL.GT.C.000)GO TO 100	FEV	71
	ITERM = -1	FEV	72
C		FEV	73
100	CONTINUE	FEV	74
	WRITE(6,900) VAL, (X(I), I=1,N)	FEV	75
C		FEV	76
900	FORMAT(1H **FEV**2X*F =*1P15.7,* X =*5E15.7/(11H 29X1P5E15.7))	FEV	77
999	RETURN	FEV	78
	END	FEV	79

CFIB1		FIB1	1
	IMPLICIT REAL*8 (A-H,O-Z)	FIB1	2
	SUBROUTINE FIB1 (AA, BB, NF, NMIMAX, ACCUR, NFUNC, XMIMAX, YMIMAX)	FIB1	3
C		FIB1	4
C		FIB1	5
C	SUBROUTINE FIB1 UTILIZES A FIBONACCI SEARCH TECHNIQUE TO	FIB1	6
C	FIND THE MAXIMUM OR MINIMUM OF A ONE VARIABLE UNIMODAL	FIB1	7
C	FUNCTION WITHIN A DEFINED REGION (AA, BB).	FIB1	8
C		FIB1	9
C		FIB1	10
	IW=6	FIB1	11
	DIMENSION E(40)	FIB1	12
	ITEM = 0	FIB1	13
	A=AA	FIB1	14
	B=BB	FIB1	15
	IF (NMIMAX) 10, 11, 11	FIB1	16
10	JMIMAX=1	FIB1	17
	GO TO 20	FIB1	18
11	JMIMAX=2	FIB1	19
20	IF (B-A) 21, 22, 23	FIB1	20
21	TEMP=A	FIB1	21
	A=B	FIB1	22
	B=TEMP	FIB1	23
	GO TO 23	FIB1	24
22	L=1	FIB1	25
25	WRITE (IW, 26) L	FIB1	26
26	FORMAT(17HC ERROR OF TYPE I2)	FIB1	27
	ITEM = 1	FIB1	28
	GO TO 170	FIB1	29
23	B=B-A	FIB1	30
	IF (NFUNC) 31, 30, 35	FIB1	31
30	IF (ACCUR) 31, 31, 40	FIB1	32
31	L=L+1	FIB1	33
	GO TO 25	FIB1	34
35	NFUNC2=NFUNC+2	FIB1	35
	R1=(1.000+SQRT(5.000))/2.000	FIB1	36
	R2=(1.000-SQRT(5.000))/2.000	FIB1	37
	ROACC=(R1*NFUNC2-R2*NFUNC)/ (R1-R2)	FIB1	38
	GO TO 45	FIB1	39
40	ROACC=B/ACCUR	FIB1	40
45	IF (ROACC-2.000) 50, 50, 51	FIB1	41
50	XMIMAX=(A+B)/2.000	FIB1	42
	YMIMAX = FUN1(XMIMAX, NF)	FIB1	43
	GO TO 170	FIB1	44
51	IF (ROACC-3.000) 60, 60, 61	FIB1	45
60	XL=A+(B-A)/3.000	FIB1	46
	XR=A+2.000*(B-A)/3.000	FIB1	47
	YL = FUN1(XL, NF)	FIB1	48
	IF (ITEM, NF, 0) GO TO 160	FIB1	49
	YR = FUN1(XR, NF)	FIB1	50
	IF (ITEM, NF, 0) GO TO 150	FIB1	51
	IF (YL-YR) 70, 70, 71	FIB1	52
70	XMIMAX=XR	FIB1	53
	YMIMAX=YR	FIB1	54
	GO TO 170	FIB1	55
71	XMIMAX=XL	FIB1	56
	YMIMAX=YL	FIB1	57
	GO TO 170	FIB1	58
61	K=2	FIB1	59

E(1) = 1.000	F101 60
F(2)=1.000	F101 61
74 E(K)=E(K-1)+F(K-2)	F101 62
IF (RDACC-E(K)) 80,80,75	F101 63
75 IF (K-40) 77,76,76	F101 64
76 L=?	F101 65
GO TO 25	F101 66
77 K=K&1	F101 67
GO TO 74	F101 68
80 N=K	F101 69
XL=A&E(N-2)*R/F(N)	F101 70
XR=A&F(N-1)*R/F(N)	F101 71
YL = FUN1(XL,NF)	F101 72
IF(ITERM.NF.0) GO TO 160	F101 73
YR = FUN1(XR,NF)	F101 74
IF(ITERM.NF.0) GO TO 150	F101 75
J=1	F101 76
90 NJ=N-J	F101 77
NJONE=N-J-1	F101 78
NJTWO=N-J-2	F101 79
GO TO (100,101), UMIMAX	F101 80
100 IF (YR-YL) 130,130,110	F101 81
101 IF (YR-YL) 110,110,130	F101 82
110 IF (J-N&3) 120,140,160	F101 83
120 N=XR	F101 84
A=A	F101 85
R=B-A	F101 86
XP=XL	F101 87
XL=A&E(NJTWO)*R/E(NJ)	F101 88
YR=YL	F101 89
YL = FUN1(XL,NF)	F101 90
IF(ITERM.NF.0) GO TO 160	F101 91
125 J=J&1	F101 92
GO TO 90	F101 93
130 IF (J-N&3) 140,150,150	F101 94
140 R=R	F101 95
A=XL	F101 96
R=B-A	F101 97
XL=XR	F101 98
XR=A&E(NJONE)*R/E(NJ)	F101 99
YL=YR	F101 100
YR = FUN1(XR,NF)	F101 101
IF(ITERM.NF.0) GO TO 150	F101 102
GO TO 125	F101 103
150 XMIMAX=XR	F101 104
YMIMAX=YR	F101 105
GO TO 170	F101 106
160 XMIMAX=XL	F101 107
YMIMAX=YL	F101 108
170 RETURN	F101 109
END	F101 110


```

CFIRF
SUBROUTINE FIRE
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON/BLK0/ H(40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
  L,GS, GSP, GTP,GSS, GTT, GSR,F,FP,FO,FOE, P,TO,RS,SL, Z,Q,A,EL,
  ZDELTA,FAC,C(40,10)
  COMMON/BLK1/M,N,L,LS,M1,MS,NS,I,T,K,NC
  COMMON /END/ ITERM
  FIRE 1
  FIRE 2
  FIRE 3
  FIRE 4
  FIRE 5
  FIRE 6
  FIRE 7
  FIRE 8
  FIRE 9
  FIRE 10
  SUBROUTINE FIRE, PART OF THE DAVENON MINIMIZATION CALCULA
  TIONS, EVALUATES THE PENALTY FUNCTION AND ITS GRADIENT AT THE
  INTERPOLATED POINT AND DETERMINES IF THE LOCAL MINIMUM HAS BEEN
  SUFFICIENTLY WELL LOCATED. IF SO, THE RATE OF CHANGE OF THE
  GRADIENT IS EVALUATED BY INTERPOLATING FROM ITS VALUES AT THE
  END POINTS OF THE INTERVAL BEING CONSIDERED AND ITS VALUE AT
  THE INTERPOLATED POINT.
  FIRE 11
  FIRE 12
  FIRE 13
  FIRE 14
  FIRE 15
  FIRE 16
  FIRE 17
  FIRE 18
  FIRE 19
  FIRE 20
  400 M1=2
  401 CALL FCN(N,GB,FP,T,M1)
  IF(ITERM.EQ.0) GO TO 402
  FIRE 21
  FIRE 22
  C
  FO 10 I = 1,N
  X(I) = T(I)
  GO TO 412
  FIRE 23
  FIRE 24
  FIRE 25
  FIRE 26
  C
  402 M=1
  403 CALL MATMP (M,N,S,GB,GSR)
  TP1= OMN1(F,FP)
  FIRE 27
  FIRE 28
  405 IF (TP1-FOE)141P,406,406
  FIRE 29
  FIRE 30
  406 TP1=A/(1.0-A)
  FIRE 31
  FIRE 32
  407 TP2=(1.0-A)/A
  FIRE 33
  FIRE 34
  408 TO=GSR*(TP1-TP2)
  FIRE 35
  FIRE 36
  409 IF(DABS(TO)-0) 413,410,410
  FIRE 37
  FIRE 38
  410 GSS=2.0*0
  FIRE 39
  FIRE 40
  411 L=1
  FIRE 41
  FIRE 42
  C
  412 RETURN
  FIRE 43
  FIRE 44
  C
  413 GSS=1062.0*0
  FIRE 45
  FIRE 46
  414 DO 415 I=1,N
  FIRE 47
  FIRE 48
  415 G(I)=(GB(I)-G(I))*TP1&(GP(I)-GB(I))*TP2
  FIRE 49
  FIRE 50
  C
  416 L=2
  FIRE 51
  FIRE 52
  417 GO TO 412
  FIRE 53
  FIRE 54
  C
  418 IF (F-FO)419,428,428
  FIRE 55
  FIRE 56
  419 WRITE(6,1)
  FIRE 57
  FIRE 58
  420 SL=(1.0-A)*EL
  FIRE 59
  FIRE 60
  421 FP=FB
  FIRE 61
  FIRE 62
  422 CSP=GSR
  FIRE 63
  FIRE 64
  423 DO 425 I=1,N
  FIRE 65
  FIRE 66
  424 XP(I)=T(I)
  FIRE 67
  FIRE 68
  425 C(I)=GB(I)
  FIRE 69
  FIRE 70
  C
  426 L=3
  FIRE 71
  FIRE 72
  427 GO TO 412
  FIRE 73
  FIRE 74
  C
  FIRE 75

```

428 WRITE(6,2)	FIRE	60
429 CL=F1*A	FIRE	61
430 F=FN	FIRE	62
431 OS=GSR	FIRE	63
432 DO 424 I=1,N	FIRE	64
433 XI(I)=T(I)	FIRE	65
434 GI(I)=GR(I)	FIRE	66
435 GO TO 426	FIRE	67
1 FORMAT (10HMOVE LEFT)	FIRE	68
2 FORMAT (11HMOVE RIGHT)	FIRE	69
END	FIRE	70

C FLOW

```

SUBROUTINE FLOWE (FNFT,R1,IND)
IMPLICIT REAL *8 (A-H,O-Z)
COMMON ZNINPUT/ AKW,RH0W,CRH0W,DELWH, TABL,OHCHFM,RHUSL,
1 PTC,BCON,ACON, P21,R22,R23, C(159)
COMMON ZTBL512/ ERNTBL (R,10),EPARTB (10),ERNUTB (R),
1 ENTABL (25,9),RSTABL (9),HSTABL (25),
2 ETABL (12,11),THTTBL (11),EMCTBL (12),
3 D(11,5,4),XYZTBL(11,3)
COMMON ZERLTHX/ RZ,KN,THTAC,OB,GAMMAF,COTA,COVA,PCICPU,PU,
1 AMU,RH0U,UU,HU,AMUO,AMINFC,RH0IC,MINFC,UINFC,AMLSO,
2 AMABLD,7,SC,ZBLT
EQUIVALENCE (OCUR(1567),CCON)
DIMENSION OCUR(400)
DIMENSION NOCUR(20)
COMMON OCCUR,NOCCUR
EQUIVALENCE
1 (RHO,RHOU),(V,UU)

```

FLOWE 1
FLOWE 2
FLOWE 3
FLOWE 4
FLOWE 5
FLOWE 6
FLOWE 7
FLOWE 8
FLOWE 9
FLOWE 10
FLOWE 11
FLOWE 12
FLOWE 13
FLOWE 14
FLOWE 15
FLOWE 16
FLOWE 17
FLOWE 18
FLOWE 19
FLOWE 20

SUBROUTINE FLOWE CALCULATES THE ELECTRON DENSITY TRANSITION,
N SUP FT (E/CC), AND THE DECAY RATE, OI.

FLOWE 21
FLOWE 22
FLOWE 23
FLOWE 24
FLOWE 25
FLOWE 26

INPUT TABLES

```

ERNTBL = EQUILIBRIUM NORMAL SHOCK ELECTRON DENSITY TABLE
(TABLE F)
ERNRTA = RHO, DENSITY ARGUMENT OF TABLE F (LBM/FT-3)
ERNUTB = VELOCITY ARGUMENT OF TABLE F (1000 FT/SEC)

ENTABL = EQUILIBRIUM ELECTRON DENSITY TABLE (TABLE B)
HSTABL = H/PTC, ENTHALPY ARGUMENT OF TABLE B, NON-DIMENSIONAL
RSTABL = RHO/RHOC-DENSITY ARGUMENT OF TABLE B, NON-DIMENSIONAL

ETABL = TABLE OF MACH NUMBER (TABLE E)
EMCTBL = MC, CONE MACH NUMBER-ARGUMENT OF TABLE E
THTTBL = THETAC, CONE HALF ANGLE-ARGUMENT OF TABLE E, DEGREES

D = ELECTRON DENSITY AS A FUNCTION OF NORMALIZED ENTHALPY
AND AIR DENSITY FOR 1000 PPM SODIUM SEED (TABLE C)
XCTABL = RHO/RHOC, AIR DENSITY ARGUMENT OF TABLE D, NON-
DIMENSIONAL
YCTABL = H/PTC, ENTHALPY ARGUMENT OF TABLE D, NON-DIMENSIONAL
ZOTABL = MPAT, THE RATIO OF ABLATED MASS LOSS RATE TO THE MASS
LOSS RATE IN THE BOUNDARY LAYER, ARGUMENT OF TABLE D

```

FLOWE 27
FLOWE 28
FLOWE 29
FLOWE 30
FLOWE 31
FLOWE 32
FLOWE 33
FLOWE 34
FLOWE 35
FLOWE 36
FLOWE 37
FLOWE 38
FLOWE 39
FLOWE 40
FLOWE 41
FLOWE 42
FLOWE 43
FLOWE 44
FLOWE 45
FLOWE 46
FLOWE 47
FLOWE 48
FLOWE 49
FLOWE 50

THE FOLLOWING ARE INPUT QUANTITIES.

```

AKW = HEATSHIELD CONDUCTIVITY, (BTU/FT-HR-DEG. R)
AMABLD = MASS ABLATED (LBM/ SEC)
AMLSO = MASS SWALLOWED BY BOUNDARY LAYER (LBM/SEC)
AMINFC = CONE MACH NUMBER
AMU = UPSTREAM MACH NUMBER

```

FLOWE 51
FLOWE 52
FLOWE 53
FLOWE 54
FLOWE 55
FLOWE 56
FLOWE 57
FLOWE 58
FLOWE 59

C	AMU = FREE STREAM VISCOSITY (LBM/FT-SEC)	FLOWF 60
C	B7 = SCALE HEIGHT (1,000 FT)	FLOWF 61
C	B21 = SCALING CONSTANT	FLOWF 62
C	B22 = SCALING CONSTANT	FLOWF 63
C	B23 = SCALING CONSTANT	FLOWF 64
C	COTA = TOTAL DRAG*AREA (FT-SQ)	FLOWF 65
C	CDVA = VISCOUS DRAG*AREA (FT-SQ)	FLOWF 66
C	CRHOW = HEATSHIELD SPECIFIC HEAT, (BTU/LB-DEG. R)	FLOWF 67
C	C(50) = PRESET CONSTANT	FLOWF 68
C	C(60) = PRESET CONSTANT	FLOWF 69
C	C(67) = PRESET CONSTANT	FLOWF 70
C	C(69) = PRESET CONSTANT	FLOWF 71
C	C(83-93) = PRESET CONSTANTS	FLOWF 72
C	C(100) = PRESET CONSTANT	FLOWF 73
C	C(115-125) = PRESET CONSTANTS	FLOWF 74
C	C(130-136) = PRESET CONSTANTS	FLOWF 75
C	C(159) = PRESET CONSTANT	FLOWF 76
C	C(160) = PRESET CONSTANT	FLOWF 77
C	C(164) = PRESET CONSTANT	FLOWF 78
C	C(165) = PRESET CONSTANT	FLOWF 79
C	C(169) = PRESET CONSTANT	FLOWF 80
C	DB = BASE DIAMETER, (FT)	FLOWF 81
C	DELWH = HEATSHIELD THICKNESS, (IN)	FLOWF 82
C	DHCHFM = CHEMICAL ENTHALPY OF HEATSHIELD (FT-SQ / SEC-SQ)	FLOWF 83
C	GAMMAE = ENTRY ANGLE (RAD)	FLOWF 84
C	HINFC = CONE ENTHALPY (FT-SQ / SEC-SQ)	FLOWF 85
C	HU = UPSTREAM STATIC ENTHALPY (FT-SQ/SEC-SQ)	FLOWF 86
C	PICIPU = RATIO OF CONE TO UPSTREAM PRESSURE	FLOWF 87
C	PU = UPSTREAM PRESSURE (LB/FT-SQ)	FLOWF 88
C	RHOIC = CONE DENSITY (LRM/FT-3)	FLOWF 89
C	RHOSL = SEA LEVEL DENSITY (LRM/FT-3)	FLOWF 90
C	RHOU = UPSTREAM DENSITY, (LRM/FT-3)	FLOWF 91
C	RHOW = HEATSHIELD DENSITY, (LRM/FT-3)	FLOWF 92
C	RN = NOSE RADIUS (FT)	FLOWF 93
C	RTO = REFERENCE ENTHALPY (FT-SQ/SEC-SQ)	FLOWF 94
C	SC = LENGTH OF CONICAL FRUSTUM (FT)	FLOWF 95
C	TABL = ABLATION TEMPERATURE OF HEATSHIELD (DEG. K)	FLOWF 96
C	THETAC = CONE ANGLE (RAD)	FLOWF 97
C	UINFC = CONE VELOCITY (1000 FT/SEC)	FLOWF 98
C	UU = UPSTREAM VELOCITY, (1000 FT/SEC)	FLOWF 99
C	Z = ALTITUDE (1,000 FT)	FLOWF 100
C	ZBLT = BOUNDARY LAYER TRANSITION ALTITUDE (1,000 FT)	FLOWF 101
C		FLOWF 102
C		FLOWF 103
C	PI = 3.141592654	FLOWF 104
C		FLOWF 105
C		FLOWF 106
C	AKV = HEATING CONSTANT	FLOWF 107
C	REUD = REYNOLDS NUMBER UPSTREAM	FLOWF 108
C		FLOWF 109
C	REUD = 1.03*RHOU*UU*DB/AMU	FLOWF 110
C	AKV = AKV*UU*(DSIN(GAMMAE))/(RHOW*CRHOW*DELWH**2.00)	FLOWF 111
C	ZKECP = Z	FLOWF 112
C	IF (Z.CE.ZBLT) GO TO 1	FLOWF 113
C	Z = ZBLT	FLOWF 114
C	GO TO 1	FLOWF 115
C		FLOWF 116
C		FLOWF 117
C	TWC = TEMPERATURE WALL CONE (DEG. K)	FLOWF 118
C		FLOWF 119

```

3 Z = ZKEFF                                FLOWF120
TWC = C(16C)*(AKV*(ZBLT-7))**C(159)/((SC*DEXP(Z/BZ)/DSIN(THETAC))*FLOWF121
1*2)**.5D0*(AKW/(DFLWH*UU**3.D0)))&TWC FLOWF122
GO TO 2 FLOWF123
1 TWC = 278.D0 & C(6C)*(AKV*Z)**C(159) / ((SC*DEXP(Z/BZ)/DSIN(THETAC))*FLOWF124
1*2)**.5D0*(AKW/(DFLWH*UU**3.D0))) FLOWF125
IF (ZKEFF.LT.ZBLT) GO TO 3 FLOWF126
C FLOWF127
C FLOWF128
C AMBLDS = MASS SWALLOWED BY BOUNDARY LAYER (LRM/ SEC) FLOWF129
C TW = TEMPERATURE WALL (DEG. K) FLOWF130
C TWSP = TEMPERATURE WALL SPHERE (DEG. K) FLOWF131
C FLOWF132
2 TWSP = 278.D0 & C(6C)*(AKV*Z)**C(159) / (AKW*DB**2.D0*DEXP(Z/14.D0 FLOWF133
1 *BZ)) / (UU**1.5D0*DFLWH)) FLOWF134
AMBLDS = AMBLSD & AMARLD FLOWF135
IF ((TABL.LE.TWC.AND.2.D0*RN.LT.9.9999*DB).OR.(TABL.LE.TWSP FLOWF136
1 .AND.2.D0*PN.GT.9.9999*DB)) TW=TABL FLOWF137
IF (2.D0*RN.LT.9.9999*DB.AND.TWC.LT.TABL) TW=TWC FLOWF138
IF (2.D0*RN.GT.9.9999*DB.AND.TWSP.LT.TABL) TW=TWSP FLOWF139
C FLOWF140
C FLOWF141
C CDSS2A = DRAG*APFA 2ND ENTPOPY LAYER (FT-SQ) FLOWF142
C HW = ENTHALPY AT THE WALL (FT-SQ/ SEC-SQ) FLOWF143
C FLOWF144
C HW = 1.08704*TW FLOWF145
C FLOWF146
C CDSS2A = COTA-CDVA FLOWF147
IF (IND.NF.1) GO TO 203 FLOWF148
WRITE (6,103) REUD,AKV,TWC,TWSP,AMBLDS,TW,HW,CDSS2A FLOWF149
103 FORMAT (//9X,4HPREUD,12X,3HAKV,11X,3HTWC,10X,4HTWSP,9X,6HAMBLDS, FLOWF150
1 10X,2HTW,14X,2HFW,7X,6HCDSS2A/2X,3D14.5) FLOWF151
C FLOWF152
C AMRAT = RATIO ABLATION TO BOUNDARY LAYER AIR FLOWF153
C THETSI = SHOCKANGLE UNIVERSAL (RAD) FLOWF154
C FLOWF155
C FLOWF156
203 CONTINUE FLOWF157
AMRAT = AMARLD/AMBLDS FLOWF158
IF (PICIPU .GT.(19.D0/6.D0)) GO TO 7 FLOWF159
THETSI = DARSIN(1.00/AMU) FLOWF160
GO TO 8 FLOWF161
7 THETSI = DARSIN(1.00/AMU * (6.D0/7.D0*(PICIPU - 13.D0/7.D0) FLOWF162
1 **2.D0)) FLOWF163
8 THETSI = DMAX1(THETSI,THETAC) FLOWF164
C FLOWF165
C AMSS2D = MASS FLOW 2ND ENTPOPY LAYER (LRM/ SEC) FLOWF166
C R35 = NSRN FACTOR FLOWF167
C THETS2 = SHOCKANGLE MAX (RAD) FLOWF168
C THETS2B = AVERAGE SHOCK ANGLE FOR SECOND ENTPOPY LAYER (RAD) FLOWF169
C U2C = VELOCITY COME-2ND ENTPOPY LAYER (KFT / SEC) FLOWF170
C FLOWF171
AMSS2D = 2.D3*RHOU *RN**2.D0*DCCTAN(THETS1)**2*UU FLOWF172
U2C = UU - 5.02*RHOU *UU**2.D0*CDSS2A/AMSS2D FLOWF173
U2C = DMAX1(U2C,1.D0) FLOWF174
AI = .2860061.029D0/AMU**2.D0*(1.00 - (U2C/UU)**2.D0) FLOWF175
THETS2 = DARSIN(1.00/AMU * (AI & (2.86736AT)**.5D0)/2.D0)**.5D0 FLOWF176
THETS2 = DMAX1(THETS2,THETAC) FLOWF177
THETS2B = .5D0*(THETS2&THETSI) FLOWF178
R35 = C(119)*UU**C(120) FLOWF179

```

	IF (IND.NF.1) GO TO 204	FLOW 180
	WRITE (6,104) AMRAT, THE TSI, AMSS2D, U2C, A1, THE TSI, THTS2D, B35	FLOW 181
104	FORMAT (7X, 5HAMRAT, 10X, 6HTHTSI, 8X, 6HAMSS2D, 9X, 3HU2C, 11X, 2HA1,	FLOW 182
1	1 8X, 6HTHTSI, 10X, 6HTHTS2D, 9X, 3HR35/2X, RDI4.5)	FLOW 183
C		FLOW 184
C		FLOW 185
C	AMNDS = MASS FLOW NOSE (LB/ SEC)	FLOW 186
C	SRAT = RATIO BOUNDARY SWALLOWING TO NOSE MASS FLOW	FLOW 187
C		FLOW 188
204	CONTINUE	FLOW 189
	AMNDS = 7.141592654*RN**2.00*C(116)*RHOU *UU	FLOW 190
	IF (AMNDS.GE.AMNDS) GO TO 9	FLOW 191
	SRAT = AMNDS/AMNDS	FLOW 192
	GO TO 10	FLOW 193
9	SRAT = 1.00	FLOW 194
C		FLOW 195
C		FLOW 196
C	ENRN = ELECTRON DENSITY AT NOSE CAP (E/CC)	FLOW 197
C	ENRN = NO. OF ELECTRONS PRODUCED BY NOSE CAP (E/ SEC)	FLOW 198
C	H2C = ENTHALPY CONF-2ND ENTROPY LAYER (FT-SQ/ SEC-SQ)	FLOW 199
C	P2C = PRESSURE CONF-2ND ENTROPY LAYER (LB/FT-SQ)	FLOW 200
C	RHO2C = DENSITY CONF-2ND ENTROPY LAYER (LBM/FT-3)	FLOW 201
C		FLOW 202
10	CALL AR2DIM (10, B, RHO, ENRN TB, V, FRNLTB, ENRN, FRAT BL)	FLOW 203
	ENRN = 33.4900**3.00*AMNDS*SRAT*ENRN/RHOU	FLOW 204
	P2C = (7.00*AMU **2.00*OSIN(THTS2D)**2.00-1.00)*PU /6.00	FLOW 205
	H2C = (H0)*P2C*(P2C/6.00*PU)/(PU*16.00*P2C&PU)	FLOW 206
	RHO2C = RHOU*(16.00*P2C&PU)/(P2C/6.00*PU)	FLOW 207
	IF (AMNDS.GE.AMNDS) GO TO 11	FLOW 208
C		FLOW 209
C		FLOW 210
C	HC = ENTHALPY CONF (FT-SQ / SEC-SQ)	FLOW 211
C	PC = PRESSURE CONF (LB/FT-SQ)	FLOW 212
C	RHO C = DENSITY CONF (LBM/FT-3)	FLOW 213
C	SCHEM = CHEMICAL LENGTH	FLOW 214
C	UC = VELOCITY CONF (KFT / SEC)	FLOW 215
C		FLOW 216
	RHO C = RHO2C	FLOW 217
	HC = H2C	FLOW 218
	PC = P2C	FLOW 219
	UC = U2C	FLOW 220
	SCHEM = RHO2C**C(117)*SC*AMNDS/(U2C*AMSS2D)	FLOW 221
	GO TO 17	FLOW 222
11	RHO C = RHO2C	FLOW 223
	HC = H2C	FLOW 224
	PC = P2C/PU	FLOW 225
	UC = U2C/PU	FLOW 226
	SCHEM = RHO2C**C(117)*SC*AMSS2D/(U2C*AMNDS) &	FLOW 227
1	RHO C**C(117)*SC*(AMNDS-AMSS2D)/(U2C*AMNDS)	FLOW 228
C		FLOW 229
C		FLOW 230
C	FNSRN = NO. OF ELECTRONS NOSE CAP -EXP AT SHOULDER (E/ SEC)	FLOW 231
C		FLOW 232
12	FNSRN = C(122)*ENRN*AMNDS*(1.00-C(121) & (1.00&ENRN*SCHEM)	FLOW 233
1	*C(121)*DEXP(-B35 * SCHEM)/(AMNDS&C(123)*SCHEM*ENRN	FLOW 234
2	*TW**C(124))	FLOW 235
	IF (IND.NF.1) GO TO 205	FLOW 236
	WRITE (6,105) AMNDS, SRAT, ENRN, ENRN, P2C, H2C, RHO2C, RHO C	FLOW 237
105	FORMAT (7X, 5HAMNDS, 11X, 4)SRAT, 9X, 5HENRN, 9X, 4HENRN, 11X, 3HP2C,	FLOW 238
1	1 10X, 3RH2C, 11X, 5HRHO2C, 8X, 4HRHO C/2X, RDI4.5)	FLOW 239

	205 CONTINUE	FLOWF240
C		FLOWF241
C		FLOWF242
C	ENEQI = EQUILIBRIUM ELECTRON DENSITY (E/CC)	FLOWF243
C		FLOWF244
C	RHOCL2 = RHOCL / RHOCL	FLOWF245
C	HCRTO = HC/RTO	FLOWF246
C	IF (AMRAT.EQ.0.D0) GO TO 13	FLOWF247
C	ENEQH = AR3DIM(0,11, 5, 4,RHOCL2,HCRTO,AMRAT,XYZTBL)	FLOWF248
C	GO TO 14	FLOWF249
C	13 CALL AR2DIM (9,25,RHOCL2,PSTABL,HCRTO,HSTABL,ENEQH,ENTABL)	FLOWF250
C	14 CONTINUE	FLOWF251
C		FLOWF252
C		FLOWF253
C	FNBL = NO. OF ELECTRONS LEAVING BOUNDARY LAYER (E/SEC)	FLOWF254
C	ENEBL = ELECTRON DENSITY IN THE BOUNDARY LAYER (E/CC)	FLOWF255
C		FLOWF256
C	ENEBL = ENEQH*(1.D0-DEXP(-R22*SCHEM**C(11,9)))*	FLOWF257
C	1 (R21 & R23 * (UC-22.D0)/22.D0)	FLOWF258
C	FNBL = 32.4RDC**2.D0*ENEBL*AMPLD5/PC	FLOWF259
C	IF (INDNE.1) GO TO 206	FLOWF260
C	WRITE (6,106) HC,PC,LC,SCHEM,FNSRN,ENEQH,ENEBL,FNBL	FLOWF261
C	106 FORMAT (/11X,2HHC,12X,2HPC,12X,2HLC,11X,5HSCHEM,8X,5HNSRN,10X,	FLOWF262
C	1 5HENEQH,10X,5HNEBL,8X,4HFNBL/2X,9D14.5)	FLOWF263
C	206 CONTINUE	FLOWF264
C		FLOWF265
C		FLOWF266
C	AM2 = MACH NUMBER 2ND ENTROPY LAYER	FLOWF267
C	AM2C = MACH NUMBER CORE-2ND ENTROPY LAYER	FLOWF268
C	AMINF = FREE STREAM MACH NUMBER	FLOWF269
C	ENS = NO. OF ELECTRONS ENTERING WAKE NECK (E / SEC)	FLOWF270
C	H2 = ENTHALPY 2ND ENTROPY LAYER (FT-SQ / SEC-SQ)	FLOWF271
C	RHO2 = DENSITY 2ND ENTROPY LAYER (LBM/FT-3)	FLOWF272
C	U2 = VELOCITY OF SECOND ENTROPY LAYER (1000 FT/SEC)	FLOWF273
C		FLOWF274
C		FLOWF275
C	ENS = FNBL & FNSRN	FLOWF276
C	AM2C = U2C*AM2 /U2 *(HL/H2C)**.500	FLOWF277
C	AM2C = DMAX11AM2C,1.D0)	FLOWF278
C	THEFCD = THEFAC*57.29577551DC	FLOWF279
C	CALL AR2DIM (11,12,THEFCD,THITL,AM2C,ENC,HL,AM2,ETAHL)	FLOWF280
C	CALL AR2DIM (11,12,THEFCD,THITL,AMINFC,FPCTHL,AMINF,ETAHL)	FLOWF281
C	RHINF = RHOIC*((1.D0E.200*AMINFC**2.D0) / (1.D0E.200*AMINF**2.D0))	FLOWF282
C	1 **2.500	FLOWF283
C	HINF = HINFC*((1.D0E.200*AMINFC**2.D0) / (1.D0E.200*AMINF**2.D0))	FLOWF284
C	UINF = UINFC*AMINF /AMINFC*(HINF/HINFC)**.500	FLOWF285
C	RHO2 = RHOIC*((1.D0E.200*AM2C**2.D0) / (1.D0E.200*AM2**2.D0))	FLOWF286
C	1 **2.500	FLOWF287
C	H2 = H2C*((1.D0E.200*AM2C**2.D0) / (1.D0E.200*AM2**2.D0))	FLOWF288
C	U2 = U2C*AM2/AM2C*(H2/H2C)**.500	FLOWF289
C	IF (AMPLSD.GE.AMSS2D) GO TO 21	FLOWF290
C		FLOWF291
C		FLOWF292
C	AMS = MACH NUMBER SHOULDER	FLOWF293
C	HS = ENTHALPY SHOULDER (FT-SQ / SEC-SQ)	FLOWF294
C	RES = REYNOLDS NUMBER SHOULDER	FLOWF295
C	RHS = DENSITY SHOULDER (LBM/FT-3)	FLOWF296
C	US = VELOCITY SHOULDER (KFT / SEC)	FLOWF297
C		FLOWF298
C	AMS = AM2	FLOWF299
C	RHS = RHO2	FLOWF300

NOT REPRODUCIBLE

	HS = H2	FLOW 300
	US = UP	FLOW 301
	GO TO 22	FLOW 302
21	AMS = AMINF	FLOW 303
	RHIS = PHOINF	FLOW 304
	HS = HINF	FLOW 305
	US = UINF	FLOW 306
22	RFS = RHOS*AMS*RCUD*(169)/(RHOL*AMU)*(HS/HU)**C(67)	FLOW 307
	IF (IND.NF.1) GO TO 207	FLOW 308
	WRITE (6,107) ENS,AM2C,AM2,RHD2,H2,U2,AMS,RHCS	FLOW 309
107	FORMAT (/10X,3HFNS,11X,4HAM2C,10X,3HAM2,10X,4HRHC2,11X,2HH2,11X,	FLOW 310
1	2HU2,13X,3HAMS,9X,4HRHOS/2X,3D14.5)	FLOW 311
207	CONT INUF	FLOW 312
C		FLOW 313
C		FLOW 314
C	ENEN = ELECTRON DENSITY AT THE NECK (E/CC)	FLOW 315
C	HN = ENTHALPY NECK (FT-SQ / SEC-SQ)	FLOW 316
C	HNHS = RATIO NECK ENTHALPY TO SHOULDER ENTHALPY	FLOW 317
C		FLOW 318
	CS = C(164)C(165)*D SIN(THE TAC)**.2500*RF S*(-3.00/8.00)	FLOW 319
1	*AMS*(-.500)	FLOW 320
	HN = HWC C5*(HS-HWC.506*US**2.00)-DHCHEM	FLOW 321
	HNHS = 2.00*HN/(2.00*HSC1.06*US**2.00)	FLOW 322
	ENEN = PHO*ENS*HS/(30.4000**3.0)*AMHLS*HN	FLOW 323
	IF (IND.NF.1) GO TO 208	FLOW 324
	WRITE (6,108) HS,US,PF S,HN,HNHS,RHOS,ENS,EAFN	FLOW 325
108	FORMAT (/10X,2HHS,13X,2HUS,11X,3HRS,11X,2HNN,10X,5HNNHS,10X,	FLOW 326
1	4HRHS,12X,3HFNS,9X,4HFNFN/2X,3D14.5)	FLOW 327
208	CONT INUF	FLOW 328
C		FLOW 329
C		FLOW 330
C		FLOW 331
C		FLOW 332
C		FLOW 333
C		FLOW 334
C		FLOW 335
	THTLS = (COTA*RIIQU/(2.00*PI*RHOS))**.500*(UU/US)	FLOW 336
	IF (AMHLS0.00*AMSS20) GO TO 23	FLOW 337
	THETSS = (RIIQU/(2.00*PI*RHOS))*(C0VAC AMSS20*CDSS2A/AMHLS0)	FLOW 338
1	**500*(UU/US)	FLOW 339
	GO TO 24	FLOW 340
23	THETSS = THTLS	FLOW 341
24	RETS = THETSS*PE S/DO	FLOW 342
	IF (Z.GT.ZBLT) GO TO 25	FLOW 343
		FLOW 344
		FLOW 345
	AKE = GTCHEM FACTOR	FLOW 346
	R1 = DECAY RATE	FLOW 347
	RS = NFI SCALING FACTOR	FLOW 348
	H11 = ELECTRON DECAY RATE FACTOR	FLOW 349
	GT = FUNCTION AFRO	FLOW 350
		FLOW 351
	GT = 0.00	FLOW 352
	GO TO 26	FLOW 353
25	GT = (1.00-HN*(1.00-HN/HS)/(2.06*C(170)*RHOS*THETSS*US)**2.00	FLOW 354
1	*RETS*(1.00*HNHS))**(-1.00)	FLOW 355
26	RS = C(187)*(1.00*C(184)*AMPAT**C(195)C(195)*AMPAT**C(197)	FLOW 356
1	C(188)*(RHOS/PHO SL)**C(191)	FLOW 357
	R11 = C(191)C(192)*AMPAT**C(193)	FLOW 358
	H1 = (AMU)*C(115)C(111)*(RHOS/RHOSL)**C(125)/(1.06*US**2.00)*THTLS	FLOW 359

REPRODUCED FROM


```

1      ))*CCON**BCON                                FLOWF360
AKE = (THE TSS)**(-2.DC)-7.5D2P*RHOS*C(100)/ENFN    FLOWF361
IF (IND.NF.1) GO TO 209                               FLOWF362
WRITE (6,109) INTRLS,THE TSS,RETS,GT,95,011,01,AKE   FLOWF363
109  FORMAT(/7X,6HTHTBLS,1CX,4HTHE TSS,9X,4HRETS,10X,2HGT,12X,2HB5,  FLOWF364
1     11X,3HB11,14X,2HB1,10X,3HAKE/2X,8D14.5)       FLOWF365
209  CONTINUE                                         FLOWF366
IF (Z.GT.ZALT) GO TO 27                               FLOWF367
C
C
C      ENFT = TRANSITION ELECTRON DENSITY (E/CC)     FLOWF368
C      GTCHEM = FUNCTION CHEMICAL                    FLOWF369
C
C      GTCHEM = 1.00                                  FLOWF370
C      GO TO 209                                      FLOWF371
27  GTCHEM = (1.0D6MN/(4.0C*C(90)**S*AKE*RHOS*THE TSS*RETS))**(-1.0D) FLOWF372
28  ENFT = (1.0-3**C(174)*AMUO *DS*RHOS**C(133)*GT/(THE TSS*US**C(134)) FLOWF373
1   *(1.0D-DEXP(-DABS(HN))*(C(135) GC(176)*DABS(HN)**C(130) FLOWF374
2   &C(131)*DABS(HN)**C(132))) & 1.11027*RHOS*C(100) FLOWF375
3   &ENFN*GTCHEM)*CCON**ACON                          FLOWF376
IF (IND.NF.1) GO TO 210                               FLOWF377
WRITE (6,110) GTCHEM,ENFT                             FLOWF378
110  FORMAT (/7X,6HGTCHEM,11X,4HENFT/2X,2D14.5//)     FLOWF379
C
C
C      210 CONTINUE                                   FLOWF380
113  CONTINUE                                         FLOWF381
RETURN                                                 FLOWF382
END                                                    FLOWF383

```

CFMIMAX		FMIMA 1
IMPLICIT REAL*8 (A-H, O-Z)		FMIMA 2
FUNCTION FMIMAX(X,NF)		FMIMA 3
COMMON/INSTR/N,K,ALB(20),UB(20),XX(20),F1		FMIMA 4
COMMON/IQPT/IPROC,IN,NCONS,IPNT,IFX,LIMIT,IPAND		FMIMA 5
COMMON/EDT/ERR,PRAND,FAC,DELTA		FMIMA 6
C		FMIMA 7
C		FMIMA 8
C	IN THE FIBONACCI OPTIMIZATION TECHNIQUE, FUNCTION FMIMAX	FMIMA 9
C	IS SET EQUAL IN TURN TO EACH OF THE N UNIMODAL FUNCTIONS OF	FMIMA 10
C	ONE VARIABLE BEING MINIMIZED (MAXIMIZED) IN ACCORDANCE WITH	FMIMA 11
C	THE VALUE OF NF.	FMIMA 12
C		FMIMA 13
C		FMIMA 14
	GO TO (1,1,2), NF	FMIMA 15
1	CALL FFV(1,X,VAL)	FMIMA 16
	FMIMAX = VAL	FMIMA 17
	GO TO 1000	FMIMA 18
3	XX(1) = X	FMIMA 19
	CALL FFV(2,XX,VAL)	FMIMA 20
	FMIMAX = VAL	FMIMA 21
1000	RETURN	FMIMA 22
	END	FMIMA 23

CFUN1		FUN1	1
	FUNCTION FUN1(XX,N)	FUN1	2
	IMPLICIT REAL*8(A-H,O-Z)	FUN1	3
	COMMON /CSIG/ XTT,WS1,XOOP,DWST,SPH2,XLMS,XMSP,XSSP,R1,ZCR2,ZCR4,	FUN1	4
	IXB20,DWBST1,SPH4,BC,DDW,WS4,STGMDS,IND2	FUN1	5
C		FUN1	6
C		FUN1	7
C	FUNCTION FUN1 COMPUTES THE PULSE SHAPE RADAR RETURN AS A	FUN1	8
C	FUNCTION OF DISTANCE BEHIND THE LEADING EDGE OF THE RADAR	FUN1	9
C	PULSE.	FUN1	10
C		FUN1	11
	SIG1= 0.000	FUN1	12
	SIG2= 0.000	FUN1	13
	SIG3= 0.000	FUN1	14
	SIG4= 0.000	FUN1	15
	XXTT= XX - XTT	FUN1	16
C		FUN1	17
C	Y1 - REGION AFFECTED BY OVERDENSE SCATTERING	FUN1	18
	701 Y1 = DMIN1(WS1,XOOP)	FUN1	19
	Y1 = DMIN1(Y1,XXTT)	FUN1	20
	WS2 = XTT & XOOP & WS1 - XX	FUN1	21
	Y1 = DMIN1(Y1,WS2)	FUN1	22
	IF(XXTT)702,703,703	FUN1	23
	703 IF(XXTT-WS1-XOOP)704,704,702	FUN1	24
C		FUN1	25
C	SIG1 - PORTION OF PULSE SHAPE DUE TO OVERDENSE RETURN	FUN1	26
	704 SIG1 = Y1 * DWST * SPH2	FUN1	27
	702 WS2 = 2.000 * XLMS	FUN1	28
C		FUN1	29
C	Y2 - REGION AFFECTED BY FIRST PART OF MULTIPLE SCATTERING	FUN1	30
	Y2 = DMIN1(WS1,WS2)	FUN1	31
	WS2 = XXTT - XMSP	FUN1	32
	Y2 = DMIN1(WS2,Y2)	FUN1	33
	WS2 = XTT & XMSP & WS1 - XX	FUN1	34
	Y2 = DMIN1(Y2,WS2)	FUN1	35
	IF(XXTT - XOOP)705,706,706	FUN1	36
	706 IF(XXTT - WS1 - XMSP)707,707,705	FUN1	37
C		FUN1	38
C	SIG2 - PORTION OF PULSE SHAPE DUE TO CONSTANT PORTION OF	FUN1	39
C		FUN1	40
C	MULTIPLE SCATTERING REGION	FUN1	41
	707 SIG2 = 0.500 * Y2 * DWST * SPH2	FUN1	42
	705 WS3 = 0.500 * WS1	FUN1	43
	IF(WS1 - (XX - XTT - XMSP))708,709,709	FUN1	44
C		FUN1	45
C	Y31 - UPPER LIMIT OF REGION AFFECTED BY SECOND PART OF	FUN1	46
C	MULTIPLE SCATTERING	FUN1	47
	708 Y31 = XX - WS1 - XTT	FUN1	48
	GO TO 710	FUN1	49
	709 Y31 = XMSP	FUN1	50
	710 IF(XX-(XTT & XSSP))711,711,712	FUN1	51
C		FUN1	52
C	Y32 - LOWER LIMIT OF REGION AFFECTED BY SECOND PART OF	FUN1	53
C	MULTIPLE SCATTERING	FUN1	54
	711 Y32 = XXTT	FUN1	55
	GO TO 713	FUN1	56
	712 Y32 = XSSP	FUN1	57
	713 IF(WS1 - (XXTT - XSSP))714,715,715	FUN1	58
C		FUN1	59

```

C      Y41 - UPPER LIMIT OF REGION AFFECTED BY SINGLE SCATTERING      FUN1 60
715 Y41 = XSSP                                                         FUN1 61
    GO TO 716                                                         FUN1 62
714 Y41 = XXTT-WS2                                                    FUN1 63
C      Y42 - LOWER LIMIT OF REGION AFFECTED BY SINGLE SCATTERING      FUN1 64
716 Y42 = XXTT                                                         FUN1 65
    IF(XXTT - XMSO)720,721,721                                       FUN1 66
721 IF(XXTT - (WS1 & XSSP))717,717,722                               FUN1 67
717 EX1 = (DEXP(-B1* Y31) - DEXP(-B1* Y32)) * ZCR2/16,000          FUN1 69
    EX2 = (DEXP(-4.000*B1* Y31) - DEXP(-4.000*B1* Y32))           FUN1 70
C      THE FOLLOWING TEST IS USED TO PREVENT UNDERFLOW                FUN1 71
C      IF(DARS(EX2) - 1,00-A)9007,9007,9008                             FUN1 72
9007 EX2 = 0,000                                                       FUN1 73
9008 EX2 = 0,2500 * ZCR4 * EX2 * ZCR4                                 FUN1 74
    IF(XB20)719,719,718                                              FUN1 75
C      SIG2 - PORTION OF PULSE SHAPE DUE TO VARIABLE PORTION OF      FUN1 76
C      MULTIPLE SCATTERING REGION                                     FUN1 77
719 SIG2 = DWSST1 * (EX1 & EX2) & (Y32 - Y31) * SPH4 * B0 * CDW     FUN1 78
    GO TO 720                                                         FUN1 79
718 SIG2 = DWSST1 * (EX1 & EX2) & 2,000 * WS4 * (DEXP(-0,500 * B1 *   FUN1 80
    1 XH20*Y31) - DEXP(-0,500 * B1 * XH20*Y32)) * B0 * SPH2**12,000-  FUN1 81
    1 XH20)/(B1*XH20) * DWSST1                                       FUN1 82
720 IF(XXTT - XSSP)730,722,722                                       FUN1 83
722 EX1 = DEXP(-4,000*B1* Y41) - DEXP(-4,000*B1* Y42)             FUN1 84
    EX2 = DEXP(-B1* Y41) - DEXP(-B1* Y42)                           FUN1 85
C      SIG4 - PORTION OF RETURN DUE TO SINGLE SCATTERING REGION      FUN1 86
C      SIG4 = 0,2500*DWSST1* ZCR4*EX1*ZCR4 & DDW/B1 *(SPH2/16,000)  FUN1 87
    1 *DWSST/DDW & B0) *ZCR2 * EX2                                  FUN1 88
C      SIG - PULSE SHAPE - RADAR RETURN AS A FUNCTION OF DISTANCE    FUN1 89
C      BEHIND THE LEADING EDGE OF THE RADAR PULSE                   FUN1 90
730 SIG = SIG1 & SIG2 & SIG3 & SIG4                                  FUN1 91
    IF(IND2-1)30005,10005,30005                                       FUN1 92
10005 WRITE (6,20005) Y1,SIG1,Y2,SIG2,Y31                            FUN1 93
20005 FORMAT(10X,7H   Y1=,1PE15,7,6H   SIG1=,1PE15,7,6H   Y2=,1PE15,7,10H  FUN1 94
    1   SIG2=,1PE15,7,6H   Y31=,1PE15,7)                             FUN1 95
10006 WRITE (6,20006) Y32,Y41,Y42,SIG3,SIG4                          FUN1 96
20006 FORMAT(10X,7H   Y32=,1PE15,7,6H   Y41=,1PE15,7,6H   Y42=,1PE15,7,10H  FUN1 97
    1   SIG3=,1PE15,7,6H   SIG4=,1PE15,7)                             FUN1 98
10010 WRITE (6,20010) XX,SIG                                          FUN1 99
20010 FORMAT(2X,20X=,1PE15,7,6H   SIG=,1PE15,7)                     FUN1 100
30005 GO TO (100,101),N                                              FUN1 101
C      FUN1 = SIG                                                    FUN1 102
F      RETURN                                                         FUN1 103
C      FUN1 = (SIGMS - SIG)**2                                        FUN1 104
C      RETURN                                                         FUN1 105
END                                                                     FUN1 106

```

INOT REPRODUCIBLE

CF 123		F123	1
	SUBROUTINE F123(XXX)	F123	2
	IMPLICIT REAL*8 (A-H,O-Z)	F123	3
C		F123	4
C		F123	5
C	SUBROUTINE F123 CAN OPTIONALLY - (1) READ DATA FOR A	F123	6
C	REENTRY VEHICLE TRAJECTORY FROM INPUT OR CALL SUBROUTINE VIXEN	F123	7
C	TO CALCULATE THIS DATA. (2) CALL VIXEN TO CALCULATE THE DATA	F123	8
C	FOR A DECDY TRAJECTORY. (3) COMPARE THE DATA FOR A REENTRY	F123	9
C	VEHICLE TRAJECTORY WITH THE DATA FOR A DECDY TRAJECTORY AND	F123	10
C	SAVE THE DIFFERENCES. (4) CALL INTGR1 TO CALCULATE AND PLOT	F123	11
C	A CORRIDOR INTEGRAL. (5) CALL INTGR1 TO CALCULATE THE SIGMA	F123	12
C	SQUARE INTEGRAL TO BE USED BY SUBROUTINE EFFECT. (6) CALL	F123	13
C	INTGR1 TO CALCULATE A TABLE OF INFLUENCE COEFFICIENTS AND TO	F123	14
C	PRODUCE INFLUENCE COEFFICIENT PLOTS AT SPECIFIED ALTITUDES.	F123	15
C		F123	16
C		F123	17
	COMMON OCCUR,NOCCUR	F123	18
	COMMON /CICCUR/ ICOCUR(1320)	F123	19
	COMMON /PCOCUR/ PCOCUR(11770)	F123	20
	COMMON /TTMCOM/ TTM	F123	21
	COMMON /INFTST/ INDOOY	F123	22
	REAL * 8 LA1, LA2, MOOT, MW, MX, MY, MZ	F123	23
	DIMENSION ALMAX(200),ALMIN(200),BCA(40),BCD(40),BCV(40),BCWL1(40)	F123	24
	DIMENSION BCWL2(40),BCWL3(40),BCWR1(40),BCWR2(40),BCWR3(40)	F123	25
	DIMENSION TBMAT(3), TT(200)	F123	26
	DIMENSION BETAP1(160),BS(160),COFINF(2,2,2),DR(160),DD(160)	F123	27
	DIMENSION DS(160),DV(160),DVAR(50),DVH(50),DVL(50),DWL1(160)	F123	28
	DIMENSION DWL2(160),DWL3(160),DWR1(160),DWR2(160),DWR3(160)	F123	29
	DIMENSION FMAX(200),FMIN(200),H(40),HOLD(3,50),ICP(90),MOOT(4,8)	F123	30
	DIMENSION NOCOMDV(50),NOCCUR(30),NPV(160),OCCUR(4000),PEPS(160)	F123	31
	DIMENSION QDOT(4,8),SAVE(9,160,3),SR(40),SD(40),SV(40),SWL1(40)	F123	32
	DIMENSION SWL2(40),SWL3(40),SWR1(40),SWR2(40),SWR3(40),TAMAX(200)	F123	33
	DIMENSION TMIN(200),TCR(40),TCO(40),TCV(40),TCWL1(40),TCWL2(40)	F123	34
	DIMENSION TCWL3(40),TCWR1(40),TCWR2(40),TCWR3(40),TM(140)	F123	35
	DIMENSION TPLDT(160),TS(160),TTM(200),VPLCT(160),VS(160)	F123	36
	DIMENSION VDGPLT(160),WL1P(160),WL1S(160),WL2P(160),WL2S(160)	F123	37
	DIMENSION WL2P(160),WL3S(160),WR1P(160),WR1S(160),WR2P(160)	F123	38
	DIMENSION WR2S(160),WR3P(160),WR3S(160),ZMAX(200),ZMIN(200)	F123	39
	DIMENSION ZPLDT(160),ZS(160)	F123	40
	EQUIVALENCE (OCCUR(00229), ALALI)	F123	41
	EQUIVALENCE (OCCUR(01044), ALMAX)	F123	42
	EQUIVALENCE (OCCUR(01244), ALMIN)	F123	43
	EQUIVALENCE (OCCUR(00002), ALPHA)	F123	44
	EQUIVALENCE (OCCUR(00003), ALPRIM)	F123	45
	EQUIVALENCE (OCCUR(06122), ALST)	F123	46
	EQUIVALENCE (OCCUR(00128), CASE)	F123	47
	EQUIVALENCE (OCCUR(00059), COB)	F123	48
	EQUIVALENCE (OCCUR(02793), COFINF)	F123	49
	EQUIVALENCE (OCCUR(00100), CDI)	F123	50
	EQUIVALENCE (OCCUR(00058), COP)	F123	51
	EQUIVALENCE (OCCUR(00101), CRPO)	F123	52
	EQUIVALENCE (OCCUR(00202), CM)	F123	53
	EQUIVALENCE (OCCUR(00203), CN)	F123	54
	EQUIVALENCE (OCCUR(00054), CODLAM)	F123	55
	EQUIVALENCE (OCCUR(00216), CPSZET)	F123	56
	EQUIVALENCE (OCCUR(00215), CTHZET)	F123	57
	EQUIVALENCE (OCCUR(00127), DATE)	F123	58
	EQUIVALENCE (OCCUR(00235), DECDY)	F123	59



NOT REPRODUCIBLE

EQUIVALENCE (OCCUR (00236), DECF TC)	F123	60
EQUIVALENCE (OCCUR (00234), DEFCOP)	F123	61
EQUIVALENCE (OCCUR (00219), DELY)	F123	62
EQUIVALENCE (OCCUR (00220), DELZ)	F123	63
EQUIVALENCE (OCCUR (00129), FMO)	F123	64
EQUIVALENCE (OCCUR (00197), FACTR)	F123	65
EQUIVALENCE (OCCUR (00244), FMAX)	F123	66
EQUIVALENCE (OCCUR (02446), FMIN)	F123	67
EQUIVALENCE (OCCUR (00175), GAMF)	F123	68
EQUIVALENCE (OCCUR (00020), GAMMA)	F123	69
EQUIVALENCE (OCCUR (00139), LA1)	F123	70
EQUIVALENCE (OCCUR (00144), LA2)	F123	71
EQUIVALENCE (OCCUR (00709), MDOT)	F123	72
EQUIVALENCE (OCCUR (00117), MW)	F123	73
EQUIVALENCE (OCCUR (00210), NX)	F123	74
EQUIVALENCE (OCCUR (00211), MY)	F123	75
EQUIVALENCE (OCCUR (00212), MZ)	F123	76
EQUIVALENCE (OCCUR (4000), LPO)	F123	77
EQUIVALENCE (OCCUR (00280), PEPSB)	F123	78
EQUIVALENCE (OCCUR (00112), PHIO)	F123	79
EQUIVALENCE (OCCUR (00223), PSIZET)	F123	80
EQUIVALENCE (OCCUR (00114), PSIG)	F123	81
EQUIVALENCE (OCCUR (02676), QDOT)	F123	82
EQUIVALENCE (OCCUR (00110), QG)	F123	83
EQUIVALENCE (OCCUR (00060), RBDOT)	F123	84
EQUIVALENCE (OCCUR (00136), RB1)	F123	85
EQUIVALENCE (OCCUR (00142), RB2)	F123	86
EQUIVALENCE (OCCUR (00059), RBDOT)	F123	87
EQUIVALENCE (OCCUR (00135), RN1)	F123	88
EQUIVALENCE (OCCUR (00141), RN2)	F123	89
EQUIVALENCE (OCCUR (00116), SIG)	F123	90
EQUIVALENCE (OCCUR (00221), SINGO)	F123	91
EQUIVALENCE (OCCUR (00111), SMRO)	F123	92
EQUIVALENCE (OCCUR (00218), SPSZET)	F123	93
EQUIVALENCE (OCCUR (00217), STIIZET)	F123	94
EQUIVALENCE (OCCUR (01444), TAMAX)	F123	95
EQUIVALENCE (OCCUR (01644), TAMIN)	F123	96
EQUIVALENCE (OCCUR (03643), TBMAT)	F123	97
EQUIVALENCE (OCCUR (00201), TH)	F123	98
EQUIVALENCE (OCCUR (00113), THFALO)	F123	99
EQUIVALENCE (OCCUR (00224), THEZET)	F123	100
EQUIVALENCE (OCCUR (00246), TZTST)	F123	101
EQUIVALENCE (OCCUR (00086), WOOT)	F123	102
EQUIVALENCE (OCCUR (01844), ZMAX)	F123	103
EQUIVALENCE (OCCUR (02044), ZMIN)	F123	104
EQUIVALENCE (PCCUR (06181), BCB)	F123	105
EQUIVALENCE (PCCUR (06101), BCD)	F123	106
EQUIVALENCE (PCCUR (06021), BCY)	F123	107
EQUIVALENCE (PCCUR (06501), BCWL1)	F123	108
EQUIVALENCE (PCCUR (06581), BCWL2)	F123	109
EQUIVALENCE (PCCUR (06661), BCWL3)	F123	110
EQUIVALENCE (PCCUR (06281), BCWR1)	F123	111
EQUIVALENCE (PCCUR (06341), BCWR2)	F123	112
EQUIVALENCE (PCCUR (06421), BCWR3)	F123	113
EQUIVALENCE (PCCUR (06641), BTAPL)	F123	114
EQUIVALENCE (PCCUR (03841), BS)	F123	115
EQUIVALENCE (PCCUR (02081), DP)	F123	116
EQUIVALENCE (PCCUR (01921), DD)	F123	117
EQUIVALENCE (PCCUR (07681), DS)	F123	118
EQUIVALENCE (PCCUR (01761), DV)	F123	119

EQUIVALENCE (PCCUR(11707), DVAR)	F123 120
EQUIVALENCE (PCCUR(11607), DVH)	F123 121
EQUIVALENCE (PCCUR(11657), DVL)	F123 122
EQUIVALENCE (PCCUR(02721), DWL1)	F123 123
EQUIVALENCE (PCCUR(02981), DWL2)	F123 124
EQUIVALENCE (PCCUR(03041), DWL3)	F123 125
EQUIVALENCE (PCCUR(02241), DWR1)	F123 126
EQUIVALENCE (PCCUR(02401), DWR2)	F123 127
EQUIVALENCE (PCCUR(02561), DWR3)	F123 128
EQUIVALENCE (PCCUR(05841), H)	F123 129
EQUIVALENCE (PCCUR(07101), HOLD)	F123 130
EQUIVALENCE (PCCUR(07251), SAVF)	F123 131
EQUIVALENCE (PCCUR(06821), SF)	F123 132
EQUIVALENCE (PCCUR(06781), SP)	F123 133
EQUIVALENCE (PCCUR(06741), SV)	F123 134
EQUIVALENCE (PCCUR(06591), SWL1)	F123 135
EQUIVALENCE (PCCUR(07021), SWL2)	F123 136
EQUIVALENCE (PCCUR(07061), SWL3)	F123 137
EQUIVALENCE (PCCUR(06861), SWR1)	F123 138
EQUIVALENCE (PCCUR(06901), SWR2)	F123 139
EQUIVALENCE (PCCUR(06941), SWR3)	F123 140
EQUIVALENCE (PCCUR(06221), TCR)	F123 141
EQUIVALENCE (PCCUR(06141), TCR)	F123 142
EQUIVALENCE (PCCUR(06061), TCV)	F123 143
EQUIVALENCE (PCCUR(06541), TCWL1)	F123 144
EQUIVALENCE (PCCUR(06621), TCWL2)	F123 145
EQUIVALENCE (PCCUR(06701), TCWL3)	F123 146
EQUIVALENCE (PCCUR(06701), TCWR1)	F123 147
EQUIVALENCE (PCCUR(06381), TCWR2)	F123 148
EQUIVALENCE (PCCUR(06461), TCWR3)	F123 149
EQUIVALENCE (PCCUR(05561), TT)	F123 150
EQUIVALENCE (PCCUR(03201), TS)	F123 151
EQUIVALENCE (PCCUR(03521), VS)	F123 152
EQUIVALENCE (PCCUR(00001), TPLOT)	F123 153
EQUIVALENCE (PCCUR(00481), VCPLOT)	F123 154
EQUIVALENCE (PCCUR(00321), VPLOT)	F123 155
EQUIVALENCE (PCCUR(01281), WL1P)	F123 156
EQUIVALENCE (PCCUR(01441), WL2P)	F123 157
EQUIVALENCE (PCCUR(01601), WL3P)	F123 158
EQUIVALENCE (PCCUR(04481), WL1S)	F123 159
EQUIVALENCE (PCCUR(04641), WL2S)	F123 160
EQUIVALENCE (PCCUR(04801), WL3S)	F123 161
EQUIVALENCE (PCCUR(00801), WR1P)	F123 162
EQUIVALENCE (PCCUR(00961), WR2P)	F123 163
EQUIVALENCE (PCCUR(01121), WR3P)	F123 164
EQUIVALENCE (PCCUR(04001), WR1S)	F123 165
EQUIVALENCE (PCCUR(04161), WR2S)	F123 166
EQUIVALENCE (PCCUR(04321), WR3S)	F123 167
EQUIVALENCE (PCCUR(00161), ZPLOT)	F123 168
EQUIVALENCE (PCCUR(03361), ZS)	F123 169
EQUIVALENCE (PCCUR(00030P), II)	F123 170
EQUIVALENCE (PCCUR(00301), IREF)	F123 171
EQUIVALENCE (PCCUR(00001), IOP)	F123 172
EQUIVALENCE (PCCUR(00031), JJ)	F123 173
EQUIVALENCE (PCCUR(00313), K)	F123 174
EQUIVALENCE (PCCUR(302), LPLOT)	F123 175
EQUIVALENCE (PCCUR(00303), MODE)	F123 176
EQUIVALENCE (PCCUR(00091), NCOMDV)	F123 177
EQUIVALENCE (PCCUR(000312), NCOM)	F123 178
EQUIVALENCE (PCCUR(304), NCP)	F123 179

NOT REPRODUCIBLE

	EQUIVALENCE (IOCCUR(00305), NDECOY)	F123 190
	EQUIVALENCE (IOCCUR(00306), NDVCH)	F123 191
	EQUIVALENCE (NOCCUR(00), INALPH)	F123 192
	EQUIVALENCE (NOCCUR(07), LOPT)	F123 193
	EQUIVALENCE (NOCCUR(10), RHEAT)	F123 194
	EQUIVALENCE (NOCCUR(03), MORT)	F123 195
	EQUIVALENCE (NOCCUR(05), NOSEOP)	F123 196
	EQUIVALENCE (NOCCUR(15), NGEOM)	F123 197
	EQUIVALENCE (NOCCUR(22), NTHRST)	F123 198
C		F123 199
E		F123 200
C		F123 191
	XXX = 0.00	F123 192
C		F123 193
C	CALLING THE TIMER SUBROUTINE TO RECORD THE START OF	F123 194
C	CALCULATIONS BY THE PROGRAM.	F123 195
C	100 CALL TIMER(S(1,TIMTAB))	F123 196
C		F123 197
C	400 NDC = NDECOY	F123 198
	NDVC = NDVCH	F123 199
C		F123 200
C	SAVE IOCCUR(122-149), OCCUR(205-209) AND OCCUR(222) IN DVAR(1-19)	F123 201
	CALL SAVEDV(1,DVAR,OCCUR)	F123 202
	IF(IREF.NE.2) GO TO 497	F123 203
	IF(NDECOY.EQ.1) GO TO 497	F123 204
	DO 111 I = 1, NDVC	F123 205
	NDON = NCOMOV(I)	F123 206
C	THE VALUE OF THE ITH DESIGN VARIABLE TO BE PERTURBED IS SAVED IN	F123 207
C		F123 208
C	HOLD(1,I). THIS VALUE WAS USED TO CALCULATE BASIC DECOY TRAJECTRY.	F123 209
	HOLD(1,I) = OCCUR(NDON)	F123 210
C		F123 211
C	HOLD(2,I) SAVES THE PERTURBED VALUE OF THIS VARIABLE TO BE USED FOR	F123 212
C	CALCULATING THE TRAJECTORY OF THE FIRST COMPARISON DECOY.	F123 213
	HOLD(2,I) = DVL(I)	F123 214
C		F123 215
C	HOLD(3,I) SAVES THE PERTURBED VALUE OF THIS VARIABLE TO BE USED FOR	F123 216
C	THE TRAJECTORY OF THE SECOND COMPARISON DECOY.	F123 217
	111 HOLD(3,I) = DVH(I)	F123 218
C		F123 219
C	FOR ONE PERTURBATION ONLY, THIS VALUE MUST BE SAVED IN HOLD(2,I).	F123 220
C	IF TWO PERTURBATIONS ARE MADE, THE SMALLER VALUE MUST BE SAVED IN	F123 221
C	HOLD(2,I) AND THE LARGER IN HOLD(3,I).	F123 222
	497 DO 777 II = 1,NDVC	F123 223
	IF(IREF.NE.2) GO TO 498	F123 224
	IF(NDECOY-2)401,402,403	F123 225
C		F123 226
C	A BASIC DECOY TRAJECTORY WILL BE CALCULATED.	F123 227
	401 NLOW = 1	F123 228
	NHIGH = 1	F123 229
	GO TO 404	F123 230
C		F123 231
C	ONE COMPARISON DECOY TRAJECTORY WILL BE CALCULATED.	F123 232
	402 NLOW = 2	F123 233
	NHIGH = 2	F123 234
	GO TO 404	F123 235
C		F123 236
C	TWO COMPARISON DECOY TRAJECTORIES WILL BE CALCULATED.	F123 237
	403 NLOW = 3	F123 238
	NHIGH = 3	F123 239

C		F123 240
C	OCUR(NCON) IS THE LOCATION OF THE DESIGN VARIABLE BEING PERTURBED.	F123 241
	4 6 NCON = NCONOV(II)	F123 242
C		F123 243
C	RESTORE (DVAR(1-19) TO OCCUR(133-145), OCCUR(205-209) AND OCCUR(222)	F123 244
	CALL SAVEDV(2,DVAR,OCCUR)	F123 245
	GO TO 499	F123 246
C		F123 247
C	IF IRFF EQUALS ONE A BENTLY VEHICLE TRAJECTORY WILL BE CALCULATED.	F123 248
C	IF IRFF EQUALS THREE REFERENCE VEHICLE TRAJECTORY DATA MUST BE INPUT.	F123 249
498	NLOW = 1	F123 250
	NHIGH = 1	F123 251
499	DO 444 KK = NLOW,NHIGH	F123 252
	IF(IRFF.NE.2) GO TO 501	F123 253
C		F123 254
C	PUT LOWER VALUE OF IITH OF SIGN VARIABLE TO BE PERTURBED IN OCCUR ARRAY	F123 255
	IF(KK.EQ.2) OCCUR(NCON) = DVL(II)	F123 256
C		F123 257
C	PUT UPPER VALUE OF IITH OF SIGN VARIABLE TO BE PERTURBED IN OCCUR ARRAY	F123 258
	IF(KK.EQ.3) OCCUR(NCON) = DVL(II)	F123 259
501	CONTINUE	F123 260
	IF(NCON.NE.1) CASE = CASE & .0100	F123 261
	WRITE(6,1100)CASE,DATE,EMO	F123 262
	IF(IRFF.EQ.3) GO TO 28	F123 263
C		F123 264
C	TESTING THE INPUT QUANTITIES.	F123 265
	IF(LOPT.LE.4)GO TO 53	F123 266
	WRITE(6,1052)LOPT	F123 267
	LOPT=1	F123 268
53	IF(NTHRST.LE.2)GO TO 54	F123 269
	WRITE(6,1054)NTHRST	F123 270
	NTHRST=0	F123 271
54	CONTINUE	F123 272
	IF(NGEOM.LT.1)WRITE(6,1102)NGEOM	F123 273
	IF(NCEOM.GT.3)WRITE(6,1102)NGEOM	F123 274
	IF(NGEOM.LT.1)NGEOM=1	F123 275
	IF(NCEOM.GT.3)NGEOM=1	F123 276
	IF(INALPH.EQ.0)GO TO 5	F123 277
	IF((LOPT*(LOPT-2)).EQ.0)WRITE(6,1077)LOPT	F123 278
5	CONTINUE	F123 279
	IF(LOPT.NE.0)GO TO 55	F123 280
	IF(MHEAT.NE.0)WRITE(6,1055)MHEAT	F123 281
	IF(NOSEOP.NE.0)WRITE(6,1056)NOSEOP	F123 282
55	IF(MHEAT.NE.0)GO TO 56	F123 283
	IF(NOSEOP.NE.0)WRITE(6,1057)NOSEOP	F123 284
56	CONTINUE	F123 285
C		F123 286
C	DEFINING NUMERICAL FACTORS.	F123 287
	COPLAM=7.6489D-9*MW/(SIG*SIG)	F123 288
	FACTR9=SQRT(GAMMA)	F123 289
	ALAL1=D.CDC	F123 290
C		F123 291
C	SETTING INPUT OPTIONS EQUAL TO A STORAGE QUANTITY IN ORDER	F123 292
C	TO CARRY THEM OVER TO THE FOLLOWING INPUT CASE, UNLESS THEY ARE	F123 293
C	REDEFINED.	F123 294
	LOPTC=LOPT	F123 295
	NTHRTC=NTHRST	F123 296
	INALPCT=INALPH	F123 297
C		F123 298
C	CHANGING THE UNITS FROM INCHES (INPUT UNITS) TO FEET (UNITS)	F123 299

NOT REPRODUCIBLE

NOT REPRODUCIBLE

C	USED INTERNALLY).	F123 300
	RN1=RN1/12.000	F123 301
	RH1=RH1/12.000	F123 302
	LA1=LA1/12.000	F123 303
	RN2=RN2/12.000	F123 304
	RH2=RH2/12.000	F123 305
	LA2=LA2/12.000	F123 306
	DEL Z =DELZ/12.000	F123 307
	DELY =DELY/12.000	F123 308
C		F123 309
C	SETTING THE DRAG COEFFICIENT QUANTITIES TO ZERO.	F123 310
	CDFINF(2,1,1)=0.00	F123 311
	CDFINF(2,2,1)=0.00	F123 312
	CDFINF(2,1,2)=0.00	F123 313
	CDFINF(2,2,2)=0.00	F123 314
	CDR=0.00	F123 315
	CDP=0.00	F123 316
	CDPL=0.00	F123 317
	CDI=0.00	F123 318
	CDLCP=0.000	F123 319
	CDGDP=0.000	F123 320
	CDGFC=0.000	F123 321
	TZTFST=0.000	F123 322
C		F123 323
C	SETTING THE ROTATIONAL QUANTITIES TO ZERO.	F123 324
	IF(LOPT.EQ.0)GO TO 201	F123 325
	CR=0.000	F123 326
	CN=0.000	F123 327
	PSIC=0.000	F123 328
	CR=0.000	F123 329
	SMPC=0.000	F123 330
	PHIC=0.000	F123 331
	IF(LOPT.EQ.1)THEALC=0.000	F123 332
201	CONTINUE	F123 333
	IF(LOPT.NE.0)GO TO 205	F123 334
	DO 200 J=1,200	F123 335
	TANX(J)=0.00	F123 336
	ZMAX(J)=0.00	F123 337
	TMAX(J)=0.00	F123 338
	ALMAX(J)=0.00	F123 339
	TMIN(J)=0.00	F123 340
	ZMIN(J)=0.00	F123 341
	TMIN(J)=0.00	F123 342
200	ALMIN(J)=0.00	F123 343
205	CONTINUE	F123 344
C		F123 345
C	FINDING THE TRIGONOMETRIC FUNCTIONS OF THE THRUST OFFSET ANGLES.	F123 346
	CTH=CT=COS(0.017453290C*THEZET)	F123 347
	STH=ST=SIN(0.017453290C*THEZET)	F123 348
	CPZET=COS(0.017453290C*PSIZET)	F123 349
	SPZET=SIN(0.017453290C*PSIZET)	F123 350
C		F123 351
C	CHANGING THE UNITS OF ANGLES FROM DEGREES TO RADIAN.	F123 352
	GAMF=0.017453290C*GAMF0	F123 353
	SING0=DSIN(GAMF0)	F123 354
	PHIC=0.017453290C*PHIC	F123 355
	PSI=0.017453290C*PSI0	F123 356
	THEALJ=0.017453290C*THEAL0	F123 357
	ALST=0.017453290C*ALST	F123 358
		F123 359

NOT REPRODUCIBLE

C		F123 360
C	SETTING THE MASS LOSS AND HEATING QUANTITIES TO ZERO.	F123 361
	QDOT(1,1)=0.0DC	F123 362
	QDOT(4,1)=0.0DC	F123 363
	DO 3 J=1,3	F123 364
	MDOT(1,J)=0.0DC	F123 365
	MDOT(2,J)=0.0DC	F123 366
	MDOT(3,J)=0.0DC	F123 367
	MDOT(4,J)=0.0DC	F123 368
	PEP8(J)=0.0DC	F123 369
	QDOT(2,J)=0.0DC	F123 370
	QDOT(3,J)=0.0DC	F123 371
C		F123 372
	WDOT=0.0DC	F123 373
	RNDOT=0.0DC	F123 374
	RMDOT=0.0DC	F123 375
C		F123 376
C	CALLING IN THE TIMER SUBROUTINE TO RECORD THE CALLING IN OF	F123 377
C	SUBROUTINE VIXEN.	F123 378
	CALL TIMERS(2,TIMTAB)	F123 379
C		F123 380
C	CALLING IN SUBROUTINE VIXEN WHICH PERFORMS THE CALCULATIONS.	F123 381
	CALL VIXEN	F123 382
C		F123 383
C	PERFORM THE CALCULATIONS CALLED FOR BY VIXEN.	F123 384
C	CALLING IN THE TIMER SUBROUTINE TO MEASURE THE TIME TO	F123 385
C	CALCULATE THE TRAJECTORY.	F123 386
	CALL TIMERS(-2,TIMTAB)	F123 387
C		F123 388
C		F123 389
C	CALLING IN THE TIMER SUBROUTINE TO MEASURE THE TOTAL TIME TO	F123 390
C	READ IN THE INPUT QUANTITIES AND PERFORM THE CALCULATIONS.	F123 391
	CALL TIMERS(-1,TIMTAB)	F123 392
C		F123 393
C	SETTING THE THRUST QUANTITIES TO ZERO.	F123 394
	TH=0.0DC	F123 395
	MX=0.0DC	F123 396
	MY=0.0DC	F123 397
	MZ=0.0DC	F123 398
	TBMAT(1)=0.0DC	F123 399
	TBMAT(2)=0.0DC	F123 400
	TBMAT(3)=0.0DC	F123 401
C		F123 402
C	CHANGING UNITS ON THE GEOMETRIC QUANTITIES FROM FEET BACK TO	F123 403
C	INCHES.	F123 404
	RN1=RN1*12.0DC	F123 405
	RH1=RH1*12.0DC	F123 406
	LA1=LA1*12.0DC	F123 407
	RN2=RN2*12.0DC	F123 408
	LA2=LA2*12.0DC	F123 409
	R02=1.37*12.0DC	F123 410
	DELY=DELY*12.0DC	F123 411
	DELZ=DELZ*12.0DC	F123 412
C		F123 413
C	CHANGING UNITS FROM RADIAN TO DEGREE TO PREPARE FOR NEW	F123 414
C	CASE.	F123 415
	ALST=ALST/0.01745329DC	F123 416
C		F123 417
C	SETTING THE OPTIONS EQUAL TO THE STORAGE INPUT VALUES TO	F123 418
C	PREPARE FOR A NEW CASE.	F123 419

NOT REPRODUCIBLE

NOT REPRODUCIBLE

```

      INALPM=INALPC
      NTHRST=NTHRTC
      LUPT=LUPTC
      IF (INT.EQ.2) GO TO 3C
21  CALL HEADER(3)
C
C SAVE REENTRY VEHICLE TRAJECTORY DATA FOR POSSIBLE COMPARISON WITH THE
C TRAJECTORY OF A DECOY.
      GO 22 I = 1, LPLOT
      ZS(I) = ZPLOT(I)
      TS(I) = TPLOT(I)
      IF (IOP(73).EQ.0) GO TO 21
      IF (IOP(64).EQ.1) VS(I) = VPLOT(I)
      IF (IOP(65).EQ.1) DS(I) = VOGPLT(I)
      IF (IOP(66).EQ.1) BS(I) = BETAPL(I)
21  IF (IOP(74).EQ.0) GO TO 22
      IF (IOP(67).EQ.1) WL1S(I) = WL1P(I)
      IF (IOP(68).EQ.1) WL2S(I) = WL2P(I)
      IF (IOP(69).EQ.1) WL3S(I) = WL3P(I)
      IF (IOP(70).EQ.1) WP1S(I) = WP1P(I)
      IF (IOP(71).EQ.1) WR2S(I) = WR2P(I)
      IF (IOP(72).EQ.1) WR3S(I) = WR3P(I)
22  CONTINUE
C
      WRITE(6,1000)(ZS(I),TS(I),VS(I),DS(I),BS(I),I=1,LPLOT)
      IF (IOP(74).EQ.1) WRITE(6,1001)(ZS(I),TS(I),WP1S(I),WR2S(I),WR3S(I),
1  WL1S(I),WL2S(I),WL3S(I), I=1,LPLOT)
      LP=LP
      IF (LP.LT.6) GO TO 4C7
C
C INTEGRATION ROUTINE HAS FAILED. RESTORE DESIGN CHARACTERISTIC VALUES
C TO OCCUR ARRAY THEN TRY NEXT RV OR DECOY.
      WRITE(6,406) LP
407  CONTINUE
      CALL SAVEDV(2,5VAR,0CCLR)
      RETURN
3.  CALL HEADER(LOW)
C
C THE DIFFERENCES BETWEEN THE REENTRY VEHICLE AND THE DECOY
C QUANTITIES - V, V DOT OVER G, BETA, WAKE LENGTH AND RADAR
C CROSS SECTION FOR THE INPUT RADAR FREQUENCIES.
      GO 33 I = 1, LPLOT
      IF (IOP(73).EQ.0) GO TO 31
      IF (IOP(64).EQ.1) DV(I) = VS(I) - VPLOT(I)
      IF (IOP(65).EQ.1) DV(I) = DS(I) - VOGPLT(I)
      IF (IOP(66).EQ.1) DV(I) = BS(I) - BETAPL(I)
31  IF (IOP(74).EQ.0) GO TO 33
      IF (IOP(67).EQ.1) DWL1(I) = WL1S(I) - WL1P(I)
      IF (IOP(68).EQ.1) DWL2(I) = WL2S(I) - WL2P(I)
      IF (IOP(69).EQ.1) DWL3(I) = WL3S(I) - WL3P(I)
      IF (IOP(70).EQ.1) DWR1(I) = WP1S(I) - WP1P(I)
      IF (IOP(71).EQ.1) DWR2(I) = WR2S(I) - WR2P(I)
      IF (IOP(72).EQ.1) DWR3(I) = WR3S(I) - WR3P(I)
33  CONTINUE
C
      WRITE(6,1000)(ZPLOT(I),TPLOT(I),VPLOT(I),VOGPLT(I),BETAPL(I),I=1,
1  LPLOT)
      IF (IOP(74).EQ.1) WRITE(6,1001)(ZPLOT(I),TPLOT(I),WP1P(I),WR2P(I),
1  WR3P(I),WL1P(I),WL2P(I),WL3P(I),I=1,LPLOT)
      LP=LP

```

```

      IF(LP.LT.6)GO TO 507
C
C INTEGRATION ROUTINE HAS FAILED. RESTORE DESIGN CHARACTERISTIC VALUES
C TO OCCUR ARRAY THEN TRY NEXT PV OR DECOY.
      WRITE(6,406) LP
      CALL SAVEDV(2,DVAR,DOCLR)
      WRITE(6,4666) IDV(I),DD(I),DB(I),I=1,LPLOT)
4666  FORMAT(1H,8X7HDELTA,VRX7HDELTA,DBX7HDELTA,R/(1H,1P3F15.7))
      GO TO 777
507  CONTINUE
C
C THE SEQUENCE OF STATEMENTS ENDING AT 50 COMBINES THE H AND ZPLT ARRAY
C THE ZPLT ARRAY HAS THE ALTITUDES AT WHICH TRAJECTORY DATA IS OUTPUT.
C THE TPLOT ARRAY HAS THE CORRESPONDING TIMES.
C THE H ARRAY HAS THE ALTITUDES AT WHICH COPRINCOR VALUES ARE INPUT.
C TO FORM THE TT ARRAY, THEN THE TIME CORRESPONDING TO EACH H IS FOUND BY
C LINEAR INTERPOLATION AND COMBINED WITH THE TPLOT ARRAY TO GIVE THE
C TT ARRAY OF TIMES CORRESPONDING TO THE TT ALTITUDE ARRAY.
      JB = 2
      K = 0
      DO 50 I = 1, LPLOT
      JA = JB
      DO 44 J = JA, NCP
      K = K & 1
      IF(ZPLOT(I)-H(J)) 35,26,37
35   JB = J & 1
      TT(K) = H(J)
      CALL LINFIT(2,2,ZPLOT(I),TPLOT(I),WK,XX)
      TIM(K)=6W*H(J)GX
      DO 10 44
36   JB = J & 1
      TT(K) = ZPLOT(I)
      TIM(K)=TPLOT(I)
      GO TO 51
37   TT(K) = ZPLOT(I)
      TIM(K)=TPLOT(I)
      GO TO 51
44   CONTINUE
50   CONTINUE
C
95   DO 333 JJ = 1,LPLOT
      SAVE(1,JJ,KK) = DV(JJ)
      SAVE(2,JJ,KK) = DD(JJ)
      SAVE(3,JJ,KK) = DP(JJ)
      SAVE(4,JJ,KK) = DWL1(JJ)
      SAVE(5,JJ,KK) = DWL2(JJ)
      SAVE(6,JJ,KK) = DWL3(JJ)
      SAVE(7,JJ,KK) = DWR1(JJ)
      SAVE(8,JJ,KK) = DWR2(JJ)
      SAVE(9,JJ,KK) = DWR3(JJ)
333  CONTINUE
C
C
      NDECOY = 1
      IF(KK.EQ.NFIGH.AND.NDECOY.EQ.2) NDECOY = 2
      IF(KK.EQ.NFIGH.AND.NDECOY.EQ.3) NDECOY = 3
C
C INCLUDE CAN PRODUCE AND OUTPUT FIRST AND SECOND DIFFERENCES BETWEEN
C BASIC DECOY DATA AND COMPARISON DECOY DATA SAVED IN DC 333 LOOP
C AT SPECIFIED ALTITUDES AND PRODUCE SCATTER PLOTS OF THE SAVED

```

```

F123 490
F123 481
F123 482
F123 483
F123 484
F123 485
F123 486
F123 487
F123 488
F123 489
F123 490
F123 491
F123 492
F123 493
F123 494
F123 495
F123 496
F123 497
F123 498
F123 499
F123 500
F123 501
F123 502
F123 503
F123 504
F123 505
F123 506
F123 507
F123 508
F123 509
F123 510
F123 511
F123 512
F123 513
F123 514
F123 515
F123 516
F123 517
F123 518
F123 519
F123 520
F123 521
F123 522
F123 523
F123 524
F123 525
F123 526
F123 527
F123 528
F123 529
F123 530
F123 531
F123 532
F123 533
F123 534
F123 535
F123 536
F123 537
F123 538
F123 539

```

NOT REPRODUCIBLE

C DIFFERENCES VS THE VALUE OF THE DESIGN VARIABLE BEING VARIED	F123 540
C THERE WILL BE THREE POINTS ON EACH GRAPH.	F123 541
IF(IOP(73),FO,0) GO TO 600	F123 542
IF(IOP(64),FO,1)	F123 543
ICALL INTGRL(1,2,1,3515,3557,RCV,OV,TCV,SV,RH VFLCCTY)	F123 544
IF(IOP(65),FO,1)	F123 545
ICALL INTGRL(2,3,4,3516,3554,RCO,DD,TCO,SO,RH DFCELRN)	F123 546
IF(IOP(66),FO,1)	F123 547
ICALL INTGRL(3,3,7,3517,3555,RCO,DD,TCO,SO,RH BALLSTC)	F123 548
600 IF(IOP(74),FO,0) GO TO 444	F123 549
IF(IOP(67),FO,1)	F123 550
ICALL INTGRL(22,6,10,3541,3556,RCWL1,OWL1,TCWL1,SWL1,RH WAKE L1)	F123 551
IF(IOP(68),FO,1)	F123 552
ICALL INTGRL(23,6,13,3542,3557,RCWL2,OWL2,TCWL2,SWL2,RH WAKE L2)	F123 553
IF(IOP(69),FO,1)	F123 554
ICALL INTGRL(24,6,16,3543,3558,RCWL3,OWL3,TCWL3,SWL3,RH WAKE L3)	F123 555
IF(IOP(70),FO,1)	F123 556
ICALL INTGRL(25,6,19,3544,3559,RCWR1,OWR1,TCWR1,SWR1,RH WAKE R1)	F123 557
IF(IOP(71),FO,1)	F123 558
ICALL INTGRL(26,6,22,3545,3560,RCWR2,OWR2,TCWR2,SWR2,RH WAKE R2)	F123 559
IF(IOP(72),FO,1)	F123 560
ICALL INTGRL(27,6,25,3546,3561,RCWR3,OWR3,TCWR3,SWR3,RH WAKE R3)	F123 561
444 CONTINUE	F123 562
C	F123 563
777 CONTINUE	F123 564
C	F123 565
C	F123 566
C	F123 567
CALL SAVEDV(2,DVAR,DCCLR)	F123 569
RETURN	F123 569
C	F123 570
406 FORMAT(1H0 'CASE FAILED. TRY NEXT DESIGN VARIABLE' ,1P =',I12)	F123 571
700 FORMAT(1H0 'REENTRY VEHICLE CHARACTERISTICS')	F123 572
900 FORMAT(1H0 'DECUY CHARACTERISTICS')	F123 573
1000 FORMAT(F123 574
11H07X'AL TITUDE'11X'TIME'7X'VELOCITY'3X'DCCFLERATION'11X'BETA'/	F123 575
201H F15.1,4F15.2)	F123 576
1001 FORMAT(1H07X8HAL TITUDE11X4H TIME8X7HWAKE R18X7HWAKE R28X7HWAKE R3	F123 577
2 8X7HWAKE L18X7HWAKE L28X7HWAKE L3/1H F15.1,F15.2,1P3E15.7,	F123 578
3 1P3E15.2))	F123 579
1000 FORMAT(1H1' INTEGRAL OF',1X,AP,' =',1PE16.7)	F123 580
1100 FORMAT(1H110X'CASE',F9.2,5X'DATE',F7.2,5X'MFMC',F7.1,5X'2542E')	F123 581
1 12 FORMAT(1H0 10X,24H*****WARNING***** NGEOM=13,23H IS THE SAME AS NGEOM=11	F123 582
1053 FORMAT(1H0 10X,*****WARNING***** LOPT=',13,' IS NOT ALLOWED, LOPT	F123 583
1 HAS BEEN SET EQUAL TO 1')	F123 584
1054 FORMAT(1H0 10X,*****WARNING***** NTHRUST=',13,' IS NOT ALLOWED,	F123 585
INTHRUST HAS BEEN SET EQUAL TO 0')	F123 586
1055 FORMAT(1H0 10X,35H*****WARNING***** MOPT=0 AND MHEAT=13,23H IS THE	F123 587
1 SAME AS MHEAT=0)	F123 588
1056 FORMAT(1H0 10X,35H*****WARNING***** MOPT=0 AND NCSECP=13,24H IS THE	F123 589
1F SAME AS NCSECP=0)	F123 590
1057 FORMAT(1H0 10X,37H*****WARNING***** MHEAT=0 AND NCSECP=13,24H IS THE	F123 591
1FE SAME AS NCSECP=C)	F123 592
1077 FORMAT(1H0 10X,*****WARNING***** / 11X,'INPUT ALPHA OPTIGN OVERKE	F123 593
1RICES LOPT=',13,' ROTATIONAL CALCULATIONS')	F123 594
END	F123 595

NOT REPRODUCIBLE

CGIMAX	IMPLICIT REAL*8 (A-H,O-Z)	GIMAX 1
	SUBROUTINE GIMAX(AA,BR,NF,NMIMAX,ACCUR,NFUNC,XMIMAX,YMIMAX)	GIMAX 2
	IW=6	GIMAX 3
	DIMENSION E(100)	GIMAX 4
	COMMON /END/ ITERM	GIMAX 5
C		GIMAX 6
C		GIMAX 7
C		GIMAX 8
C	SUBROUTINE GIMAX, THE CONTROLLING SUBROUTINE FOR THE TWO	GIMAX 9
C	VARIABLE FIBONACCI SEARCH TECHNIQUE, OPTIMIZES THE SECONDARY	GIMAX 10
C	INDEPENDENT VARIABLE OF THE FUNCTION. AT EACH EVALUATION POINT	GIMAX 11
C	IN THIS OPTIMIZATION, GIMAX SUMMONS FUNCTION GMIMAX WHICH	GIMAX 12
C	CALLS FOR THE OPTIMIZATION OF THE PRIMARY VARIABLE.	GIMAX 13
C		GIMAX 14
C		GIMAX 15
	A=AA	GIMAX 16
	B=BR	GIMAX 17
	IF (NMIMAX) 10,11,11	GIMAX 18
10	JMIMAX=1	GIMAX 19
	GO TO 20	GIMAX 20
11	JMIMAX=2	GIMAX 21
20	IF (B-A) 21,22,23	GIMAX 22
21	TEMP=A	GIMAX 23
	A=B	GIMAX 24
	B=TEMP	GIMAX 25
	GO TO 23	GIMAX 26
22	L=1	GIMAX 27
25	WRITE (IW,26) L	GIMAX 28
26	FORMAT(17H0 ERROR OF TYPE I2)	GIMAX 29
	ITERM = 1	GIMAX 30
	GO TO 170	GIMAX 31
23	R=B-A	GIMAX 32
	IF (NFUNC) 31,30,35	GIMAX 33
30	IF (ACCUR) 31,31,40	GIMAX 34
31	L=2	GIMAX 35
	GO TO 25	GIMAX 36
35	NFUNC2=NFUNC&2	GIMAX 37
	R1=(1.0D0&SQRT(5.0D0))/2.0D0	GIMAX 38
	R2=(1.0D0-SQRT(5.0D0))/2.0D0	GIMAX 39
	ROACC=(R1**NFUNC2-R2**NFUNC2)/(R1-R2)	GIMAX 40
	GO TO 45	GIMAX 41
40	ROACC=R/ACCUR	GIMAX 42
45	IF (ROACC-2.0D0) 50,50,51	GIMAX 43
50	XMIMAX=(A&R)/2.0D0	GIMAX 44
	YMIMAX=GMIMAX(XMIMAX,NF)	GIMAX 45
	GO TO 170	GIMAX 46
51	IF (ROACC-3.0D0) 60,60,61	GIMAX 47
60	XL=A&(B-A)/3.0D0	GIMAX 48
	XR=A&2.0D0*(B-A)/3.0D0	GIMAX 49
	YL=GMIMAX(XL,NF)	GIMAX 50
	IF(ITERM,NF,C) GO TO 160	GIMAX 51
	YR=GMIMAX(XR,NF)	GIMAX 52
	IF(ITERM,NF,U) GO TO 150	GIMAX 53
	IF(YL-YR) 70,70,71	GIMAX 54
70	XMIMAX=XR	GIMAX 55
	YMIMAX=YR	GIMAX 56
	GO TO 170	GIMAX 57
71	XMIMAX=XL	GIMAX 58
	YMIMAX=YL	GIMAX 59

NOT REPRODUCIBLE

CO TO 170	GIMAX 60
61 K=3	GIMAX 61
F(1)=1.000	GIMAX 62
F(2)=1.000	GIMAX 63
74 F(K)=F(K-1)G(K-2)	GIMAX 64
IF (PIACC-F(K)) 80,80,75	GIMAX 65
75 IF (K-40) 77,76,76	GIMAX 66
76 L=3	GIMAX 67
CO TO 25	GIMAX 68
77 K=K&1	GIMAX 69
CO TO 74	GIMAX 70
80 N=K	GIMAX 71
XL=AGE(N-2)*R/F(N)	GIMAX 72
XR=AGE(N-1)*R/F(N)	GIMAX 73
YL=GIMAX(XL,NF)	GIMAX 74
IF(ITERM.NF.0) GO TO 160	GIMAX 75
YR=GIMAX(XR,NF)	GIMAX 76
IF(ITERM.NF.0) GO TO 150	GIMAX 77
J=1	GIMAX 78
90 NJ=N-J	GIMAX 79
NJINF=N-J-1	GIMAX 80
NJTWO=N-J-2	GIMAX 81
GO TO (100,101), JMINAX	GIMAX 82
100 IF (YR-YL) 130,130,110	GIMAX 83
101 IF (YR-YL) 110,110,130	GIMAX 84
110 IF (J-N&3) 120,160,160	GIMAX 85
120 R=XR	GIMAX 86
A=A	GIMAX 87
R=R-A	GIMAX 88
XR=XL	GIMAX 89
XL=AGE(NJTWO)*R/E(NJ)	GIMAX 90
YR=YL	GIMAX 91
YL=GIMAX(XL,NF)	GIMAX 92
IF(ITERM.NE.0) GO TO 160	GIMAX 93
12* J=J&1	GIMAX 94
CO TO 90	GIMAX 95
130 IF (J-N&3) 140,150,150	GIMAX 96
140 R=B	GIMAX 97
A=XL	GIMAX 98
R=B-A	GIMAX 99
XL=XR	GIMAX 100
XR=AGE(INJONE)*R/E(NJ)	GIMAX 101
YL=YR	GIMAX 102
YR=GIMAX(XR,NF)	GIMAX 103
IF(ITERM.NE.0) GO TO 150	GIMAX 104
CO TO 125	GIMAX 105
150 XMIMAX=XR	GIMAX 106
YMIMAX=YR	GIMAX 107
CO TO 170	GIMAX 108
160 XMIMAX=XL	GIMAX 109
YMIMAX=YL	GIMAX 110
170 RETURN	GIMAX 111
END	GIMAX 112

GMIMAX	GMIMA	1
IMPL ICIT REAL*8 (A-H, O-Z)	GMIMA	2
FUNCTION GMIMAX(X, NF)	GMIMA	3
COMMON /END/ ITERM	GMIMA	4
COMMON /XXSAVE/ X1SAVE(40), X2SAVE(40), KOUNT	GMIMA	5
COMMON /MINSK/ N, K, ALB(20), UB(20), XX(2), F1	GMIMA	6
COMMON /IOPT/ IPROC, IN, NCONS, IPNT, ICX, LIMIT, IRAND	GMIMA	7
COMMON /EOP/ ERR, PRAND, FAC, DELTA	GMIMA	8
C	GMIMA	9
C	GMIMA	10
C	GMIMA	11
C	GMIMA	12
C	GMIMA	13
C	GMIMA	14
C	GMIMA	15
2	GMIMA	16
XX(2) = X	GMIMA	17
CALL MIMAX(ALB(1), UB(1), 3, -1, ERR, LIMIT, XX(1), VAL)	GMIMA	18
GMIMAX = VAL	GMIMA	19
KOUNT = KOUNT + 1	GMIMA	20
IF(KOUNT.GT.40) ITERM = 1	GMIMA	21
IF(KOUNT.GT.40) RETURN	GMIMA	22
X1SAVE(KOUNT) = XX(1)	GMIMA	23
X2SAVE(KOUNT) = XX(2)	GMIMA	24
1000 RETURN	GMIMA	25
END	GMIMA	25

CCRAM	IMPLICIT REAL*8 (A-H,O-Z)	GRAM	1
	SUBROUTINE GRAM	GRAM	2
	COMMON/IOPTR/IPROC,NMAX,NCONS,IPNT,LEX,LIMIT,IRAND	GRAM	3
	COMMON/HBK0/HH(160),X(40),D(40),F(40),DP(40),A(40),	GRAM	4
	I(40),GS,GSP,GTP,GSS,GTT,GSD,FZZ,F9,F9,FC,Z,PZZ,TC,RS,SL,ZZZ,	GRAM	5
	1QZZ,AXX,FL,	GRAM	6
	2DELXX,FAK,CXX(40,10)	GRAM	7
	COMMON OCCUR,NUCCUR	GRAM	8
	DIMENSION OCCUR(4000),NOCCUR(30),OB(20,20),C(20,20)	GRAM	9
	EQUIVALENCE(HH(1),BB(1)),(HH(401),C(1))	GRAM	10
C		GRAM	11
C		GRAM	12
C		GRAM	13
C	SUBROUTINE GRAM USES THE GRAM-SCHMIDT ORTHOGONALIZATION	GRAM	14
C	PROCESS. GIVEN A MATRIX C AND A COLUMN VECTOR D, THIS ROUTINE	GRAM	15
C	DEFINES A SET OF VECTORS, A, WHICH ARE USED TO CALCULATE AN	GRAM	16
C	ORTHOGONAL SET OF VECTORS, B, TO BE USED AS THE ROW VECTORS OF	GRAM	17
C	A MATRIX BB. THE INPUT MATRIX C IS THEN REDEFINED AS THE	GRAM	18
C	INVERSE (TRANSPOSE) OF THE MATRIX BB.	GRAM	19
C		GRAM	20
C		GRAM	21
	DO 1 II = 1,NMAX	GRAM	22
	A(II) = 0.00	GRAM	23
	B(II) = 0.00	GRAM	24
	DO 1 JJJ = 1,NMAX	GRAM	25
	BB(II,JJJ) = 0.00	GRAM	26
1	CONTINUE	GRAM	27
C		GRAM	28
C		GRAM	29
	DO 107 I=1,NMAX	GRAM	30
C		GRAM	31
	DO 101 L=1,NMAX	GRAM	32
	A(L) = 0.00	GRAM	33
	DO 101 N=1,NMAX	GRAM	34
	A(L)=A(L)+C(L,N)*D(N)	GRAM	35
101	CONTINUE	GRAM	36
C		GRAM	37
	DO 102 L=1,NMAX	GRAM	38
	IF(I-1) 111,111,112	GRAM	39
111	B(L)=A(L)	GRAM	40
	GO TO 102	GRAM	41
112	IL=I-1	GRAM	42
	DO 103 J=1,IL	GRAM	43
	AB = 0.00	GRAM	44
	DO 104 K=1,NMAX	GRAM	45
	AB=ABGA(K)*OB(J,K)	GRAM	46
104	CONTINUE	GRAM	47
	B(L)=A(L)-AB*BB(J,L)	GRAM	48
103	CONTINUE	GRAM	49
C		GRAM	50
102	CONTINUE	GRAM	51
C		GRAM	52
	BMAGS = 0.00	GRAM	53
	DO 106 L=1, NMAX	GRAM	54
	BMAGS=BMAGS+(L**2)	GRAM	55
106	CONTINUE	GRAM	56
	BMAG=SQRT(BMAGS)	GRAM	57
C		GRAM	58
C		GRAM	59

	DO 107 L=1,NMAX	GRAM 60
	BB(I,L)=B(L)/BMAG	GRAM 61
107	CONTINUE	GRAM 62
C		GRAM 63
C		GRAM 64
C	C MATRIX IS INVERSE OF BB MATRIX. THIS INVERSE IS THE TRANSPOSE	GRAM 65
	DO 108 I=1,NMAX	GRAM 66
	DO 108 J=1, NMAX	GRAM 67
	C(J,I)=BB(I,J)	GRAM 68
108	CONTINUE	GRAM 69
C		GRAM 70
C		GRAM 71
	WRITE(6,5)	GRAM 72
5	FORMAT(1H018HC MATRIX FROM GRAM)	GRAM 73
C		GRAM 74
	DO 3 I = 1,NMAX	GRAM 75
2	WRITE(6,2) (C(I,J),J=1,NMAX)	GRAM 76
2	FORMAT(1H 6(F20.10))	GRAM 77
C		GRAM 78
	RETURN	GRAM 79
	END	GRAM 80

C	HEADER		HEADF 1
	IMPLICIT REAL*(A-H,O-Z)		HEADF 2
	SUBROUTINE HEADER(K)		HEADF 3
	COMMON OCCUR, NOCCUR		HEADF 4
	DIMENSION OCCUR(4000), NOCCUR(30)		HEADF 5
	EQUIVALENCE(OCCUR(4000),LP)		HEADF 6
C			HEADF 7
C			HEADF 8
C	SUBROUTINE HEADER PRINTS THE TYPE OF VEHICLE WHOSE TRAJECTORY		HEADF 9
C	IS BEING PROCESSED ALONG WITH THE VALUES OF NINETEEN DESIGN		HEADF 10
C	PARAMETERS, THE VALUE OF NGEOM, AND THE VALUE OF LP, AN ERROR		HEADF 11
C	TEST PARAMETER.		HEADF 12
C			HEADF 13
C			HEADF 14
	K = K		HEADF 15
	GO TO(10, 20, 30), K		HEADF 16
10	WRITE(6,700)		HEADF 17
	GO TO 40		HEADF 18
20	WRITE(6,800)		HEADF 19
	GO TO 40		HEADF 20
30	WRITE(6,900)		HEADF 21
40	CONTINUE		HEADF 22
	WRITE(6,1000)(OCCUR(I),I=133,145),(OCCUR(I),I=205,209),OCCUR(222),		HEADF 23
	NOCCUR(15),LP		HEADF 24
1000	FORMAT(1H0* W1 THETA1 RN1 RB1 L		HEADF 25
	1AMDA1 LA1 W2 THETA2 RN2 RB2*		HEADF 26
	2/1H 10F12.2/1H		HEADF 27
	* LAMDA2 LA2 ZTURN ZON		HEADF 28
	3 ZOFF TH0 TON TOFF ISP NGEOM		HEADF 29
	4 LP*1H 2F12.2,1PE12.4,CP6F12.2,216)		HEADF 30
700	FORMAT(1H0 *BASIC DECOY CHARACTERISTICS*)		HEADF 31
800	FORMAT(1H0 *COMPARISON DECOY CHARACTERISTICS*)		HEADF 32
900	FORMAT(1H0 *REFERENCE REENTRY VEHICLE CHARACTERISTICS*)		HEADF 33
	RETURN		HEADF 34
	END		HEADF 35

CINFCOF

	SUBROUTINE INFCOF(DC,IZ,XQ)	INFCD 1
	IMPLICIT REAL*8(A-H,O-Z)	INFCD 2
	REAL*4 DC, XQ	INFCD 3
	COMMON OCCUR, NOCCUR	INFCD 4
	COMMON /GOCUR/ POCUP(11770)	INFCD 5
	COMMON /GICUP/ IOCCUP(320)	INFCD 6
	DIMENSION DC(2),H(3,50),IZ(19),NOCCUR(3),OCCUR(4000)	INFCD 7
	DIMENSION XQ(40)	INFCD 8
	DIMENSION S(9,160,3), T(160), Z(160)	INFCD 9
	EQUIVALENCE (POCUP(07101), H)	INFCD 10
	EQUIVALENCE (POCUP(07251), S)	INFCD 11
	EQUIVALENCE (POCUP(00001), T)	INFCD 12
	EQUIVALENCE (POCUP(00161), Z)	INFCD 13
	EQUIVALENCE (IOCCUP(000309), II)	INFCD 14
	EQUIVALENCE (IOCCUP(000310), JJ)	INFCD 15
	EQUIVALENCE (IOCCUP(000305), NDECOY)	INFCD 16
	EQUIVALENCE (IOCCUP(000302), LPLDT)	INFCD 17
	EQUIVALENCE (IOCCUP(000312), NCON)	INFCD 18
C		INFCD 19
C		INFCD 20
C	SUBROUTINE INFCOF CALCULATES A TABLE OF INFLUENCE COEFFI-	INFCD 21
C	CIENTS AND POSSIBLY SECOND DERIVATIVES AT EACH ALTITUDE FOR	INFCD 22
C	WHICH VIXEN STORES TRAJECTORY INFORMATION.	INFCD 23
C		INFCD 24
C		INFCD 25
C		INFCD 26
	DO 27 I = 1, 19	INFCD 27
	JP = 1	INFCD 28
	IF (IZ(I),EQ,NCON) GO TO 31	INFCD 29
27	CONTINUE	INFCD 30
C		INFCD 31
	JP = 0	INFCD 32
31	CONTINUE	INFCD 33
C		INFCD 34
	JP = JP & JP	INFCD 35
	X21 = H(2,II)-H(1,II)	INFCD 36
	IF(NDECOY,EQ,2) GO TO 100	INFCD 37
	X32 = H(3,II)-H(1,II)	INFCD 38
	X21 = (H(3,II)-H(2,II))/2.DC	INFCD 39
	DO TO 500	INFCD 40
100	AMID = .5DC*(H(1,II)&H(2,II))	INFCD 41
	IF(JP,EQ,0)GO TO 102	INFCD 42
	WRITE(6,1000, XQ(JP-1) , XQ(JP), (H(K,II),K=1,2) , AMID	INFCD 43
	DO TO 103	INFCD 44
102	WRITE(6,1050)NCON , (H(K,II),K=1,2) ,AMID	INFCD 45
103	WRITE(6,1100) DC	INFCD 46
	WRITE(6,1200)	INFCD 47
	DO 222 I = 1, LPLDT	INFCD 48
	XA = S(JJ, I, 2) - S(JJ, I, 1)	INFCD 49
	XA = XA/X21	INFCD 50
170	WRITE(6,1300)Z(I), T(I), (S(JJ, I, K), K=1, 2), XA	INFCD 51
222	CONTINUE	INFCD 52
	RETURN	INFCD 53
C		INFCD 54
C		INFCD 55
500	AMID = .5DC*(H(1,II)&H(2,II))	INFCD 56
	CMID = .5DC*(H(1,II)&H(3,II))	INFCD 57
	CYID = .5DC*(H(2,II)&H(3,II))	INFCD 58
	IF (JP,EQ,0)GO TO 502	INFCD 59

```

WRITE(6,950) X0(JP-1) , X0(JP) INFO 60
DO TO 504 INFO 61
502 WRITE(6,1025) NCON INFO 62
504 WRITE(6,1400) ((H(K,I),K=1,3),AMID,BMID,CMID INFO 63
503 WRITE(6,1100) DC INFO 64
WRITE(6,1500) INFO 65
C INFO 66
C INFO 67
DO 555 I = 1, LPLOT INFO 68
XA = S(JJ,I,2)-S(JJ,I,1) INFO 69
XB = S(JJ,I,3) -S(JJ,I,1) INFO 70
XA = XA/X21 INFO 71
XB = XB/X32 INFO 72
XC = (XB-XA)/X31 INFO 73
530 WRITE(6,1600)Z(I),T(I),((S(JJ,I,K),K=1,3),XA,XB,XC) INFO 74
555 CONTINUE INFO 75
RETURN INFO 76
C INFO 77
C INFO 78
717 FORMAT(16A4) INFO 79
950 FORMAT(1H 2A4, 6HVARIABLES 32X 1CHVALUES ARE 33X 14HMIDPOINTS ARE ) INFO 80
1025 FORMAT(1H 18, 6 VARIES, 32X VALUES ARE, 33X MIDPOINTS ARE) INFO 81
1050 FORMAT(1H 01P, 4 VARIES VALUES ARE, 4X F15.2, 1 MIDPCINT IS, F15.2) INFO 82
1000 FORMAT(1H 02A4, 17HVARIABLES VALUES ARE 2F15.2, 1X 11HMIDPCINT IS 1X , INFO 83
1F15.2) INFO 84
1100 FORMAT(1H 5HDELTA 2A4, 1X 5HTABLE) INFO 85
1200 FORMAT(1H 7X ALTITUDE, 11X TIME, 4X FIRST DELTA, 3X SECOND DELTA, INFO 86
14X INFLU COEFF) INFO 87
1300 FORMAT(1H F15.1, 4F15.2) INFO 88
1400 FORMAT(1H 30X6F15.2) INFO 89
1500 FORMAT(1H 7X ALTITUDE, 11X TIME, 4X FIRST DELTA, 3X SECOND DELTA, INFO 90
14X THIRD DELTA FIRST INF COEF SECON INF COEF, 5X 2ND DERIV) INFO 91
1600 FORMAT(1H F15.1, 7F15.2) INFO 92
END INFO 93

```

C	INTERP	INTER	1
	SUBROUTINE INTERP(M,N,W,X,Y,Z)	INTER	2
	IMPLICIT REAL*8(A-H,O-Z)	INTER	3
	DIMENSION X(M), Y(M)	INTER	4
C		INTER	5
C		INTER	6
C	GIVEN THE TWO TABLES X AND Y(X) AND A PARTICULAR VALUE OF X	INTER	7
C	CALLED W, SUBROUTINE INTERP FINDS Z EQUAL TO Y(W) BY LINEAR	INTER	8
C	INTERPOLATION.	INTER	9
C		INTER	10
C	M IS THE DIMENSION GIVEN X AND Y IN THE CALLING PROGRAM.	INTER	11
C	X IS THE INDEPENDENT VARIABLE ARRAY. IT CONTAINS N ELEMENTS.	INTER	12
C	Y IS THE DEPENDENT VARIABLE ARRAY. IT CONTAINS N ELEMENTS.	INTER	13
C	W IS A GIVEN VALUE OF X FOR WHICH THE CORRESPONDING VALUE OF	INTER	14
C	Y IS TO BE FOUND.	INTER	15
C	Z IS THE VALUE OF Y FOUND TO CORRESPOND WITH W.	INTER	16
C		INTER	17
C	THIS VALUE IS FOUND BY LINEAR INTERPOLATION.	INTER	18
C		INTER	19
C	IF N IS LESS THAN 1 OR GREATER THAN M, AN ERROR MESSAGE IS	INTER	20
C	WRITTEN AND THE RUN IS TERMINATED.	INTER	21
C		INTER	22
C	THE ELEMENTS OF THE X ARRAY SHOULD BE EITHER MONOTONICALLY	INTER	23
C	INCREASING OR MONOTONICALLY DECREASING.	INTER	24
C		INTER	25
C	NO CHECK IS MADE FOR SUCH CONSISTENCY.	INTER	26
C		INTER	27
C	IF W IS OUTSIDE THE RANGE OF X, THE VALUE OF Z WILL BE FOUND	INTER	28
C	BY EXTRAPOLATION USING THE FIRST TWO OR THE LAST TWO POINTS	INTER	29
C	AS APPROPRIATE.	INTER	30
C		INTER	31
C	IF(N.GT.0.AND.N.LE.M) GO TO 20	INTER	32
	WRITE(6,10) M, N	INTER	33
10	FORMAT(1H0 11HDIMENSION = I12, 1X 12HTABLE SIZE = I12)	INTER	34
	CALL EXIT	INTER	35
C		INTER	36
C		INTER	37
20	IF(X(1).LT.X(2)) GO TO 50	INTER	38
	GO 22 I = 2, N	INTER	39
	IF(W.GE.X(I)) GO TO 40	INTER	40
22	CONTINUE	INTER	41
C		INTER	42
30	I = N	INTER	43
40	Z = ((Y(I)-Y(I-1))/(X(I)-X(I-1)))*(W-X(I-1)) + Y(I-1)	INTER	44
C		INTER	45
	RETURN	INTER	46
C		INTER	47
C		INTER	48
50	GO 55 I = 2, N	INTER	49
	IF(W.LE.X(I)) GO TO 40	INTER	50
55	CONTINUE	INTER	51
	GO TO 20	INTER	52
C		INTER	53
	END	INTER	54

C	INTGRL		INTGR	1
	SUBROUTINE INTGRL(KA,KB,LX,KD,KE,Q,R,S,SIG,DC)		INTGR	2
	IMPLICIT REAL*8(A-H,O-Z)		INTGR	3
	REAL * 4 DC, XQ		INTGR	4
	COMMON OCCUR,NOCCUR		INTGR	5
	COMMON /CPCUR/ PCUR(11770)		INTGR	6
	COMMON /CICUR/ IOCCUR(320)		INTGR	7
	COMMON /TTCUR/ TTM		INTGR	8
	COMMON /INFTST/ NOFCOY		INTGR	9
	DIMENSION OCCUR(4000), NOCCUR(30)		INTGR	10
	DIMENSION A(27),B(40),C(40),D(160),E(160),F(40),G(40),H(40)		INTGR	11
	DIMENSION IOP(90),IZ(15),Q(40),R(160),S(40),SIG(40),T(20)		INTGR	12
	DIMENSION ZZ(200), DC(2)		INTGR	13
	DIMENSION TPLDT(160), TTM(200), XQ(40), XX(200), YY(200), Z(160)		INTGR	14
	EQUIVALENCE (PCUR(11571), A)		INTGR	15
	EQUIVALENCE (PCUR(05841), H)		INTGR	16
	EQUIVALENCE (PCUR(11759), HDIF)		INTGR	17
	EQUIVALENCE (PCUR(05561), T)		INTGR	18
	EQUIVALENCE (PCUR(00001), TPLDT)		INTGR	19
	EQUIVALENCE (PCUR(00161), Z)		INTGR	20
	EQUIVALENCE (IOCCUR(00001), IOP)		INTGR	21
	EQUIVALENCE (IOCCUR(00031), JJ)		INTGR	22
	EQUIVALENCE (IOCCUR(00312), KX)		INTGR	23
	EQUIVALENCE (IOCCUR(00302), LPLDT)		INTGR	24
	EQUIVALENCE (IOCCUR(00304), NCP)		INTGR	25
	EQUIVALENCE (PCUR(04961), R)		INTGR	26
	EQUIVALENCE (PCUR(05001), C)		INTGR	27
	EQUIVALENCE (PCUR(05041), D)		INTGR	28
	EQUIVALENCE (PCUR(05201), F)		INTGR	29
	EQUIVALENCE (PCUR(05361), E)		INTGR	30
	EQUIVALENCE (PCUR(05401), G)		INTGR	31
	DATA 17,137,136,135,136,137,138,139,140,141,142,143,144, 145,		INTGR	32
	* 205,206,207,208,209,227/		INTGR	33
	DATA XQ/160HW1 THETA1 RN1 RB1 LAMDA1 LA1 W2		INTGR	34
	1 THETA2 RN2 RB2 LAMDA2 LA2 ZTURN ZCN ZOFF		INTGR	35
	2THD TON TOFF ISP ERROR /		INTGR	36
C			INTGR	37
C			INTGR	38
C	SUBROUTINE INTGRL PERFORMS ONE OF FOUR MAIN TASKS DEPENDING		INTGR	39
C	UPON CERTAIN INPUT OPTIONS. THESE ARE (1) TO EVALUATE A		INTGR	40
C	CORRIDOR INTEGRAL AND PRINT A TABULAR ARRAY OF DATA AT EACH		INTGR	41
C	OF A NUMBER OF ALTITUDES. (2) TO EVALUATE A SIGMA SQUARE		INTGR	42
C	INTEGRAL WHICH IS USED IN THE CALCULATIONS OF SUBROUTINE		INTGR	43
C	EFFECT. (3) TO CALL THE PLOT ROUTINE TO OBTAIN A CORRIDOR		INTGR	44
C	PLOT. (4) TO CALL THE PLOT ROUTINE TO OBTAIN AN INFLUENCE		INTGR	45
C	COEFFICIENT PLOT.		INTGR	46
C			INTGR	47
C			INTGR	48
C			INTGR	49
	JJ = (LX&2)/3		INTGR	49
	IF(IOP(KA&2*KB),EQ,0) GO TO 1120		INTGR	50
	KO = IOP(KA&4*KB)		INTGR	51
	IF(KQ,PQ,1) WRITE(6,1000) DC		INTGR	52
	JGO = 0		INTGR	53
	IF(IOP(KA&3*KB),EQ,1) JGO = 30		INTGR	54
	CALL LINFIT(40,NCP,H,Q,R,C)		INTGR	55
	CALL LINFIT(40,NCP,H,S,F,G)		INTGR	56
	CALL LINFIT(160,LPLDT,Z,P,D,E)		INTGR	57
	IX = 1		INTGR	58
	INFTST = 0		INTGR	59

	DA = 0.00	INT GR 60
	DB = 0.00	INT GR 61
	Q1 = B(1)*Z(1) & C(1)	INT GR 62
	R1 = D(1)*Z(1) & E(1)	INT GR 63
	S1 = F(1)*Z(1) & G(1)	INT GR 64
	XX(1) = Q1	INT GR 65
	YY(1) = R1	INT GR 66
	ZZ(1) = S1	INT GR 67
	IF(R1.GT.S1) NX = 2	INT GR 68
	IF(R1.LT.01) NX = C	INT GR 69
	IF(K0.EQ.0)GO TO 40	INT GR 70
	IF(NX.NE.1) IIFST = 1	INT GR 71
	IF(NX.NE.1) DP = Z(1)	INT GR 72
	IF(NX=1) IG, 20, 20	INT GR 73
10	WRITE(6,1100) Z(1), TPLOT(1),Q1, R1, S1, DA, Z(1)	INT GR 74
	GO TO 40	INT GR 75
20	WRITE(6,1100) Z(1), TPLOT(1),Q1, R1, S1, DA, Z(1)	INT GR 76
	GO TO 40	INT GR 77
30	WRITE(6,1100) Z(1), TPLOT(1), Q1, R1, S1, DA	INT GR 78
		INT GR 79
C		INT GR 80
40	NS = LPLOT -1	INT GR 81
	NR = NCP -1	INT GR 82
C		INT GR 83
		INT GR 84
	DO 1111 M = 2, KX	INT GR 85
	DO 44 K = 2, NCP	INT GR 86
	J = K -1	INT GR 87
	IF(T(N).GE.H(K)) GO TO 50	INT GR 88
44	CONTINUE	INT GR 89
	J = NR	INT GR 90
50	DO 55 K = 2, IPLOT	INT GR 91
	I = K -1	INT GR 92
	IF(T(M).GE.Z(K)) GO TO 60	INT GR 93
55	CONTINUE	INT GR 94
	I = NS	INT GR 95
60	Q2 = B(J)*T(M) & C(J)	INT GR 96
	R2 = D(I)*T(M) & E(I)	INT GR 97
	S2 = F(J)*T(M) & G(J)	INT GR 98
	XX(M) = Q2	INT GR 99
	YY(M) = R2	INT GR 100
	ZZ(M) = S2	INT GR 101
	NY = 4	INT GR 102
	IF(R2.GT.S2) NY = 7	INT GR 103
	IF(R2.LT.02) NY = 1	INT GR 104
	IGO = NX & NY	INT GR 105
C		INT GR 106
		INT GR 107
	GO TO(100,200,300,400,500,600,700,800,900),IGO	INT GR 108
100	T2 = 0.00	INT GR 109
	T1 = T(M-1)	INT GR 110
	DA = DA & ADD(LX,T1,T(M),B(J),C(J),D(I),E(I),A)	INT GR 111
	CALL WRITEM(J00&10,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 112
	GO TO 999	INT GR 113
200	T1 = T(M-1)	INT GR 114
	T2 = -(C(J)-F(I))/(D(J)-D(I))	INT GR 115
	DA = DA & ADD(LX,T2,T(M),B(J),C(J),D(I),E(I),A)	INT GR 116
	CALL WRITEM(J00&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 117
	IF(ITEST.EQ.0) GO TO 999	INT GR 118
	GO TO 999	INT GR 119
300	T2 = -(F(I)-G(J))/(D(I)-F(J))	INT GR 120

	T1 = T(M-1)	INT GR 120
	DA = DA & ADD(LX,T1,T2,D(I),F(I),F(J),G(J),A)	INT GR 121
	CALL WRITEM(JG0&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 122
	IF(ITEST.EQ.1) GO TO 350	INT GR 123
	ITEST = 1	INT GR 124
	DR = T2	INT GR 125
350	T2 = -(C(J)-E(I))/(B(J)-D(I))	INT GR 126
	DA = DA & ADD(LX,T2,T(M),B(J),C(J),D(I),F(I),A)	INT GR 127
	CALL WRITEM(JG0&30,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 128
	GO TO 999	INT GR 129
400	T1 = T(M-1)	INT GR 130
	T2 = -(C(J)-E(I))/(B(J)-D(I))	INT GR 131
	DA = DA & ADD(LX,T1,T2,B(J),C(J),D(I),E(I),A)	INT GR 132
	CALL WRITEM(JG0&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 133
	IF(ITEST.EQ.0) GO TO 550	INT GR 134
	GO TO 999	INT GR 135
500	T1 = T(M-1)	INT GR 136
	T2 = 0.00	INT GR 137
	CALL WRITEM(JG0&10,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 138
	GO TO 999	INT GR 139
600	T1 = T(M-1)	INT GR 140
	T2 = -(F(I)-G(J))/(D(I)-F(J))	INT GR 141
	DA = DA & ADD(LX,T1,T2,D(I),F(I),F(J),G(J),A)	INT GR 142
	CALL WRITEM(JG0&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 143
	IF(ITEST.EQ.0) GO TO 550	INT GR 144
	GO TO 999	INT GR 145
700	T2 = -(C(J)-E(I))/(B(J)-D(I))	INT GR 146
	T1 = T(M-1)	INT GR 147
	DA = DA & ADD(LX,T1,T2,B(J),C(J),D(I),E(I),A)	INT GR 148
	CALL WRITEM(JG0&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 149
	IF(ITEST.EQ.1) GO TO 750	INT GR 150
	ITEST = 1	INT GR 151
	DR = T2	INT GR 152
750	T2 = -(F(I)-G(J))/(D(I)-F(J))	INT GR 153
	DA = DA & ADD(LX,T2,T(M),D(I),F(I),F(J),G(J),A)	INT GR 154
	CALL WRITEM(JG0&30,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 155
	GO TO 999	INT GR 156
800	T1 = T(M-1)	INT GR 157
	T2 = -(E(I)-G(J))/(D(I)-F(J))	INT GR 158
	DA = DA & ADD(LX,T2,T(M),D(I),F(I),F(J),G(J),A)	INT GR 159
	CALL WRITEM(JG0&30,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 160
	IF(ITEST.EQ.0) GO TO 550	INT GR 161
	GO TO 999	INT GR 162
900	T1 = T(M-1)	INT GR 163
	T2 = 0.00	INT GR 164
	DA = DA & ADD(LX,T1,T(M),D(I),E(I),F(J),G(J),A)	INT GR 165
	CALL WRITEM(JG0&10,T(M),TTM(M),Q2,R2,S2,DA,T2,D(I),KQ)	INT GR 166
	GO TO 999	INT GR 167
950	ITEST = 1	INT GR 168
	DR = T2	INT GR 169
999	NX=NY/3	INT GR 170
1111	CONTINUE	INT GR 171
C		INT GR 172
C		INT GR 173
	OCOUR(KD) = DA	INT GR 174
	OCOUR(KD&KB) = DR	INT GR 175
	IF(KQ.FQ.0) WRITE(6,1900) DC, DA, DR	INT GR 176
C		INT GR 177
1120	IF(IOP(KA&KB).EQ.0) GO TO 2300	INT GR 178
	SUM = 0.00	INT GR 179

	RLO = (R(1)/SIG(1))**2	INT GP 180
	DO 2222 I = 2, NCP	INT GP 181
	CALL INTERP(160,1,PLOT,H(I),Z,P,RHI)	INT GP 182
	RHI = (RHI/SIG(I))**2	INT GP 183
	SUM = SUM - (RLO & RHI)*(H(I) - H(I-1))*500	INT GP 184
	RLO = RHI	INT GP 185
2222	CONTINUE	INT GP 186
	OCCUR(KF) = SUM	INT GP 187
	EDIF = H(1) - H(NCP)	INT GP 188
	WRITE(6,1800) DC, SUM	INT GP 189
C		INT GP 190
2300	IF(IOP(KA).EQ.0) GO TO 4000	INT GP 191
	CALL AVPLT(XQ,IZ,XX,YY,ZZ,2,DC)	INT GP 192
4000	IF(NDECOY.LT.2) RETURN	INT GP 193
	IF(IOP(KA&5*KB).EQ.1) CALL INFCDF(DC,IZ,XQ)	INT GP 194
	IF(NDECOY.LT.3) RETURN	INT GP 195
	IF(IOP(KA&6*KB).EQ.1) CALL AVPLT(XQ,IZ,T,T,T,1,DC)	INT GP 196
	RETURN	INT GP 197
C		INT GP 198
1000	FORMAT(1H 6X 3H ALTITUDE 10X 4H TIME 1X 13H LOWER CORRIDR 1X 5HRV-DE	INT GP 199
	12A4, 1X 13H UPPER CORRIDR 1X 13H INTEGRAL VALUE 1X 13H LEAVE CORRIDR	INT GP 200
	2 1X 13H ENTER CORRIDR 5X 5H SLOPE)	INT GP 201
1100	FORMAT(1H F14.1,4F14.2)	INT GP 202
1200	FORMAT(1H 1P6F15.7,15X,E15.7)	INT GP 203
1900	FORMAT(1H 2A4, 1X 11H INTEGRAL = 1P6F15.7,1X 17H LEAVE CORRIDR AT	INT GP 204
	1 F15.7)	INT GP 205
1900	FORMAT(1H 13H INTEGRAL OF (2A4,10H/SIGMA)**2 1P6F15.7)	INT GP 206
	END	INT GP 207

CJNXBES	IMPLICIT REAL *8 (A-H, O-P)	JNXBF 1
	IMPLICIT REAL *8 (R-Z)	JNXBF 2
	IMPLICIT COMPLEX * 16 (Q)	JNXBF 3
	SUBROUTINE JNXBES(ZR,ZI, NN, C, ZNR, ZNI)	JNXBF 4
	DIMENSION SR(2), SI(2), TEMP(2)	JNXBF 5
C		JNXBF 6
C		JNXBF 7
C		JNXBF 8
C	SUBROUTINE JNXBES IS CALLED BY Bessel TO CALCULATE THE BESSEL	JNXBF 9
C	FUNCTIONS OF MAXIMUM ORDER, N, AND OF N-1.	JNXBF 10
C		JNXBF 11
	QX=DCMPLX(ZR,ZI)	JNXBF 12
	Z=CDABS(QX)	JNXBF 13
	QXXI=(0.000,0.000)	JNXBF 14
	QXNU=(0.000,0.000)	JNXBF 15
C		JNXBF 16
	IF(Z= 15.999999999999999 110.11.11	JNXBF 17
C		JNXBF 18
11	QXI = (1.00, 0.00)	JNXBF 19
C	FORMULA 9.2.5 HANDBOOK OF MATH FUNCTIONS-U.S. DEPT. COMMERCE	JNXBF 20
	QCC=DCMPLX(C, 0.00)	JNXBF 21
	QXR=(0.00,0.00) * QX	JNXBF 22
	QPI=(3.141592653589793, 0.00)	JNXBF 23
	QXNU=DFLOAT(NN)	JNXBF 24
	QCHI = QX - ((0.500,0.000) * QXNU &(2.500,0.000)) * CPI	JNXBF 25
	QXMU = (4.00,0.000) * QXNU * QXNU	JNXBF 26
	QP = (1.00,0.00)	JNXBF 27
	QQ=(QXMU - (1.00,0.00)) / QXR	JNXBF 28
	QZR = QXR ** 2	JNXBF 29
	QPT = QP	JNXBF 30
C		JNXBF 31
	DO 12 I=2, 4000, 2	JNXBF 32
	QXXI=DFLOAT(I*(I-1))	JNXBF 33
	QPT = -QPT * (QXMU - QXI * QXI) * (QXMU - (QXI &(2.00,0.00))	JNXBF 34
	1**2) / QXXI / QZR	JNXBF 35
	QPNEW = QP & QPT	JNXBF 36
	PTPN = CDABS(QPT/QPNEW)	JNXBF 37
	IF(PTPN - C) 13, 14, 14	JNXBF 38
14	CONTINUE	JNXBF 39
	QP = QPNEW	JNXBF 40
12	QXI = QXI &(4.00, 0.00)	JNXBF 41
13	CONTINUE	JNXBF 42
C		JNXBF 43
	QXI = (0.00,0.00)	JNXBF 44
	QQT = QQ	JNXBF 45
	DO 15 I=3,4000, 2	JNXBF 46
	QXXI=DFLOAT(I*(I-1))	JNXBF 47
	QQT = -QQT * (QXMU - QXI * QXI) * (QXMU - (QXI &(2.00,0.00))	JNXBF 48
1	1) / QXXI / QZR	JNXBF 49
	QPNEW = QQ & QQT	JNXBF 50
	PQTON = CDABS(QQT / QPNEW)	JNXBF 51
	IF(PQTON - C) 16, 17, 17	JNXBF 52
17	CONTINUE	JNXBF 53
	QQ = QPNEW	JNXBF 54
15	QXI = QXI &(4.00,0.00)	JNXBF 55
16	CONTINUE	JNXBF 56
C		JNXBF 57
	QANS = COSQRT((2.00,0.00) / QP / QX) * (QPNEW * CDABS(QCHI)	JNXBF 58
	1-QPNEW * CDABS(QCHI))	JNXBF 59

	CALL DRLIME QANS , TEMP)	JNXDE 60
	ZNR = TEMP(1)	JNXDE 61
	ZNI = TEMP(2)	JNXDE 62
	RETURN	JNXDE 63
C		JNXDE 64
10	CONTINUE	JNXDE 65
	A=1.00	JNXDE 66
	PI 1 1=1,NN	JNXDE 67
1	A=DFLOAT(1) *A	JNXDE 68
	Z2R=Z2R/2.00	JNXDE 69
	Z2I=Z2I/2.00	JNXDE 70
	R=SQRT(Z2R**2+Z2I**2)	JNXDE 71
	THET=ATAN2(Z2I,Z2R)	JNXDE 72
	SUMP=1.00	JNXDE 73
	SUMI=0.000	JNXDE 74
	ANG=DFLOAT(NN) *THET	JNXDE 75
	R/NNA=R**NN/A	JNXDE 76
	X2NR=COS(ANG)*R/NNA	JNXDE 77
	X2NI=SIN(ANG)*R/NNA	JNXDE 78
	DIV=1.00	JNXDE 79
	SR(1) = 1.065	JNXDE 80
	SR(2) = 1.065	JNXDE 81
C		JNXDE 82
	DO 5 I=1,32000	JNXDE 83
	IF (MOD(I,2))2,3,2	JNXDE 84
3	M=1	JNXDE 85
	GO TO 4	JNXDE 86
2	M=2	JNXDE 87
4	K=2*I	JNXDE 88
	AI=DFLOAT(K)*THET	JNXDE 89
	DIV=- (DIV*R/DFLOAT(I))*R /DFLOAT(NN&I)	JNXDE 90
	SR(M)=DIV*COS(AI)	JNXDE 91
	SI(M)=DIV*SIN(AI)	JNXDE 92
	SUMP=SUMP+SR(M)	JNXDE 93
	SUMI=SUMI+SI(M)	JNXDE 94
	IF (ABS(SR(2)-SR(1))-C16,5,5	JNXDE 95
6	IF (ABS(SI(2)-SI(1))-C17,5,5	JNXDE 96
5	CONTINUE	JNXDE 97
C		JNXDE 98
	WRITE(6,1000)	JNXDE 99
1000	FORMAT (52H SERIES NOT CONVERGED IN 32000 ATTEMPTS FOR JNX BES.)	JNXDE 100
7	ZNR=X2NR*SUMP-X2NI*SUMI	JNXDE 101
	ZNI=X2NR*SUMI+X2NI*SUMP	JNXDE 102
	RETURN	JNXDE 103
	END	JNXDE 104

CLINEFIT		LINF1 1
IMPLICIT REAL*8(A-H,O-Z)		LINF1 2
SUBROUTINE LINF1(ND,NENTRY,X,Y,A,B)		LINF1 3
DIMENSION X(ND), Y(ND), A(ND), B(ND)		LINF1 4
C		LINF1 5
C		LINF1 6
C	GIVEN THE NENTRY POINTS (X,Y), SUBROUTINE LINF1 FINDS THE	LINF1 7
C	EQUATION OF THE LINE PASSING THROUGH THE POINTS (X,Y) AND	LINF1 8
C	(X(I),Y(I)) IN THE FORM Y=A(I)*X+B(I) FOR I=1,2,...,NENTRY-1.	LINF1 9
C	THESE COEFFICIENTS CAN THEN BE USED BY THE CALLING PROGRAM	LINF1 10
C	FOR MAKING A LINEAR FIT WHEN THE INTERVAL REQUIRED IS KNOWN	LINF1 11
C	IN ADVANCE.	LINF1 12
C		LINF1 13
C		LINF1 14
N = NENTRY - 1		LINF1 15
DO 77 J = 1, N		LINF1 16
A(J) = (Y(J+1)-Y(J))/(X(J+1)-X(J))		LINF1 17
B(J) = Y(J) - A(J)*X(J)		LINF1 18
77 CONTINUE		LINF1 19
1111 RETURN		LINF1 20
END		LINF1 21

```

CLNTERP                                LNTFR 1
      IMPLICIT REAL*8 (A-H,O-Z)        LNTFR 2
      SUBROUTINE LNTERP(TBATMZ,TABRHO,TABSD,UPBNDZ,DNBNDZ,Z,RHO,ASOUND) LNTFR 3
C                                         LNTFR 4
      DIMENSION TBATMZ(1),TABRHO(1),TABSD(1) LNTFR 5
C                                         LNTFR 6
C      SUBROUTINE LNTERP DOES A LOGARITHMIC INTERPOLATION TO OBTAIN LNTFR 7
C      FREE STREAM DENSITY AND A LINEAR INTERPOLATION TO OBTAIN FREE LNTFR 8
C      STREAM SOUND SPEED FROM THE INPUT TABLES WHEN Z IS LESS THAN OR LNTFR 9
C      EQUAL TO UPBNDZ AND GREATER THAN DNBNDZ. LNTFR 10
C                                         LNTFR 11
C      TABLES MUST BE INPUT WITH LOWEST VALUE OF ALTITUDE FIRST LNTFR 12
C      TABRHO IS IN LBM/FT**3 TABSD IS IN FT/SEC TBATMZ IN FEET LNTFR 13
C      IF Z IS LESS THAN UPBNDZ AND GREATER THAN DNBNDZ USE TABLES INPUT LNTFR 14
C                                         LNTFR 15
      IF(Z-UPBNDZ)10,40,30 LNTFR 16
10 IF(Z-DNBNDZ)30,30,40 LNTFR 17
40 DO 45 I=1,50 LNTFR 18
      J=I LNTFR 19
      IF(TBATMZ(I)-Z)45,46,46 LNTFR 20
45 CONTINUE LNTFR 21
      CALL EXIT LNTFR 22
C                                         LNTFR 23
46 EX=(Z-TBATMZ(J-1))/(TBATMZ(J)-TBATMZ(J-1)) LNTFR 24
      RHO=(TABRHO(J-1)*(TABRHO(J)/TABRHO(J-1))**EX)/32.174 LNTFR 25
      RATIO=(TABSD(J)-TABSD(J-1))/(TBATMZ(J)-TBATMZ(J-1)) LNTFR 26
      ASOUND=TABSD(J-1)+(Z-TBATMZ(J-1))*RATIO LNTFR 27
C      RHO IS IN SLUG/FT**3 ASOUND IS IN FT/SEC LNTFR 28
      DO TO 50 LNTFR 29
C                                         LNTFR 30
30 CALL APEDT2(Z,ASOUND,RHO) LNTFR 31
50 RETURN LNTFR 32
      END LNTFR 33

```

C	MASSLO	MASSL 1
	IMPLICIT REAL*8 (A-H,O-Z)	MASSL 2
	SUBROUTINE MASSLO	MASSL 3
C		MASSL 4
	REAL*8 LAMDA	MASSL 5
C		MASSL 6
	DIMENSION OCCUR(4000),NOCCUR(30),TW(4,8),SQDT(4,8)	MASSL 7
	1,QDOT(4,8)	MASSL 8
C		MASSL 9
	EQUIVALENCE	MASSL 10
	1(OCCUR(032),LAMDA 1),(OCCUR(068),SPD),(OCCUR(073),TWSTAG),	MASSL 11
	2(OCCUR(091),Z),(OCCUR(092),ZTR),(OCCUR(115),C),	MASSL 12
	3(OCCUR(2644),TW(1,1)),(OCCUR(2676),QDOT(1,1)),	MASSL 13
	4(OCCUR(2740),SQDT(1,1))	MASSL 14
	EQUIVALENCE	MASSL 15
	1(NOCCUR(01),JHOLD),(NOCCUR(02),KKICLD)	MASSL 16
C		MASSL 17
	COMMON OCCUR,NOCCUR	MASSL 18
C		MASSL 19
C		MASSL 20
C		MASSL 21
C		MASSL 22
C	SUBROUTINE MASSLO CONTROLS THE CALLING OF SUBROUTINE EVIL.	MASSL 23
C	MASSLO CALLS IN EVIL FOR EACH BODY STATION FOR WHICH MASS LOSS	MASSL 24
C	CALCULATIONS ARE REQUIRED.	MASSL 25
C		MASSL 26
C	J=1 STAG	MASSL 27
C	J=2 LAM	MASSL 28
C	J=3 TURB	MASSL 29
C	J=4 SONIC	MASSL 30
C		MASSL 31
C	STAGNATION POINT CALCULATIONS.	MASSL 32
	J=1	MASSL 33
	K=1	MASSL 34
	ASSIGN 7 TO NEXT	MASSL 35
	QDOT(1,1)=QDOT(1,1)*C	MASSL 36
	GO TO 100	MASSL 37
	2 TWSTAG=TW(1,1)	MASSL 38
	QDOT(1,1)=QDOT(1,1)/C	MASSL 39
	IF(Z.LT.ZTR)GO TO 3	MASSL 40
	IF(LAMDA.GT.1,C0-3)GO TO 4	MASSL 41
C		MASSL 42
C	LAMINAR FLOW, SHARP CONE AT MAXIMUM DIAMETER .	MASSL 43
	J=2	MASSL 44
	K=8	MASSL 45
	ASSIGN 200 TO NEXT	MASSL 46
	GO TO 100	MASSL 47
C		MASSL 48
C	LAMINAR FLOW, BLUNT CONE CALCULATIONS.	MASSL 49
	4 J=2	MASSL 50
	K=0	MASSL 51
	20 K=K&1	MASSL 52
	IF(K.EQ.9)GO TO 200	MASSL 53
	ASSIGN 20 TO NEXT	MASSL 54
	GO TO 100	MASSL 55
C		MASSL 56
	3 IF(LAMDA.GT.1,C0-3)GO TO 5	MASSL 57
C		MASSL 58
C	TURBULENT FLOW, CALCULATIONS AT MAX. DIAMETER POINT OF A	MASSL 58
C	SHARP CONE .	MASSL 59

	J=7	MASSL 60
	K=8	MASSL 61
	ASSIGN 200 TO NEXT	MASSL 62
	GO TO 100	MASSL 63
C		MASSL 64
C	TURBULENT FLOW, BLUNT CONE CALCULATIONS.	MASSL 65
C	SONIC POINT TURBULENT CALCULATIONS.	MASSL 66
5	J=4	MASSL 67
	K=1	MASSL 68
	ASSIGN 6 TO NEXT	MASSL 69
	GO TO 100	MASSL 70
C	TANGENT POINT AND CONICAL FRUSTUM POINTS TURBULENT FLOW.	MASSL 71
6	J=3	MASSL 72
	K=0	MASSL 73
30	K=K&1	MASSL 74
	IF(K.EQ.9)GO TO 200	MASSL 75
	ASSIGN 20 TO NEXT	MASSL 76
	GO TO 100	MASSL 77
200	RETURN	MASSL 78
C		MASSL 79
C	CALLING IN SUBROUTINE EVIL TO PERFORM MASS LOSS RATE	MASSL 80
C	CALCULATION.	MASSL 81
100	JJHOLD=J	MASSL 82
	KKHOLD=K	MASSL 83
	CALL EVIL(ZZ)	MASSL 84
	TX(J,K)=ZZ	MASSL 85
	SDOT(J,K)=SPD	MASSL 86
C		MASSL 87
	GO TO NEXT.(2,200,20,6,30)	MASSL 88
	END	MASSL 89

C	MATM	1
	IMPLICIT REAL*8 (A-H,O-Z)	MATM 2
	SUBROUTINE MATMP (M,N,H,G,S)	MATM 3
	DIMENSION G(40), S(40)	MATM 4
	DIMENSION H(40,40)	MATM 5
	MATRIX MULTIPLICATION	MATM 6
C		MATM 7
C		MATM 8
C	SUBROUTINE MATMP MULTIPLIES THE VECTOR G ON THE LEFT BY THE	MATM 9
C	MATRIX H TO PRODUCE A NEW VECTOR S.	MATM 10
C		MATM 11
C		MATM 12
	702 DO 703 I=1,M	MATM 13
	700 S(I)=0.0	MATM 14
	DO 703 J=1,N	MATM 15
	703 S(I)=H(J,I)*G(J)+S(I)	MATM 16
	704 RETURN	MATM 17
	END	MATM 18

MATMPY		MATMP	1
	IMPLICIT REAL*8 (A-H,O-Z)	MATMP	2
	SUBROUTINE MATMPY(A,B,C)	MATMP	3
	DIMENSION A(3),B(3,3),C(3)	MATMP	4
C		MATMP	5
C	SUBROUTINE MATMPY MULTIPLIES A THREE BY THREE MATRIX BY A	MATMP	6
C	COLUMN MATRIX HAVING THREE COMPONENTS.	MATMP	7
C		MATMP	8
	DO 1 I=1,3	MATMP	9
	A(I)=0.000	MATMP	10
	DO 1 J=1,3	MATMP	11
	1 A(I)=A(I)+B(I,J)*C(J)	MATMP	12
	RETURN	MATMP	13
	END	MATMP	14

C	MAXMIN		MAXMI	1
	IMPLICIT REAL *8(A-H,O-Z)		MAXMI	2
	SUBROUTINE MAXMIN(NTABLE,NDIMEN,X,XMIN,XMAX)		MAXMI	3
	DIMENSION X(NDIMEN)		MAXMI	4
C			MAXMI	5
C	SUBROUTINE MAXMIN FINDS THE LARGEST AND SMALLEST VALUES IN		MAXMI	6
C	THE ARRAY, WHICH CONTAINS NTABLE NUMBERS. THE SMALLEST VALUE		MAXMI	7
C	IS PLACED IN LOCATION XMIN AND THE LARGEST IN LOCATION XMAX.		MAXMI	8
C			MAXMI	9
C			MAXMI	10
	IF(NDIMEN.GE.NTABLE) GO TO 20		MAXMI	11
	WRITE(6,10) NDIMEN, NTABLE		MAXMI	12
	CALL EXIT		MAXMI	13
20	XMIN = X(1)		MAXMI	14
	XMAX = X(1)		MAXMI	15
	IF(NTABLE.EQ.1) RETURN		MAXMI	16
C			MAXMI	17
	DO 77 J = 2, NTABLE		MAXMI	18
	IF(X(J).LT.XMIN) XMIN = X(J)		MAXMI	19
	IF(X(J).GT.XMAX) XMAX = X(J)		MAXMI	20
77	CONTINUE		MAXMI	21
C			MAXMI	22
1111	RETURN		MAXMI	23
10	FORMAT(1H112NDIMENSION IS 112, 1X 13NTABLE SIZE IS 112/1H 4HEXIT)		MAXMI	24
	END		MAXMI	25

CMIMAX		MIMAX 1
	IMPLICIT REAL*8 (A-H,O-Z)	MIMAX 2
	SUBROUTINE MIMAX (AA,BB,NF,NMIMAX,ACCUR,NFUNC,XMIMAX,YMIMAX)	MIMAX 3
	IW=6	MIMAX 4
	DIMENSION E(100)	MIMAX 5
	COMMON /END/ ITERM	MIMAX 6
C		MIMAX 7
C		MIMAX 8
C	SUBROUTINE MIMAX UTILIZES A FIBONACCI SEARCH TECHNIQUE TO	MIMAX 9
C	FIND THE MAXIMUM OR MINIMUM OF A ONE VARIABLE UNIMODAL	MIMAX 10
C	FUNCTION WITHIN A DEFINED REGION (AA,BB).	MIMAX 11
C		MIMAX 12
C		MIMAX 13
	A=AA	MIMAX 14
	B=BB	MIMAX 15
	IF (NMIMAX) 10,11,11	MIMAX 16
10	JMIMAX=1	MIMAX 17
	GO TO 20	MIMAX 18
11	JMIMAX=2	MIMAX 19
20	IF (B-A) 21,22,23	MIMAX 20
21	TEMP=A	MIMAX 21
	A=B	MIMAX 22
	B=TEMP	MIMAX 23
	GO TO 23	MIMAX 24
22	L=1	MIMAX 25
C		MIMAX 26
25	WRITE (IW,26) L	MIMAX 27
26	FORMAT(17H0 ERROR OF TYPE 12)	MIMAX 28
	ITERM = 1	MIMAX 29
	GO TO 170	MIMAX 30
23	R=B-A	MIMAX 31
	IF (NFUNC) 31,30,35	MIMAX 32
30	IF (ACCUR) 31,31,40	MIMAX 33
31	L=2	MIMAX 34
	GO TO 25	MIMAX 35
35	NFUNC2=NFUNC/2	MIMAX 36
	R1=(1.0005*SQRT (5.0001))/2.000	MIMAX 37
	R2=(1.000-SQRT (5.0001))/2.000	MIMAX 38
	ROACC=(R1**NFUNC2-R2**NFUNC2)/(R1-R2)	MIMAX 39
	GO TO 45	MIMAX 40
40	ROACC=R/ACCUR	MIMAX 41
C		MIMAX 42
45	IF (ROACC-2.000) 50,50,51	MIMAX 43
50	XMIMAX=(A+B)/2.000	MIMAX 44
	YMIMAX=FMIMAX(XMIMAX,NF)	MIMAX 45
	GO TO 170	MIMAX 46
C		MIMAX 47
C		MIMAX 48
C		MIMAX 49
51	IF (ROACC-3.000) 60,60,61	MIMAX 50
60	XL=A+(B-A)/3.000	MIMAX 51
	XR=A+2.000*(B-A)/3.000	MIMAX 52
	YL=FMIMAX(XL,NF)	MIMAX 53
	IF (ITERM.NE.0) GO TO 160	MIMAX 54
	YR=FMIMAX(XR,NF)	MIMAX 55
	IF (ITERM.NE.0) GO TO 150	MIMAX 56
	IF (YL-YR) 70,70,71	MIMAX 57
70	XMIMAX=XR	MIMAX 58
	YMIMAX=YR	MIMAX 59

	GO TO 170	MIMAX 60
	71 XMIMAX=XL	MIMAX 61
	YMIMAX=YL	MIMAX 62
	GO TO 170	MIMAX 63
C		MIMAX 64
C		MIMAX 65
C		MIMAX 66
	61 K=3	MIMAX 67
	E(1) = 1.000	MIMAX 68
	E(2)=1.000	MIMAX 69
C		MIMAX 70
	74 E(K)=E(K-1)&E(K-2)	MIMAX 71
	IF (R(IACC-E(K)) 80,80,75	MIMAX 72
	75 IF (K=40) 77,76,76	MIMAX 73
	76 L=3	MIMAX 74
	GO TO 25	MIMAX 75
	77 K=K&1	MIMAX 76
	GO TO 74	MIMAX 77
	80 N=K	MIMAX 78
	XL=A&E(N-2)*R/E(N)	MIMAX 79
	XR=A&E(N-1)*R/E(N)	MIMAX 80
	YL=FMIMAX(XL,NF)	MIMAX 81
	IF(ITERM.NE.0) GO TO 160	MIMAX 82
	YR=FMIMAX(XR,NF)	MIMAX 83
	IF(ITERM.NE.0) GO TO 150	MIMAX 84
	J=1	MIMAX 85
C		MIMAX 86
	90 NJ=N-J	MIMAX 87
	NJONE=N-J-1	MIMAX 88
	NJTWO=N-J-2	MIMAX 89
	GO TO (100,101), JMIMAX	MIMAX 90
	100 IF (YR-YL) 130,130,110	MIMAX 91
	101 IF (YR-YL) 110,110,130	MIMAX 92
	110 IF (J-N&3) 120,100,160	MIMAX 93
	120 B=XR	MIMAX 94
	A=A	MIMAX 95
	R=B-A	MIMAX 96
	XL=XL	MIMAX 97
	XL=A&E(NJTWO)*R/E(NJ)	MIMAX 98
	YR=YL	MIMAX 99
	YL=FMIMAX(XL,NF)	MIMAX100
	IF(ITERM.NE.0) GO TO 160	MIMAX101
	125 J=J&1	MIMAX102
	GO TO 90	MIMAX103
C		MIMAX104
	130 IF (J-N&3) 140,150,150	MIMAX105
	140 B=B	MIMAX106
	A=XL	MIMAX107
	R=B-A	MIMAX108
	XL=XR	MIMAX109
	XR=A&E(NJONE)*R/E(NJ)	MIMAX110
	YL=YR	MIMAX111
	YR=FMIMAX(XR,NF)	MIMAX112
	IF(ITERM.NE.0) GO TO 150	MIMAX113
	GO TO 125	MIMAX114
C		MIMAX115
	150 XMIMAX=XR	MIMAX116
	YMIMAX=YR	MIMAX117
	GO TO 170	MIMAX118
	160 XMIMAX=XL	MIMAX119

YMIMAX=YL
170 RETURN
END

MIMAX120
MIMAX121
MIMAX122

C	MISC		MISC	1
		SUBROUTINE MISC(N,X,VAL)	MISC	2
		IMPLICIT REAL*8(A-H,O-Z)	MISC	3
		REAL*8 LA1,LA2,LA1F,LAMDA1,LAMDA2,LAMD1F	MISC	4
		COMMON OCCUR, NOCCUR	MISC	5
		COMMON /IGDHL/ IGDL(20), IGDH(20)	MISC	6
		COMMON /IXCOM/ XCOM(200), ICOM(200)	MISC	7
		COMMON /PCCUR/ PCUR(11770)	MISC	8
		DIMENSION ACOE(140)	MISC	9
		DIMENSION DIF(6),NOCCUR(30), OCCUR(4000),RATIO(8),X(20)	MISC	10
		DIMENSION GD(20)	MISC	11
		EQUIVALENCE (PCUR(05881), ACOE)	MISC	12
		EQUIVALENCE (OCCUR(03964), DELRCS)	MISC	13
		EQUIVALENCE (OCCUR(170),W1F),(OCCUR(150),THET1F),(OCCUR(169),RN1F),	MISC	14
		1(OCCUR(147),R1.),(OCCUR(151),LAMD1F),(OCCUR(146),LA1F)	MISC	15
		EQUIVALENCE	MISC	16
		1(OCCUR(133),W1),(OCCUR(134),THETA1),(OCCUR(135),RN1),	MISC	17
		2(OCCUR(136),RB1),(OCCUR(137),LAMD1),(OCCUR(138),LA1),	MISC	18
		3(OCCUR(139),W2),(OCCUR(140),THETA2),(OCCUR(141),RN2),	MISC	19
		4(OCCUR(142),RB2),(OCCUR(143),LAMD2),(OCCUR(144),LA2),	MISC	20
		5(OCCUR(145),ZTURN)	MISC	21
		EQUIVALENCE (OCCUR(42),PI)	MISC	22
		EQUIVALENCE (OCCUR(3921), GD(1))	MISC	23
			MISC	24
C			MISC	25
C		AT PRESENT SUBROUTINE MISC CAN CALL SUBROUTINE POLCAL TO	MISC	26
C		CALCULATE THE FREE SPACE RADAR CROSS SECTION OF A VEHICLE.	MISC	27
C		HOWEVER, THE FIRST THREE ARGUMENTS OF POLCAL HAVE BEEN GENERAL-	MISC	28
C		IZED SO THAT ANY THREE ELEMENTS OF THE OCCUR ARRAY CAN BE USED.	MISC	29
C		IT CALCULATES THE ELEMENTS OF THE GD ARRAY OF GENERALIZED	MISC	30
C		DIFFERENCES BETWEEN UP TO TWENTY PAIRS OF ELEMENTS OF THE	MISC	31
C		OCCUR ARRAY. MISC IS INTENDED, IN GENERAL, TO PERFORM ANY	MISC	32
C		TASKS WHICH DO NOT SEEM TO FIT READILY INTO ANY OF THE OTHER	MISC	33
C		SUBROUTINES.	MISC	34
C			MISC	35
C			MISC	36
		IF (ICOM(7)*ICOM(8)*ICOM(9).NE.0) GO TO 30	MISC	37
		ANS = 0.00	MISC	38
		GO TO 40	MISC	39
C			MISC	40
C			MISC	41
30		ID = ICOM(7)	MISC	42
		DV1 = OCCUR(ID)	MISC	43
		ID = ICOM(8)	MISC	44
		DV2 = OCCUR(ID)	MISC	45
		ID = ICOM(9)	MISC	46
		DV3 = OCCUR(ID)	MISC	47
		CALL POLCAL(DV1,DV2,DV3,ICOM(4),ICOM(5),ICOM(6),2,ACOE,ANS)	MISC	48
40		DELRCS = XCOM(7)-ANS*XCOM(8)	MISC	49
C			MISC	50
C			MISC	51
		DO 111 J = 1, 20	MISC	52
		IF (IGDH(J).EQ.0.OR.IGDL(J).EQ.0) GO TO 112	MISC	53
		GD(J) = OCCUR(IGDH(J)) - OCCUR(IGDL(J))	MISC	54
111		CONTINUE	MISC	55
112		CONTINUE	MISC	56
C			MISC	57
C			MISC	58
C			MISC	59


```

DO 11 I = 1, 8                                MISC 60
IF(I.GT.6) GO TO 11                            MISC 61
DIF(1) = 0.00                                  MISC 62
11  RATIO(1) = 0.00                             MISC 63
    TR = .174532930-1*THETA1                    MISC 64
    R = RN1*DCOS(TR)                             MISC 65
    H = RN1*(1.DO-DSIN(TR))                     MISC 66
    V1=PI*(H*H*(RN1-H/3.DO)&(LA1-H)*(RB1*(RB1&R)&R)/3.DO) MISC 67
    RATIO(7) = W1 / V1                           MISC 68
    IF(2TURN.LE.0.DO) GO TO 200                 MISC 69
    TR = .174532930-1*THETA2                    MISC 70
    R = RN2*DCOS(TR)                             MISC 71
    H = RN2*(1.DO-DSIN(TR))                     MISC 72
    V2=PI*(H*H*(RN2-H/3.DO)&(LA2-H)*(RB2*(RB2&R)&R)/3.DO) MISC 73
    RATIO(8) = W2 / V2                           MISC 74
    DIF(1) = W2 -W1F                             MISC 75
    DIF(2) = THETA2 -THET1F                      MISC 76
    DIF(3) = RN2 -RN1F                            MISC 77
    DIF(4) = RB2 -RB1F                            MISC 78
    DIF(5) = LAMDA2 -LAMDA1F                     MISC 79
    DIF(6) = LA2 -LA1F                            MISC 80
    RATIO(1) = W2/W1F                             MISC 81
    RATIO(2) = THETA2/THET1F                     MISC 82
    IF(RN1F.NE.0.DO) RATIO(3) = RN2/RN1F        MISC 83
    RATIO(4) = RB2/RB1F                           MISC 84
    IF(LAMDA1F.NE.0.DO) RATIO(5) = LAMDA2/LAMDA1F MISC 85
    RATIO(6) = LA2/LA1F                            MISC 86
200 WRITE(6,1000) (RATIO(I),I=1,8), (DIF(I),I=1,6) MISC 87
C                                                MISC 88
    DO 44 I = 1, 8                                MISC 89
    IF(I.GT.6) GO TO 44                          MISC 90
    OCCUR(1&3900) = DIF(1)                       MISC 91
44  OCCUR(1&3906) = RATIO(1)                    MISC 92
C                                                MISC 93
    RETURN                                         MISC 94
C                                                MISC 95
1000 FORMAT(1H0'MISC', W2/W1F, TH2/TH1F, RN2/RN1F, MISC 96
1  RB2/RB1F, LM2/LM1F, LA2/LA1F, W1/V1, W2MISC 97
2/V2'/LH 4X 8F15.3/1H0' W2-W1F, TH2-TH1F, RMISC 98
3RN2-RN1F, RB2-RB1F, LM2-LM1F, LA2-LA1F'/LH 6F15.3) MISC 99
    END                                           MISC 100

```

```

CNEUMAN
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE NEUMAN(C,XR,XI,N,XL,ZZNR,
      1ZZNI,XOR,XOI,XIR,XII,IT)
      DIMENSION ZZNR(200,1),ZZNI(200,1)
C
C
C      SUBROUTINE NEUMAN CALCULATES FOR COMPLEX ARGUMENTS THE
C      NEUMANN FUNCTIONS OF INTEGRAL ORDER ZERO THROUGH TWENTY-FOUR.
C      FOR ARGUMENTS OF MAGNITUDE LESS THAN XL THE STANDARD POWER
C      SERIES IS USED TO CALCULATE THE NEUMANN FUNCTIONS OF ORDER ZERO
C      AND ONE. NEUMAN CALLS IN NEUM0 AND NEUM1 TO CALCULATE THE
C      POWER SERIES FOR THE ZEROth AND FIRST ORDER FUNCTIONS, RESPEC-
C      TIVELY. FOR ARGUMENTS GREATER THAN XL AN ASYMPTOTIC SERIES IS
C      USED FOR ORDERS ZERO AND ONE. NEUMAN CALLS IN NEUMPO AND
C      NEUMQO TO PERFORM THE CALCULATIONS USED IN OBTAINING THE ZEROth
C      ORDER FUNCTIONS AND DIVLT CALCULATES THE FIRST ORDER FUNCTIONS
C      BY ASYMPTOTIC SERIES. A FORMULA IS THEN USED TO GENERATE EACH
C      INCREASING ORDER UNTIL THE NTH ORDER IS DETERMINED.
C
      D=DSQRT(XR*XR+XI*XI)
      IF (D)8,10,8
10    ZZNR(1,IT)=0.000
      ZZNI(1,IT)=0.000
      ZZNR(2,IT)=0.000
      ZZNI(2,IT)=0.000
      GO TO 11
      8 CONTINUE
C
      IF (D-XL)1,1,2
C
      NEUMANN FUNCTION CALCULATIONS FOR LARGE X
2    CALL NEUMPO(XR,XI,C,PNR,PNI)
      CALL NEUMQO(XR,XI,C,QNR,QNI)
      THET=ATANQ(XI,XR)
      PI2RT=DSQRT(2.000/3.141592600)
      XRP=XR-.785398200
      SINRP=DSIN(XRP)
      COSRP=DCOS(XRP)
      E=DEXP(XI)
      COSHH=(E+1.000/E)/2.000
      SINHH=(E-1.000/E)/2.000
      COSR=COSRP*COSHH
      CCSI=-SINRP*SINHH
      SINR=SINRP*COSHH
      SINI=COSRP*SINHH
      OR=XR/D**2
      OI=-XI/D**2
C
      IF (XI)6,5,6
5    XOR=DSQRT(OR)
      XOI=0.000
      GO TO 7
      6 THTO=ATANQ(OI,OR)
      RO=DSQRT(DSORT((OI*OI+OR*OR)))
      XPR=-RO*DCOS(THTO/2.000)
      XPI=-RO*DSIN(THTO/2.000)
      7 ZNIR=CNR+COSR-QNI*CSI+PNR*SINR-PNI*SINI
      ZNII=QNI*COSR+QNR*CSI+PNI*SINR+PNI*SINI

```

NOT REPRODUCIBLE

ZZNR(1,IT)=PI2RT*(XOR*ZNIIR-XOI*ZNI(1,IT))	NEUMA 61
ZZI(1,IT)=PI2RJ*(XOI*ZNIIR+XOR*ZNI(1,IT))	NEUMA 62
XOR=ZZNR(1,IT)*XIR-ZZI(1,IT)*XI-2.00073.141592600*OR	NEUMA 63
XOI=ZZNI(1,IT)*XIR+ZZNR(1,IT)*XI-2.00073.141592600*OI	NEUMA 64
CALL DIVMLT(XOR,XOI,XOR,XOI,ZZNR(2,IT),ZZNI(2,IT))	NEUMA 65
GO TO 3	NEUMA 66
	NEUMA 67
	NEUMA 68
NEUMANN FUNCTION CALCULATIONS FOR SMALL X	NEUMA 69
1 CALL NEUMNO(XR,XI,XOR,XOI,ZZNR(1,IT),ZZNI(1,IT),C)	NEUMA 70
CALL NEUMNI(XR,XI,XIR,XII,ZZNR(2,IT),ZZNI(2,IT),C)	NEUMA 71
	NEUMA 72
3 D=XR**2+XI*XI	NEUMA 73
XNR=XR/D	NEUMA 74
XIB=-XI/D	NEUMA 75
11 IM=N+1	NEUMA 76
IF 4 I=3,NN	NEUMA 77
XNR=DEL/PAT(I-2)	NEUMA 78
IM2=I-2	NEUMA 79
IM=I-1	NEUMA 80
ZZNR(1,IT)=2.000*XNR*(ZZNR(IM,IT)*XRB-ZZI(IM,IT)*XIB)-ZZNR(IM2,	NEUMA 81
1IT)	NEUMA 82
4 ZZI(1,IT)=2.000*XNR*(ZZNR(IM,IT)*XIB+ZZNI(IM,IT)*XRB)-ZZNI(IM2,	NEUMA 83
1IT)	NEUMA 84
RETURN	NEUMA 85
	NEUMA 86



NOT REPRODUCIBLE

CNEUMP	IMPLICIT REAL*8 (A-H,O-Z)	NEUMP	1
	SUBROUTINE NEUMPO(XR,XI,C,PNR,PNI)	NEUMP	2
	DIMENSION SR(2),SI(2)	NEUMP	3
C		NEUMP	4
C	SUBROUTINE NEUMPO CALCULATES THE REAL AND IMAGINARY COMPO-	NEUMP	5
C	NENTS OF THE P-COMPONENT OF THE ZEROth ORDER NEUMANN FUNCTION	NEUMP	6
C	(P,Q).	NEUMP	7
C		NEUMP	8
	LTAPS=6	NEUMP	9
	D=XR**2&XI**2	NEUMP	10
	RR=1.00/D	NEUMP	11
	RI=-2.00*XR*XI/D**2	NEUMP	12
	SR(1)=1.000	NEUMP	13
	SI(1)=0.000	NEUMP	14
	SR(2)=1.065	NEUMP	15
	PNR=0.000	NEUMP	16
	PNI=0.000	NEUMP	17
C		NEUMP	18
	DO 1 I=1,32000	NEUMP	19
	IF (MOD(I,2))3,2,3	NEUMP	20
	2 MM=1	NEUMP	21
	M=2	NEUMP	22
	GO TO 4	NEUMP	23
	3 MM=2	NEUMP	24
	M=1	NEUMP	25
C		NEUMP	26
	4 PNR=PNR&SR(M)	NEUMP	27
	PNI=PNI&SI(M)	NEUMP	28
	IF(DABS(SR(1)-SR(2))-C)5,5,6	NEUMP	29
5	IF(DABS(SI(1)-SI(2))-C)7,7,6	NEUMP	30
6	XH=DFLOAT(2*I)	NEUMP	31
	XT=2.00*XB-1.00	NEUMP	32
	XMULT=-1.00/(XB-1.00)*(XT-2.00)**2/XB*XT*XT/64.00	NEUMP	33
	SR(MM)=XMULT*(SR(M)*RR-SI(M)*RI)	NEUMP	34
	1 SI(MM)=XMULT*(SR(M)*RI&SI(M)*RR)	NEUMP	35
	WRITE (LTAPS,1000)	NEUMP	36
1000	FORMAT (47H SERIES NOT CONVERGED IN 32000 ATTEMPTS FOR P)	NEUMP	37
C		NEUMP	38
	7 CONTINUE	NEUMP	39
	RETURN	NEUMP	40
	END	NEUMP	41
		NEUMP	42

CNEUMQ	IMPLICIT REAL*8 (A-H,O-Z)	NEUMQ 1
	SUBROUTINE NEUMQO(XR, XI, C, QNR, QNI)	NEUMQ 2
	DIMENSION SR(2), SI(2)	NEUMQ 3
C		NEUMQ 4
C	SUBROUTINE NEUMQO CALCULATES THE REAL AND IMAGINARY COMPO-	NEUMQ 5
C	NENTS OF THE Q COMPONENT OF THE ZERO-ORDER NEUMANN FUNCTION	NEUMQ 6
C	(P, Q).	NEUMQ 7
C		NEUMQ 8
	LTAP5=6	NEUMQ 9
	D=XR**2&XI**2	NEUMQ 10
	TNR=XR/D	NEUMQ 11
	TNI=-XI/D	NEUMQ 12
	RR=1.00/D	NEUMQ 13
	KI=-2.00*XR*XI/D**2	NEUMQ 14
	SR(2)=1.065	NEUMQ 15
	QNR=0.000	NEUMQ 16
	QNI=0.000	NEUMQ 17
	SR(1)=-.1250*TNR	NEUMQ 18
	SI(1)=-.1250*TNI	NEUMQ 19
C		NEUMQ 20
	DO 1 I=1, 32000	NEUMQ 21
	IF (MOD(I,2))3,2,3	NEUMQ 22
2	MM=1	NEUMQ 23
	M=2	NEUMQ 24
	GO TO 4	NEUMQ 25
3	MM=2	NEUMQ 26
	M=1	NEUMQ 27
C		NEUMQ 28
4	QNR=QNR&SR(M)	NEUMQ 29
	QNI=QNI&SI(M)	NEUMQ 30
	I=(DAHS(SR(1)-SR(2))-C)5,5,6	NEUMQ 31
5	IF(DAHS(SI(1)-SI(2))-C)7,7,6	NEUMQ 32
6	XB=DFLOAT(2*I)	NEUMQ 33
	XT=2.00*XB&1.00	NEUMQ 34
	XMULT=-1.00/XB*XT**2/(XB&1.00)*(XT-2.00)**2/64.00	NEUMQ 35
	SR(MM)=XMULT*(SR(M)*RK-SI(M)*RI)	NEUMQ 36
1	SI(MM)=XMULT*(SR(M)*RI&SI(M)*RK)	NEUMQ 37
	WRITE (LTAP5,1000)	NEUMQ 38
1000	FORMAT (47H SERIES NOT CONVERGED IN 32000 ATTEMPTS FOR QO)	NEUMQ 39
C		NEUMQ 40
7	CONTINUE	NEUMQ 41
	RETURN	NEUMQ 42
	END	NEUMQ 43
		NEUMQ 44

C	NEUMNO	NEUMNO	1
		IMPLICIT REAL*8 (A-H,O-Z)	NEUMNO	2
		SUBROUTINE NEUMNO(ZR,ZI,XR,XI,ZNR,ZNI,	NEUMNO	3
	1C)	NEUMNO	4
		DIMENSION SR(2),SI(2)	NEUMNO	5
C		NEUMNO	6
C		SUBROUTINE NEUMNO CALCULATES THE REAL AND IMAGINARY COMP-	NEUMNO	7
C		ONENTS OF THE OTH ORDER NEUMANN FUNCTION FROM THE STANDARD POWER	NEUMNO	8
C		SERIES.	NEUMNO	9
C		NEUMNO	10
		LTAP5=6	NEUMNO	11
		DATA PI/3.141592653589793/	NEUMNO	12
		R=DSQRT(ZR**2+ZI**2)/2.00	NEUMNO	13
		RLG=DLG(R)	NEUMNO	14
		THET=ATANOR(ZI,ZR)	NEUMNO	15
		ZNR=(.57721566000+RLG)*XR-THET*XI	NEUMNO	16
		ZNI=(.57721566000+RLG)*XI+THET*XR	NEUMNO	17
		FACT=1.00	NEUMNO	18
		T=1.00	NEUMNO	19
		SR(1)=1.065	NEUMNO	20
		SI(1)=SR(1)	NEUMNO	21
C		NEUMNO	22
		DO 1 I=1,32000	NEUMNO	23
		IF (MOD(I,2))2,3,2	NEUMNO	24
	3	N=1	NEUMNO	25
		GO TO 4	NEUMNO	26
	2	N=2	NEUMNO	27
	4	SUM=(-1.00)**N*T	NEUMNO	28
		XK=I	NEUMNO	29
		XN=XK*THET*2.00	NEUMNO	30
		TEMP=R**XK	NEUMNO	31
		SR(N)=SUM*(TEMP/FACT)**2*DCOS(XN)	NEUMNO	32
		SI(N)=SUM*(TEMP/FACT)**2*DSIN(XN)	NEUMNO	33
		ZNR=ZNR+SR(N)	NEUMNO	34
		ZNI=ZNI+SI(N)	NEUMNO	35
		IF(DABS(SR(2)-SR(1))-C15.6*6	NEUMNO	36
		IF(DABS(SI(2)-SI(1))-C17.6*6	NEUMNO	37
	5	XXI=DFLOAT(I+1)	NEUMNO	38
	6	FACT=FACT*XXI	NEUMNO	39
	1	T=T+1.0/XXI	NEUMNO	40
C		NEUMNO	41
		7 ZNR=ZNR*2.00/PI	NEUMNO	42
		ZNI=ZNI*2.00/PI	NEUMNO	43
		RETURN	NEUMNO	44
		END	NEUMNO	45

NOT REPRODUCIBLE

```

CNEUM1 _____ NEUM1 1
      IMPLICIT REAL*8 (A-H,O-Z) _____ NEUM1 2
      SUBROUTINE NEUMN1(ZR,ZI,XR,XI,ZNR,ZNI, _____ NEUM1 3
1C) _____ NEUM1 4
      DIMENSION SR(2),SI(2) _____ NEUM1 5
C _____ NEUM1 6
C      SUBROUTINE NEUMN1 CALCULATES THE REAL AND IMAGINARY COMPO- _____ NEUM1 7
C      NENTS OF THE 1ST ORDER NEUMANN FUNCTION FROM THE STANDARD POWER _____ NEUM1 8
C      SERIES. _____ NEUM1 9
C _____ NEUM1 10
      DATA PI/3.141592653589793/ _____ NEUM1 11
      R=DSQRT(ZR**2+ZI**2)/2.D0 _____ NEUM1 12
      RLG=DLOG(R) _____ NEUM1 13
      THET=ATANOR(ZI,ZR) _____ NEUM1 14
      ZNR=2.00D*(.577215660D0+RLG)*XR-2.00D*THET*XI _____ NEUM1 15
      ZNI=2.00D*(.577215660D0+RLG)*XI+2.00D*THET*XR _____ NEUM1 16
      X=XR**2+ZI**2 _____ NEUM1 17
      XR=XR/3 _____ NEUM1 18
      XI=XI/(-3) _____ NEUM1 19
      ZNR=XNR-XNI*2.D0 _____ NEUM1 20
      ZNI=XNI-XNR*2.D0 _____ NEUM1 21
      FACT=2.D0 _____ NEUM1 22
      FACT=1.D0 _____ NEUM1 23
      SR(1)=1.065 _____ NEUM1 24
      SI(1)=SR(1) _____ NEUM1 25
      SERS=1.000 _____ NEUM1 26
C _____ NEUM1 27
      DO 1 I=1,32000 _____ NEUM1 28
      IF (MOD(I,2))2,3,2 _____ NEUM1 29
3  N=1 _____ NEUM1 30
      GO TO 4 _____ NEUM1 31
2  N=2 _____ NEUM1 32
4  P1=DFLOAT(I) _____ NEUM1 33
      SR(N)=-(-1.D0)**N/FACT/PI*SERS _____ NEUM1 34
      XI=DFLOAT(2*I-1) _____ NEUM1 35
      XR=XI*THET _____ NEUM1 36
      SR(N)=SUM**R**XK*DCOS(XN)/FACT _____ NEUM1 37
      SI(N)=SUM**R**XK*DSIN(XN)/FACT _____ NEUM1 38
      ZNR=ZNR+SR(N) _____ NEUM1 39
      ZNI=ZNI+SI(N) _____ NEUM1 40
      IF (DABS(SR(2)-SR(1))-C)5,6,6 _____ NEUM1 41
      IF (DABS(SI(2)-SI(1))-C)7,6,6 _____ NEUM1 42
6  FACT=FACT#PI _____ NEUM1 43
1  SERS=SERS+1.D0/PI+1.D0/(PI+1.D0) _____ NEUM1 44
C _____ NEUM1 45
7  ZNR=ZNR/PI _____ NEUM1 46
      ZNI=ZNI/PI _____ NEUM1 47
      RETURN _____ NEUM1 48
      END _____ NEUM1 49

```

NOT REPRODUCIBLE

C	CNOSERL		NUSEB 1
	IMPLICIT REAL*8 (A-H,O-Z)		NUSEB 2
	SUBROUTINE NOSEBL		NUSEB 3
C			NUSEB 4
	REAL*8 LAMDA		NUSEB 5
C			NUSEB 6
	DIMENSION SDOOT(4,8), OCCUR(4000), NOCCUR(30)		NUSEB 7
C			NUSEB 8
	EQUIVALENCE		NUSEB 9
	1(OCCUR(008),CUST),(OCCUR(032),LAMDA),(OCCUR(059),RNDOT),		NUSEB 10
	2(OCCUR(060),RNDOT),(OCCUR(064),SINT),		NUSEB 11
	3(OCCUR(090),XBAR),(OCCUR(091),Z),(OCCUR(092),ZTR),		NUSEB 12
	4(OCCUR(237),XUP),(OCCUR(2740),SDOUT(1,1))		NUSEB 13
C			NUSEB 14
	COMMON OCCUR,NUCCUR		NUSEB 15
C			NUSEB 16
C			NUSEB 17
C			NUSEB 18
C	SUBROUTINE NOSEBL OBTAINS DERIVATIVES OF NOSE RADIUS AND BASE		NUSEB 19
C	RADIUS WITH TIME IN ORDER TO CALCULATE SHAPE CHANGE FOR		NUSEB 20
C	CONSTANT CONE HALF ANGLE IN CONTINUUM FLOW. IN RAREFIED FLOW,		NUSEB 21
C	THE DERIVATIVES ARE SET EQUAL TO ZERO.		NUSEB 22
C			NUSEB 23
C	RNDOT IS CALCULATED FROM STAG. POINT RECESSION RATE.		NUSEB 24
	1-IF(XBAR.GT.XURIGD TO 2-		NUSEB 25
	RNDOT=SDOUT(1,1)*SINT/(1.00-SINT)		NUSEB 26
C			NUSEB 27
C	BLUNTNES RATIO LESS THAN OR = 1.00-3 IS TREATED AS A SHARP		NUSEB 28
C	CONE AND SHARP CONE SIDEWALL RECESSION RATES ARE USED.		NUSEB 29
C	BLUNTNES RATIO GREATER THAN 1.00-3 USES BLUNT CONE SIDEWALL		NUSEB 30
C	RECESSION RATES.		NUSEB 31
	IF(LAMDA.GT.1.00-3)GO TO 3		NUSEB 32
C			NUSEB 33
C	SHARP CONE SIDEWALL RECESSION RATE FACTORS.		NUSEB 34
C	TURBULENT		NUSEB 35
	IF(Z.LT.ZTR)TEM=C.970600*SDOUT(3,8)		NUSEB 36
C	LAMINAR		NUSEB 37
	IF(Z.GE.ZTR)TEM=C.707100*SDOUT(2,8)		NUSEB 38
	GO TO 4		NUSEB 39
C			NUSEB 40
C	BLUNT CONE SIDEWALL RECESSION RATE FACTORS		NUSEB 41
C	TURBULENT		NUSEB 42
	1-IF(Z.LT.ZTR)TEM=SDOUT(3,7)		NUSEB 43
C	LAMINAR		NUSEB 44
	IF(Z.GE.ZTR)TEM=SDOUT(2,7)		NUSEB 45
C			NUSEB 46
C	RATE OF CHANGE OF THE BASE		NUSEB 47
	4 RNDOT=-TEM/CUST		NUSEB 48
	RETURN		NUSEB 49
C			NUSEB 50
C	DERIVATIVES ARE ZERO IN NON-CONTINUUM FLOW		NUSEB 51
	2 RNDOT=0.00		NUSEB 52
	RNDOT=0.00		NUSEB 53
	RETURN		NUSEB 54
	END		NUSEB 55

CPLT	-----	PLT	1
	SUBROUTINE PLT	PLT	2
C	-----	PLT	3
C	SUBROUTINE PLT IS A DUMMY SUBROUTINE INCLUDED SO THAT CALLS	PLT	4
C	TO THE AEROSPACE PLOTTER CAN BE SIMULATED.	PLT	5
	-----	PLT	6
	RETURN	PLT	7
	END	PLT	8

C	POLCAL		POLCA 1
	SUBROUTINE POLCAL(X,Y,Z,MAX,NAY,NAZ,ITP,ACOE,ANS)		POLCA 2
	IMPLICIT REAL*8 (A-H, O-Z)		POLCA 3
	DIMENSION ACOE(140)		POLCA 4
C			POLCA 5
C			POLCA 6
C	SUBROUTINE POLCAL EVALUATES THE POLYNOMIAL DEFINED BY THE		POLCA 7
C	TRIPLE SUMMATION (I FROM 1 TO MAX, J FROM 1 TO NAY, K FROM 1		POLCA 8
C	TO NAZ) OF THE EXPRESSION - (A(MAX*NAY*(K-1)+MAX*(J-1)+1)		POLCA 9
C	*X**(I-1) *Y**(J-1) *Z**(K-1)).		POLCA 10
C			POLCA 11
C			POLCA 12
	M=0		POLCA 13
	ANS=0.000		POLCA 14
	IF(ITP-1) 20,20,10		POLCA 15
	10 NAZ=NAZ		POLCA 16
	GO TO 30		POLCA 17
	20 NAZ=1		POLCA 18
	30 DO 130 K=1,NAZ		POLCA 19
	DO 130 J=1,NAY		POLCA 20
	DO 130 I=1,MAX		POLCA 21
	M=M+1		POLCA 22
	IF(I-1) 40,40,50		POLCA 23
	40 A=1.000		POLCA 24
	GO TO 60		POLCA 25
	50 A=X**(I-1)		POLCA 26
	60 IF(J-1) 70,70,80		POLCA 27
	70 B=1.000		POLCA 28
	GO TO 90		POLCA 29
	80 B=Y**(J-1)		POLCA 30
	90 IF(K-1) 100,100,110		POLCA 31
	100 C=1.000		POLCA 32
	GO TO 120		POLCA 33
	110 C=Z**(K-1)		POLCA 34
C			POLCA 35
	120 ANS=ANS + ACOE(M) * A * B * C		POLCA 36
C			POLCA 37
	130 CONTINUE		POLCA 38
C			POLCA 39
	RETURN		POLCA 40
	END		POLCA 41

C	CPREL I		PREL I 1
		IMPLICIT REAL*8 (A-H,O-Z)	PREL I 2
		SUBROUTINE PRELIM(LP)	PREL I 3
C		REAL*8 LAMDA, LA, MUINF, MINF, ME, M, MX, MY, MZ	PREL I 4
		REAL*8 MINF2, MINF3, MLF, MUW	PREL I 5
C		DIMENSION A(514), TXCGD(50), TABZ(50), TW(4,8), TBMAT(3),	PREL I 6
	1	DOCCUR(4000), NOCCUR(3), THDELZ(25), THTHO(25), THDEL(25)	PREL I 7
		DIMENSION TARRHO(50), TABSND(50), TBATMZ(50)	PREL I 8
C			PREL I 9
		EQUIVALENCE	PREL I 10
		1(OCCUR(001),AREF),(OCCUR(008),COST),(OCCUR(009),CODRAG),	PREL I 11
		2(OCCUR(010),CAPL),(OCCUR(011),COSLAM),(OCCUR(012),CNALP),	PREL I 12
		3(OCCUR(013),CMALP),(OCCUR(017),CPF),(OCCUR(018),CPW),	PREL I 13
		4(OCCUR(020),CMO),(OCCUR(021),O),(OCCUR(027),G)	PREL I 14
		EQUIVALENCE	PREL I 15
		1(OCCUR(028),GAMMA),(OCCUR(029),HSRT),(OCCUR(031),HS),	PREL I 16
		2(OCCUR(032),LAMDA),(OCCUR(033),LA),(OCCUR(034),MUINF),	PREL I 17
		3(OCCUR(035),MINF),(OCCUR(036),ME),(OCCUR(037),M),	PREL I 18
		4(OCCUR(042),PT),(OCCUR(046),PTNPS),(OCCUR(047),PS),	PREL I 19
		5(OCCUR(048),PF),(OCCUR(049),PINF),(OCCUR(051),QD),	PREL I 20
		6(OCCUR(052),RN),(OCCUR(053),RB),(OCCUR(054),RESINT)	PREL I 21
		EQUIVALENCE	PREL I 22
		1(OCCUR(055),RHDINF),(OCCUR(056),RHDINF),(OCCUR(057),R),	PREL I 23
		2(OCCUR(061),RHOE),(OCCUR(062),REYL),(OCCUR(064),SINT),	PREL I 24
		3(OCCUR(066),SOCDST),(OCCUR(067),SINT4),(OCCUR(069),THETA),	PREL I 25
		4(OCCUR(070),TANT),(OCCUR(072),TINF),(OCCUR(076),THETA),	PREL I 26
		5(OCCUR(079),TE),(OCCUR(080),TIMER)	PREL I 27
		EQUIVALENCE	PREL I 28
		1(OCCUR(082),V),(OCCUR(083),VE),(OCCUR(084),W),	PREL I 29
		2(OCCUR(085),W),(OCCUR(088),XCPD),(OCCUR(089),XBARST),	PREL I 30
		3(OCCUR(091),XMAP),(OCCUR(091),Z),(OCCUR(092),ZTR),	PREL I 31
		4(OCCUR(094),CUDLAM),(OCCUR(097),DFLW),(OCCUR(126),XBAR1),	PREL I 32
		5(OCCUR(196),CMTIN),(OCCUR(198),S),(OCCUR(201),TH)	PREL I 33
		EQUIVALENCE	PREL I 34
		1(OCCUR(204),WTH),(OCCUR(205),ZOH),(OCCUR(206),ZOFF),	PREL I 35
		2(OCCUR(207),THO),(OCCUR(208),TCN),(OCCUR(209),TOFF),	PREL I 36
		3(OCCUR(210),MX),(OCCUR(211),MY),(OCCUR(212),MZ),	PREL I 37
		4(OCCUR(213),THINF),(OCCUR(214),AF),(OCCUR(215),CTHZLT),	PREL I 38
		5(OCCUR(216),CPSZET),(OCCUR(217),STHZET),(OCCUR(218),SPSZET),	PREL I 39
		6(OCCUR(219),DELY),(OCCUR(220),DELZ),(OCCUR(225),MWBAR)	PREL I 40
		EQUIVALENCE	PREL I 41
		1(OCCUR(226),DELW),(OCCUR(227),DELW),(OCCUR(229),WTCTAL),	PREL I 42
		2(OCCUR(243),TRZTR),(OCCUR(244),REYINF),(OCCUR(245),PTUTAL),	PREL I 43
		3(OCCUR(247),UPHNDZ),(OCCUR(248),DN1A7Z)	PREL I 44
		EQUIVALENCE	PREL I 45
		1(OCCUR(261), A(1)),(OCCUR(2644),TXCGD(1)),	PREL I 46
		2(OCCUR(2994), TABZ(1)),(OCCUR(2644),TW(1,1)),	PREL I 47
		3(OCCUR(3568), THTHO(1)),(OCCUR(3593),THDELZ(1)),	PREL I 48
		4(OCCUR(3618), THDEL(1)),(OCCUR(3643), TBMAT(1))	PREL I 49
		EQUIVALENCE	PREL I 50
		1(OCCUR(3721), TBATMZ(1)),(OCCUR(3771), TARRHO(1)),	PREL I 51
		2(OCCUR(3821), TABSND(1))	PREL I 52
		EQUIVALENCE	PREL I 53
		1(OCCUR(04),MAXTAN),(OCCUR(07),LOUT),(OCCUR(08),IATFUS),	PREL I 54
		2(OCCUR(09),IKCMO),(OCCUR(14),NPRINT),(OCCUR(22),NTHRSI),	PREL I 55
		3(OCCUR(23),ITHRSI)	PREL I 56
C			PREL I 57
			PREL I 58
			PREL I 59
			PREL I 60

NOT REPRODUCIBLE

C	COMMON OCCUR,NOCCUR	PREL I 60
C		PREL I 61
C		PREL I 62
C		PREL I 63
C	SUBROUTINE PRELIM DOES PRELIMINARY CALCULATIONS OF GEOMETRIC,	PREL I 64
C	FLOW FIELD, AND THRUSTING PARAMETERS.	PREL I 65
C		PREL I 66
C		PREL I 67
C	GEOMETRIC PARAMETERS	PREL I 68
C	CAPL=SHARP CONE SLANT LENGTH, LAMDA=BLUNTNES RATIO, COSLAM=	PREL I 69
E	COSINE OF THE TA*LAMDA, LA=AXIAL LENGTH FROM STAGNATION POINT,	PREL I 70
C	ARFF=AREA OF THE BASE, D=BASE DIAMETER, S=SURFACE DISTANCE	PREL I 71
C	FROM THE NOSE OF BLUNT CONE	PREL I 72
C		PREL I 73
C	TEMPN=32.200*2.270-E	PREL I 74
C	CAPL=RP*RESINT	PREL I 75
C	LAMDA=PN/PB	PREL I 76
C	COSLAM=COST*LAMDA	PREL I 77
C	LA=CAPL*(COST-LAMDA*(1.00-SINT))	PREL I 78
C	ARFF=PI*RB*RB	PREL I 79
C	D=2.00*RB	PREL I 80
C	S=RN*(C.500*PI-THTA)&RB*RESINT-RN/TANT	PREL I 81
C		PREL I 82
C	CALCULATIONS FOR AERO. COEFF. CN ALPHA,CMQ,CM ALPHA	PREL I 83
C	XCGD AND XCPD ARE THE DISTANCES OF CENTER OF GRAVITY AND	PREL I 84
C	CENTER OF PRESSURE FROM THE NOSE NON-DIMENSIONALIZED BY THE	PREL I 85
C	BASE DIAMETER.	PREL I 86
C	CMALP=1.92*SQCSST*(1.00-COSLAM**2)	PREL I 87
C	LL=-1	PREL I 88
C	XCG=D*TABLE(2,TABZ, TXCGD,MAXTAB,LL)	PREL I 89
C	CMALP=-((2.00*(1.00-COSLAM**3)-3.00*SQCSST*COSLAM*(1.00-COSLAM**2))	PREL I 90
C	/((3.00*TANT)	PREL I 91
C	CMALP=CMALP-(RN-XCG)*SQCSST*(1.00-0.5*COSLAM*COSLAM)/RB	PREL I 92
C	CMALP=CMALP/LAMDA*SINT*SQCSST*(1.00-COSLAM*COSLAM)	PREL I 93
C	CMALP=C.56*CMALP	PREL I 94
C	IF(CMALP.GT.C.000)ILP=6	PREL I 95
C	IF(CMALP.GT.C.000)WRITE(6,1000)	PREL I 96
C	1000 FORMAT(1H,1CX,2CHCMALP GREATER THAN 0)	PREL I 97
C	IF(ILP.EQ.6)RETURN	PREL I 98
C	XCPD=-CMALP/CMALP	PREL I 99
C	XCPD=XCPD*TABLE(2,TABZ, TXCGD,MAXTAB,LL)	PREL I 100
C	TEM=COSLAM*SQCSST	PREL I 101
C	IF(IKCMQ.GT.C100)GO TO 43	PREL I 102
C		PREL I 103
C	CALCULATED CMQ BEING USED	PREL I 104
C	CMQ=-TEM*(TEM*(1.00-COSLAM**2)-4.00*(1.00-COSLAM**3)/3.00)/(SINT*PREL I 105	
C	121	PREL I 106
C	CMQ=CMQ-C.500*(1.00-COSLAM**4)/(SINT*SINT)	PREL I 107
C	CMQ=CMQ-C.500*SQCSST*(COSLAM*(RN-XCG)/RB)**2	PREL I 108
C	TEM=XCG-RN*(1.00-SINT)	PREL I 109
C	CMQ=CMQ&2.00*TEM*(2.00*(1.00-COSLAM**3)-3.00*SQCSST*COSLAM*(1.00-	PREL I 110
C	COSLAM*COSLAM))/((3.00*RB*TANT)	PREL I 111
C	CMQ=CMQ-TEM*TEM*SQCSST*(1.00-COSLAM**2)/(RB*RB)	PREL I 112
C	CMQ=C.9600*CMQ	PREL I 113
C	GO TO 44	PREL I 114
C		PREL I 115
C	INPUT CMQ BEING USED	PREL I 116
C	43 CMQ=CMQIN	PREL I 117
C	44 CONTINUE	PREL I 118
C		PREL I 119

C	IF LOPT=4 THE FREE STREAM FLOW PARAMETERS ARE DETERMINED	PRFL 1120
C	FROM THE WIND TUNNEL INPLTS.	PRFL 1121
C	IF (LOPT,50,4) GO TO 79	PRFL 1122
C		PRFL 1123
C	IF (IATMOS,GT,0) GO TO 47	PRFL 1124
C		PRFL 1125
C	USING BUILT IN 1962 STANDARD ATMOSPHERE	PRFL 1126
C	CALL AREFOT(2IZ,ASOUND,PHO)	PRFL 1127
C	CO TO 48	PRFL 1128
C		PRFL 1129
E	USING INPUT ATMOSPHERE FOR SPECIFIED RANGE OF ALTITUDE AND	PRFL 1130
C	THE 1962 STANDARD ATMOSPHERE OUTSIDE THIS RANGE.	PRFL 1131
C	47 CONTINUE	PRFL 1132
C	CALL ENTERPT(TBATM2,TABRH0,TABEND,UPHNDZ,DAHNDZ,Z,PHC,ASOUNI)	PRFL 1133
C	48 CONTINUE	PRFL 1134
C		PRFL 1135
C	CALCULATION OF FREE STREAM CONDITIONS	PRFL 1136
C	REYLA=FREE STREAM REYNOLDS NUMBER BASED ON LA, TINF IS IN	PRFL 1137
C	DEGREES RANKINE, MUINF IS IN LB/FT-SEC, RHOINF IS IN	PRFL 1138
C	SLUGS/FT**3, RHOINI IS IN LB/FT**3, PINF IS IN LB/FT**2.	PRFL 1139
C	ALL LENGTHS ARE IN FEET	PRFL 1140
C	AINF=ASOUNI	PRFL 1141
C	RHOINF=RHO	PRFL 1142
C	RHOINI=RHOINF*G	PRFL 1143
C	CHDRAG=CODLAW/RHOINI	PRFL 1144
C	TINF=AINF*AINF*4.160-4	PRFL 1145
C	MUINF=TEMTEM*TIME**1.500/(TIME 6198.600)	PRFL 1146
C	REYLA=RHOINI*V*LA/MUINF	PRFL 1147
C	MINF=V/AINF	PRFL 1148
C	MINF2=MINF*MINF	PRFL 1149
C	MINF3=MINF2*MINF	PRFL 1150
C	PINF=RHOINI*P*TIME	PRFL 1151
C	TIME=TIME*G	PRFL 1152
C	CO TO 89	PRFL 1153
C		PRFL 1154
C	CALCULATIONS OF FREE STREAM FLOW PROPERTIES FROM WIND TUNNEL	PRFL 1155
C	CONDITIONS INPUT.	PRFL 1156
C	REYINF IS THE REYNOLDS NUMBER PER INCH, REYFT IS THE	PRFL 1157
C	REYNOLDS NUMBER PER FOOT. PTOTAL IS IN LB/FT**2.	PRFL 1158
C	79 CONTINUE	PRFL 1159
C	MINF2=MINF*MINF	PRFL 1160
C	MINF3=MINF2*MINF	PRFL 1161
C	REYFT=REYINF*12.0	PRFL 1162
C	PINF=PTOTAL/(1.000*(GAMMA-1.000)*MINF2/2.000)**(GAMMA/(GAMMA-1.000	PRFL 1163
C	10.0))	PRFL 1164
C	TTK1=PINF*MINF/(2.03560-C2**TEMTEM)	PRFL 1165
C	TTK2=198.600*TTK1	PRFL 1166
C	T2TK1=TTK1*TTK1	PRFL 1167
C	TTFAC=T2TK1*4.000*REYFT*TTK2	PRFL 1168
C	IF (TTFAC,LT,0.000) GO TO 55	PRFL 1169
C	TIME=(TTK1*DSQRT(TTFAC))/2.000*REYFT	PRFL 1170
C	CO TO 57	PRFL 1171
C	50 WRITE(6,1059)	PRFL 1172
C	1059 FORMAT(1X,2B1NEGATIVE SQUARE ROOT IN TIME)	PRFL 1173
C	LP = 6	PRFL 1174
C	RETURN	PRFL 1175
C	57 CONTINUE	PRFL 1176
C	PH-INI=PINF/(R*TIME)	PRFL 1177
C	RHOINF=RHOINI/G	PRFL 1178
C	AINF=DSQRT(TINF)/2.03560-2	PRFL 1179

NOT REPRODUCIBLE

	MU INF=32,200*2,270-2*INF** (.50)/(TINF(198,600)	PREL 1180
	V=M INF*A INF	PREL 1181
C		PREL 1182
	89 IF (LOPT.GE.3) GO TO 16	PREL 1187
	IF (NTHRST.EQ.0) GO TO 16	PREL 1194
C		PREL 1185
C	THRUSTING OPTION CALCULATIONS	PREL 1186
	GO TO (11,12),NTHRST	PREL 1197
C		PREL 1188
C	THRUST IS READ IN AS A FUNCTION OF ALTITUDE	PREL 1199
	11 IF ((Z-ZON).GT.1.D-4) GO TO 16	PREL 1190
	IF (Z.LE.ZOFF) GO TO 13	PREL 1191
	IF ((Z-ZON).GE.0.CDC) KKTHR=0	PREL 1192
	IF (KKTHR.EQ.0.AND.NPRINT.NE.0) WRITE (6,2222) Z	PREL 1193
	2222 FORMAT(1H0,25X,***THRUST TURNED ON AT Z=',F15.6,***'/15X,'THIS MESSAGE IS VALID ONLY WHEN IT IS IMMEDIATELY BEFORE PRINTOUT'/)	PREL 1194
	JJTHR=0	PREL 1196
	KKTHR=KKTHR&1	PREL 1197
	DELTAZ=ZON-Z	PREL 1198
	LL=-1	PREL 1199
	TINF=TH*TABLE(DELTAZ,THDELZ,THTHO,I THRST,LL)	PREL 1200
	GO TO 14	PREL 1201
C		PREL 1202
C	THRUST IS READ IN AS A FUNCTION OF TIME	PREL 1203
	12 IF (TIME-TON).LT.1.D-4) GO TO 16	PREL 1204
	IF (TIME.GE.TOFF) GO TO 13	PREL 1205
	IF ((TON-TIME).GE.0.CDC) KKTHR=0	PREL 1206
	IF (KKTHR.EQ.0.AND.NPRINT.NE.0) WRITE (6,2223) TIME	PREL 1207
	2223 FORMAT(1H0,25X,***THRUST TURNED ON AT TIME=',F12.4,***'/15X,'THIS MESSAGE IS VALID ONLY WHEN IT IS IMMEDIATELY BEFORE PRINTOUT'/)	PREL 1208
	JJTHR=0	PREL 1210
	KKTHR=KKTHR&1	PREL 1211
	DELTA T=TIME-TON	PREL 1212
	LL=-1	PREL 1213
	TINF=TH*TABLE(DELTA T,THDEL T,THTHO,I THRST,LL)	PREL 1214
	14 TH=THINF-AD*INF	PREL 1215
	TMAT(1)=TH*CMIZE 1*CP SIZE T	PREL 1216
	TMAT(2)=TH*CMIZE T*SP SIZE T	PREL 1217
	TMAT(3)=-TH*STHIZE T	PREL 1218
	RK=X CG-LA	PREL 1219
	RK=DELTA TMAT(3)-DELTA TMAT(2)	PREL 1220
	MY=DELTA TMAT(1)-RX*TMAT(3)	PREL 1221
	MZ=RX*TMAT(2)-HELY*TMAT(1)	PREL 1222
	GO TO 10	PREL 1223
C		PREL 1224
C	THRUST PARAMETERS ARE ZEROED OUT WHEN NOT BEING USED	PREL 1225
	13 IF (JJTHR.NE.0) GO TO 16	PREL 1226
	WRITE (6,2224) TIME,Z	PREL 1227
	2224 FORMAT(1H0,25X,***THRUST SHUT OFF AT TIME=',F12.4,' AND Z=',E15.6,PREL 1228	
	1,***'/15X,'THIS MESSAGE IS VALID ONLY WHEN IT IS IMMEDIATELY BEFORE PRINTOUT'/)	PREL 1229
	JJTHR=1	PREL 1230
	16 MX = J.CDC	PREL 1231
	MY=0.CDC	PREL 1232
	MZ=0.CDC	PREL 1233
	TMAT(1)=0.CDC	PREL 1234
	TMAT(2)=0.CDC	PREL 1235
	TMAT(3)=0.CDC	PREL 1236
	TH=0.CDC	PREL 1237
	TINF=0.CDC	PREL 1238
		PREL 1239

NOT REPRODUCIBLE

```

C      KK=K-1
C      CONTINUE
C      CALCULATIONS FOR SHARP CONE EDGE PROPERTIES, STAGNATION
C      ENTHALPY AND PRESSURE
C      PSETC=(C.24DC*TIMEF20.0D-6**V*V)/33.86D0
C      HS=32.96D0**HSP T0
C      PINEFS=(2.0D/(GAMMA61.0D)*MINE2      )**((GAMMA/(GAMMA-1.0D))
C      PINEFS=PINEFS*(2.0D**GAMMA*MINE2      -GAMMA61.0D)/(GAMMA61.0D)**(1
C      X.0D/(
C      (GAMMA-1.0D))
C      PS=PINEF/PINEFS
C      ZP=.50D**HSP T0**V*V
C      RAYL=PIRIN1**V*CAPL/MLINEF
C      SINTM=MINE*SINT
C      VC=V*SQRT(1.0D-1.40D*SINTM**1.9D/(MINE2))
C      PE=PINEF*(1.0D62.8D**SINTM*SINTM*(2.5D)60.0D**SINTM)/(1.0D616.0D*SIN
C      XTM))
C      IF(SINTM.GE.5.7D(10) TO 1
C      TETINF=1.0D66SINTM*10.0966D(60.2267D**SINTM)
C      GO TO 2
C      1 TETINF=C.0D
C      GO 2 K=1.3
C      KK=K-1
C      ISUB1=9005**KK
C      PTEM=(PINEF/2116.7D)**KK
C      GO 3 I=1.5
C      II=I-1
C      ISUB=ISUB16II
C      3 TETINF=TETINF6(I SUB)*(SINTM**II)*PTEM
C      2 TETINF=TINF
C      RHOE=PE/(R*TE)
C      VE=VF**MINE/(V*SQRT(TETINF))
C      MUFE=TETINF**1E**1.4D/(TE&198.6D)
C      REYFS=RHOE*VE*S/MUFE
C      IF(TPZTR.EQ.0.0D)GO TO 46
C      TRANSITION ALTITUDE IS IMPLY
C      ZTR=TPZTR
C      GO TO 45
C      TRANSITION ALTITUDE IS CALCULATED
C      46 CONTINUE
C      ZTR=C.0D
C      GO 8 J=1.2
C      JJ=J-1
C      ISUB1=9006**JJ
C      TEMJ=CAPL**JJ
C      GO 4 I=1.2
C      II=I-1
C      ISUB2=ISUB16II
C      TEM1=TEMJ**II
C      IF(TETAD.GT.15.0D)TEM1=(15.0D)**II
C      GO 4 K=1.2
C      KK=K-1
C      ISUB=ISUB26**KK
C      TEMC=A(I SUB)*TEMJ*TEM1
C      TEML=LAMDA
C      IF(LAMDA.GT.0.3D)TEML=C.3D
C      IF(KK.NE.1)TEM1=TEMC*TEM1**KK

```

```

PREL 1240
PREL 1241
PREL 1242
PREL 1243
PREL 1244
PREL 1245
PREL 1246
PREL 1247
PREL 1248
PREL 1249
PREL 1250
PREL 1251
PREL 1252
PREL 1253
PREL 1254
PREL 1255
PREL 1256
PREL 1257
PREL 1258
PREL 1259
PREL 1260
PREL 1261
PREL 1262
PREL 1263
PREL 1264
PREL 1265
PREL 1266
PREL 1267
PREL 1268
PREL 1269
PREL 1270
PREL 1271
PREL 1272
PREL 1273
PREL 1274
PREL 1275
PREL 1276
PREL 1277
PREL 1278
PREL 1279
PREL 1280
PREL 1281
PREL 1282
PREL 1283
PREL 1284
PREL 1285
PREL 1286
PREL 1287
PREL 1288
PREL 1289
PREL 1290
PREL 1291
PREL 1292
PREL 1293
PREL 1294
PREL 1295
PREL 1296
PREL 1297
PREL 1298
PREL 1299

```

NOT REPRODUCIBLE

REPRODUCED FROM TOZT

```

      8 ZTR=ZTR&TEMO
      45 CONTINUE
C
      IF(Z.GE.ZTR)J=2
      IF(Z.LT.ZTR)J=3
C
      CALCULATION OF CONSTANT PRESSURE SPECIFIC HEAT AT THE EDGE OF
      SHARP CONE BOUNDARY LAYER
      IF(TT.GT.5000.00)CPF=A(111)GA(112)*TE
      IF(TW(J,8).GT.5000.00)CPW=A(111)GA(112)*TW(J,8)
      IF(TT.LT.700.00)CPF=.239800
      IF(TW(J,8).LT.700.00)CPW=.239800
      IF(TT.GT.5000.00)GO TO 4
      IF(TT.LT.700.00)GO TO 4
      CPE=0.00
      DO 5 I=1,6
      II=I-1
      5 CPF=CPE&A(II&105)*TE**II
C
      CALCULATION OF CONSTANT PRESSURE SPECIFIC HEAT AT WALL
      4 IF(TW(J,8).GT.5000.00)GO TO 6
      IF(TW(J,8).LT.700.00)GO TO 6
      CPW=0.00
      DO 7 I=1,6
      II=I-1
      7 CPW=CPW&A(II&105)*TW(J,8)**II
      6 CONTINUE
C
      MWBAR=CPW*TW(J,8)/33.8600
      MUW=TEMTEM*TW(J,8)**1.500/(TW(J,8)&198.600)
C
      CALCULATION OF XBAR, VISCOUS INTERACTION PARAMETER, USED IN
      DETERMINING START OF CONTINUUM FLOW REGIME
      CINF=MUW*TE/(MUINF*TW(J,8))
      XBAR=M*MF*ME*SQRT(CINF/REYS)
C
      CALCULATION OF RAREFACTION PARAMETER USED IN DETERMINING END
      OF FREE MOLECULE AND THE START OF STRONG INTERACTION REGIME
      CINF=MUW*TE/(MUINF*TW(J,8))
      XBARST=M*INF*SQRT(CINF/(REYL*S/CAPL))/ISINT*SINT
      XBAR=XBARST*SINT*SINT
C
      IF(LOPT.EQ.3)GO TO 91
C
      DELW2=WEIGHT LOSS DUE TO ABLATION, DELW3=WEIGHT LOSS DUE TO
      THRUSTING, DELW=TOTAL WEIGHT LOSS, WTOTAL=TOTAL WEIGHT OF
      VEHICLE
      DELW2=WC-W
      DELW3=WC-WTH
      DELW=DELW2&DELW3
      WTOTAL=WC-DELW
      IF(WTOTAL.GT.0.00)GO TO 91
      WRITE(6,1091)WTOTAL
      1091 FORMAT(1H0,10X,'*****WARNING***** WTOTAL =',E11.4,' IS NOT ACCEPTABLE')
      LP=6
C
      91 CONTINUE
      M=WTOTAL/G
      RETURN

```

```

PREL 1300
PREL 1301
PREL 1302
PREL 1303
PREL 1304
PREL 1305
PREL 1306
PREL 1307
PREL 1308
PREL 1309
PREL 1310
PREL 1311
PREL 1312
PREL 1313
PREL 1314
PREL 1315
PREL 1316
PREL 1317
PREL 1318
PREL 1319
PREL 1320
PREL 1321
PREL 1322
PREL 1323
PREL 1324
PREL 1325
PREL 1326
PREL 1327
PREL 1328
PREL 1329
PREL 1330
PREL 1331
PREL 1332
PREL 1333
PREL 1334
PREL 1335
PREL 1336
PREL 1337
PREL 1338
PREL 1339
PREL 1340
PREL 1341
PREL 1342
PREL 1343
PREL 1344
PREL 1345
PREL 1346
PREL 1347
PREL 1348
PREL 1349
PREL 1350
PREL 1351
PREL 1352
PREL 1353
PREL 1354
PREL 1355
PREL 1356
PREL 1357
PREL 1358
PREL 1359

```

NOT REPRODUCIBLE

END

PPFL 1360

CRANDOM	RANUU	1
FUNCTION RANDOM(X)	RANUU	2
IMPLICIT REAL*8 (A-H,O-Z)	RANUU	3
C	RANUU	4
C	RANUU	5
C	SUBROUTINE RANDOM IS A DUMMY SUBROUTINE.	RANUU	6
C	RANUU	7
	RANUU	8
RANDOM = X	RANUU	9
RETURN	RANUU	10
END	RANUU	11

CRCSEC	RCSEC	1
C	RCSEC	2
SUBROUTINE RCSEC(B1,ZNET,WAKEL,SIGP,AOPT)	RCSEC	3
IMPLICIT REAL*8(A-H,O-Z)	RCSEC	4
COMMON OCCUR,NCCUR	RCSEC	5
COMMON /CRCSEC/ F,TAU,SIGMDS,PHI,XBZ,ZME,HH,DDW,C1,C2,C3,C4,C5	RCSEC	6
COMMON /DMCSEC/ BZERO,B2,B3,BTWEN,B24,DX,ZNUS,CNE,OSB,X2BOD,X3B,	RCSEC	7
IND2,NSTWL	RCSEC	8
COMMON /CSIG/ XTT,WS1,XDDP,DWST,SPH2,XLMS,XMSP,XSSP,CB1,ZCR2,ZCR4,	RCSEC	9
IXBZ0,DWBSTL,SPH4,B0,CDDW,WS4,CIGMDS,INDD	RCSEC	10
DIMENSION OCCUR(4000),NCCUR(30)	RCSEC	11
EQUIVALENCE (OCCUR(4000),LP)	RCSEC	12
C	RCSEC	13
C	RCSEC	14
C SUBROUTINE RCSEC COMPUTES THE RADAR CROSS SECTION AND PULSE	RCSEC	15
C SHAPES FROM THE GIVEN OBSERVING RADAR AND THE TRANSITION POINT	RCSEC	16
C ELECTRON DENSITY OF THE VEHICLE AND ALSO COMPUTES WAKE LENGTH.	RCSEC	17
C	RCSEC	18
C	RCSEC	19
C	RCSEC	20
C IND2=0 - NO OUTPUT GENERATED BY THIS ROUTINE.	RCSEC	21
C IND2=1 - INTERMEDIATE STEPS PRINTED OUT	RCSEC	22
C ALTITUDE INDEPENDENT QUANTITIES	RCSEC	23
C SCALING CONSTANTS - BZERO,B2,B3,BTWEN,B24	RCSEC	24
C SCALING CONSTANTS - XB0,B2,B3,XB20,B24	RCSEC	25
C FREQUENCY - F	RCSEC	26
C PULSE LENGTH - TAU	RCSEC	27
C SEA LEVEL COLLISION FREQ. - ZNUS	RCSEC	28
C PRESET CONSTANTS - C1,C2,C3,C4,C5	RCSEC	29
C NOISE LEVEL TO WHICH WAKE LENGTH IS MEASURED - SIGMDS	RCSEC	30
C STEP IN AXIAL COORDINATE - DX	RCSEC	31
C MAXIMUM NUMBER OF STEPS USED TO COMPUTE WAKE LENGTH - NSTWL	RCSEC	32
C ALTITUDE DEPENDENT QUANTITIES	RCSEC	33
C ALTITUDE - HH	RCSEC	34
C MACH NUMBER(UPSTREAM) - ZME	RCSEC	35
C LOOK ANGLE - PHI	RCSEC	36
C WAKE DIAMETER = BASE DIAM. - DDW	RCSEC	37
C SCALE HEIGHT - XBZ	RCSEC	38
C TRANSITION POINT ELECTRON DENSITY - ZNET	RCSEC	39
C 2 BODY OVERDENSE LENGTH - X2BOD	RCSEC	40
C STATION WHERE LINEAR PRODUCTION TERMS FIRST DOMINATE OVER	RCSEC	41
C NON-LINEAR PRODUCTION TERMS - X3B	RCSEC	42
C TRANSITION ELECTRON DENSITY WHEN NONLINEAR PRODUCTION TERMS	RCSEC	43
C ARE CONSIDERED IN TURBULENT WAKE - CNE	RCSEC	44
C ADDITIONAL RCS DUE TO CONSIDERATION ON NONLINEAR PRODUCTION	RCSEC	45
C TERMS IN TURBULENT WAKE - OSB	RCSEC	46
C CB1 = B1	RCSEC	47
C CDDW = DDW	RCSEC	48
C CIGMDS = SIGMDS	RCSEC	49
C XB20 = BTWEN	RCSEC	50
C INDD = IND2	RCSEC	51
200 F2 = F * F	RCSEC	52
WK = 6.28318530869 * F2	RCSEC	53
WC1 = F2/1.009	RCSEC	54
INS2 = ZNUS * ZNUS	RCSEC	55
C	RCSEC	56
C RD = ATMOSPHERIC DENSITY AS A FUNCTION OF ALTITUDE	RCSEC	57
C RD = DEXP(-HH/XBZ)	RCSEC	58
C	RCSEC	59

C	ZNCR - ELECTRON DENSITY AT WHICH THE PLASMA BECOMES OVERDENSE	RCSEC 60
	ZNCR = WC1*(1.000 & RO * RO * ZNS2/F2)**C5)	RCSEC 61
C		RCSEC 62
C	XTT - TRANSITION PT. ONSET OF OVERDENSE REGION	RCSEC 63
	XTT = C1/RO * (1.000 - C2/(ZNE ** C3)) & C4	RCSEC 64
	ZNNCR = ZNET/ZNCR	RCSEC 65
	WS4 = ZNNCR ** XB20	RCSEC 66
	ZCR2 = CNE * CNE/(ZNCR * ZNCR)	RCSEC 67
	IF(PHI)605,604,605	RCSEC 68
604	PHI = 0.0100	RCSEC 69
605	IF(PHI - 90.000)790,790,791	RCSEC 70
791	PHI = 180.000 - PHI	RCSEC 71
C		RCSEC 72
C	CONVERT PHI TO RADIANS AND FIND ITS SINE AND COSINE	RCSEC 73
790	PH = PHI * 1.745329250-2	RCSEC 74
	SPH = DSIN(PH)	RCSEC 75
	CPH = DCOS(PH)	RCSEC 76
	SPH2 = SPH * SPH	RCSEC 77
	SPH4 = SPH2 * SPH2	RCSEC 78
C		RCSEC 79
C	TEST TO PREVENT OVERFLOW (COS 90 = 0)	RCSEC 80
	IF(PHI-90.000)606,607,606	RCSEC 81
607	WS1 = 5000.000	RCSEC 82
	GO TO 608	RCSEC 83
606	WS1 = 150.000 * TAU/CPH	RCSEC 84
608	ZCR2 = ZNNCR * ZNNCR	RCSEC 85
	ZCR4 = ZCR2 * ZCR2	RCSEC 86
	WS7 = B1 * XB20	RCSEC 87
C		RCSEC 88
C	XOD - UNSET OF CONSTANT MULTIPLE SCATTERING REGION	RCSEC 89
	XOD = X2B0D	RCSEC 90
	IF(X2B0D-X38)53,51,51	RCSEC 91
51	XOD = DLOG(ZCR2)/B1	RCSEC 92
53	IF(XOD)1,2,2	RCSEC 93
1	CL0D = 0.000	RCSEC 94
C		RCSEC 95
C	COMPUTE XODP WHICH WILL REPLACE XOD IN ALL EQUATIONS -	RCSEC 96
C	XODP = XOD OR 0.0 IF XOD IS NEGATIVE	RCSEC 97
	XODP = 0.000	RCSEC 98
	GO TO 3	RCSEC 99
	2 CL0D = DMIN(WS1,XOD)	RCSEC100
	XODP = XOD	RCSEC101
C		RCSEC102
C	XSS - PT. OF ONSET OF SINGLE SCATTERING REGION	RCSEC103
	3 XSS = DLOG(ZCR2/SPH4)/B1	RCSEC104
	IF(IN02-1)30000,10000,30000	RCSEC105
10000	WRITE (6,20000) KO,ATT,ZNCR,ZNNCR,XOD	RCSEC106
20000	FORMAT(18X,7H RO=,1PE15.7,6H XT=,1PE15.7,6H NECR=,1PE15.7,10H	RCSEC107
	1 NET/NECR=,1PE15.7,6H XOD=,1PE15.7)	RCSEC108
30000	IF(XSS)4,5,5	RCSEC109
	4 CL5S = 0.000	RCSEC110
C		RCSEC111
C	COMPUTE XSSP WHICH WILL REPLACE XSS IN ALL EQUATIONS -	RCSEC112
C	XSSP = XSS OR 0.0 IF XSS IS NEGATIVE	RCSEC113
	XSSP = 0.000	RCSEC114
	GO TO 4	RCSEC115
	5 CL5S = DMIN(WS1,XSS)	RCSEC116
	XSSP = XSS	RCSEC117
	6 CONTINUE	RCSEC118
	IF(IN02-1)30001,10001,30001	RCSEC119

```

10001 WRITE (6,20001) CLOD,XODP,XSS,CLSS,XSSP
20001 FORMAT(18X,7H CLOD=,1PE15.7,6H XODP=,1PE15.7,6H XSS=,1PE15.7,10H
1 CLSS=,1PE15.7,6H XSSP=,1PE15.7)
30001 DDW2 = DDW * DDW
      DDW2 = DDW * DDW
      DDW4 = DDW2 * DDW2
      XXXX = (ZK * B24 * DDW)**3
C
C DWST - SURFACE SCATTERING CORRECTION TERM FOR BASE DIAMETER
      DWST = (XXXX/(2.000 * SPH2 & XXXX)) * DDW
      DEN = (1.000 & B2 * F2/1.008 * DDW2)**B3
      HO = (B2&K0 * (F2/1.008)**2 * DDW4)/DEN
      DWB1 = DDW * SPH2/B1
      DWBST1 = DWST * SPH2/B1
      IF(XB20)800,801,800
C
C XMS - PT. OF ONSET OF VARIABLE MULTIPLE SCATTERING REGION
801 XMS = XODP
      GU TO 802
800 CONTINUE
      XMS = 2.000 * DLOG(80 * WS4 * SPH**(2.000*(1.000- XB20))*2.000)/
1 WS7
802 IF(XMS-XODP)760,761,761
760 XMS = XODP
      GU TO 762
761 XMS = XMS
762 XLMSD = DMAX1(XMS,XODP)
      IF(IND2-1)30002,10002,30002
10002 WRITE (6,20002) XXXX,DWST,DEN,HO,XMS
20002 FORMAT(18X,7H XXXX=,1PE15.7,6H DWST=,1PE15.7,6H DEN=,1PE15.7,10H
1 HO=,1PE15.7,6H XMS=,1PE15.7)
30002 XLMSD = DMIN1(XLMSD,WS1)
      XLMS = 0.000
      IF(XMSP-XODP)742,742,743
743 XLMS = (XMSP-XODP)/2.000
      XLMS = DMIN1(0.500 * (WS1-XODP),XLMS)
      XLMS = DMAX1(0.000,XLMS)
742 CONTINUE
      WS6 = X3B/WS1
      E1 = DEXP(-B1 * CLOD)
      E2 = DEXP(-B1 * CLSS)
      E3 = DEXP(-B1 * WS1)
      SPH16 = SPH2/16.000
      TRM1 = CLOD * DWST * SPH2 & DWST/B1*ZCR2*((E1-E3)*SPH16
1 & B0 *(E2-E3)) & XLMS * DWST * SPH2
      IF(IND2-1)30003,10003,30003
10003 WRITE (6,20003) XLMSD,XLMS,TRM1,E1,E2
20003 FORMAT(18X,7H XLMSD=,1PE15.7,6H XLMS=,1PE15.7,6H TRM1=,1PE15.7,10H
1 E1=,1PE15.7,6H E2=,1PE15.7)
30003 CONTINUE
      E1 = DEXP(-4.000 * B1 * CLOD) - DEXP(-4.000*WS1 * B1)
C
C THE FOLLOWING TEST IS USED TO PREVENT UNDERFLOW
      IF(DABS(E1) - 1.00-6)9005,9005,9006
9005 E1 = 0.000
9006 CONTINUE
      IF(XB20)780,781,780
780 E2 = DEXP(-0.500 * WS7 * XLMSD) - DEXP(-WS7 * CLSS/2.000)
C
C SIGP - PEAK WAKE RADAR CROSS SECTION

```

```

RCSEC120
RCSEC121
RCSEC122
RCSEC123
RCSEC124
RCSEC125
RCSEC126
RCSEC127
RCSEC128
RCSEC129
RCSEC130
RCSEC131
RCSEC132
RCSEC133
RCSEC134
RCSEC135
RCSEC136
RCSEC137
RCSEC138
RCSEC139
RCSEC140
RCSEC141
RCSEC142
RCSEC143
RCSEC144
RCSEC145
RCSEC146
RCSEC147
RCSEC148
RCSEC149
RCSEC150
RCSEC151
RCSEC152
RCSEC153
RCSEC154
RCSEC155
RCSEC156
RCSEC157
RCSEC158
RCSEC159
RCSEC160
RCSEC161
RCSEC162
RCSEC163
RCSEC164
RCSEC165
RCSEC166
RCSEC167
RCSEC168
RCSEC169
RCSEC170
RCSEC171
RCSEC172
RCSEC173
RCSEC174
RCSEC175
RCSEC176
RCSEC177
RCSEC178
RCSEC179

```

NOT REPRODUCIBLE

	SIGP = TRM1 & DWST * SPH2 * ZCR4*0.2500/B1*E1*ZCR4 & 2.000*B0*SPH2	RCSEC180
	1 ** (2.000 - XB201/WS7 * ZNNCR**XB20 * E2 * DWST	RCSEC181
	GO TO 782	RCSEC182
781	SIGP = TRM1 & 0.2500*DWST*SPH2/B1*ZCR4*E1*ZCR4&B0*SPH4	RCSEC183
	1 * (CLSS - XLMSD) * DDW	RCSEC184
782	CONTINUE	RCSEC185
	IF(IND2-1)30004,10004,30004	RCSEC186
10004	WRITE (6,20004) SIGP	RCSEC187
20004	FORMAT(18X,7H SIGP=,1PE15.7 //)	RCSEC188
	THE FOLLOWING SET OF STATEMENTS COMPUTE THE WAKE LENGTH(WAKEL)	RCSEC189
30004	WAKEL = 0.000	RCSEC190
	NOPT = 0, COMPUTE SIGP ONLY	RCSEC191
	NOPT = 1, COMPUTE SIGP AND WAKEL	RCSEC192
	IF(NOPT)65,900,65	RCSEC193
65	J = 0	RCSEC194
	N1 = NSTWL & 1	RCSEC195
	IT = 1	RCSEC196
	NF = 1	RCSEC197
740	XJ = DFLOAT(J)	RCSEC198
	XX = XJ * DX & XTT	RCSEC199
	SIG = FUN1(XX,NF)	RCSEC200
	IF(J-N1)750,750,751	RCSEC201
751	WRITE (6,20005)	RCSEC202
20005	FORMAT(2X,69HWARNING-NUMBER OF STEPS REQUIRED TO COMPUTE WAKE LENG	RCSEC203
	1TH EXCEEDS NSTWL /)	RCSEC204
	LP = 6	RCSEC205
	RETURN	RCSEC206
750	GO TO (731,732,733), IT	RCSEC207
731	XL1 = XX	RCSEC208
	IT = 2	RCSEC209
734	J = J& 1	RCSEC210
	XN = XX	RCSEC211
	SIGN = SIG	RCSEC212
	GO TO 740	RCSEC213
732	IT = 3	RCSEC214
	GO TO 734	RCSEC215
733	IF(SIG-SIGN)736,736,735	RCSEC216
735	XL1 = XN	RCSEC217
	GO TO 734	RCSEC218
736	XH1 = XX	RCSEC219
	CALL FIB1(XL1,XH1,1,1,1.0D-4,20,XPEAK,SPEAK)	RCSEC220
	IF(SPEAK-SIGMDS)737,737,738	RCSEC221
737	WAKEL = 0.000	RCSEC222
	RETURN	RCSEC223
		RCSEC224
		RCSEC225
		RCSEC226
		RCSEC227
		RCSEC228
		RCSEC229
		RCSEC230
		RCSEC231
		RCSEC232
		RCSEC233
		RCSEC234
		RCSEC235
		RCSEC236
		RCSEC237
		RCSEC238
		RCSEC239

C	738	XN = XPEAK	RCSEC240
		J = 1	RCSEC241
	741	XJ = DFLUAT(J)	RCSEC242
		XX = XPEAK & DX * XJ	RCSEC243
		SIG = FUN1(XX,NF)	RCSEC244
		IF(SIG - SIGMDS)739,739,7422	RCSEC245
	7422	J = J&1	RCSEC246
		IF(J-N1)752,752,751	RCSEC247
	752	XN = XX	RCSEC248
		GO TO 741	RCSEC249
C			RCSEC250
	739	CALL FIB1(XN,XX,2,-1,1.00-4,20,XNL,DUM)	RCSEC251
C			RCSEC252
		WAKEL = XNL - XPEAK	RCSEC253
C			RCSEC254
	900	RETURN	RCSEC255
		END	RCSEC256
			RCSEC257

C	CREADIT	READJ 1
	IMPLICIT REAL*8 (A-H,O-Z)	READJ 2
	SUBROUTINE READIT(ILL111,IMPL0T)	READJ 3
C		READJ 4
C		READJ 5
C		READJ 6
C	SUBROUTINE READIT READS IN THE INPUT FOR ONE CASE AND TESTS	READJ 7
C	CERTAIN VALUES IN THE IUP ARRAY.	READJ 8
C		READJ 9
C		READJ 10
	REAL*8 LAMDA1,LAMDA2,LA1,LA2	READJ 11
	REAL*8 ISP,MW,NSL1,NSL2,NST1,NST2,NGL1,NGL2,NGT1,NGT2	READJ 12
C		READJ 13
	COMMON OCCUR,NOCCUR	READJ 14
	COMMON /CICCUR/ IOCCUR(320)	READJ 15
	DIMENSION IUP(90)	READJ 16
	DIMENSION ALPTAB(75),OCCUR(4000),NOCCUR(30),NPLOT(5),	READJ 17
1	CHIGH(16),CDOWN(16),THTHO(25),THDELZ(25),THDELT(25),	READJ 18
2	FXCGO1(50),FXCGO2(50),TAB11(50),TAB12(50),TABIX1(50),	READJ 19
3	TABZ1(50),TABZ2(50),MTAB(75),WHTAB(75),COTAB(75),	READJ 20
4	TABIX2(50),WCDTAB(75)	READJ 21
	DIMENSION WTZ(75),WTMINF(75),WTKINF(75),WTPTOT(75),TRAJT(75),	READJ 22
1	TRAJZ(75),TRAJVI(75),TRAJW(75),TRAJRN(75),TRJALP(75)	READJ 23
	DIMENSION TABRHO(50),TABSND(50),TBATHZ(50)	READJ 24
	DIMENSION A(514),B(21)	READJ 25
C		READJ 26
	EQUIVALENCE (OCCUR(00301), A)	READJ 27
	EQUIVALENCE (OCCUR(00214), AE)	READJ 28
	EQUIVALENCE (OCCUR(03046), ALPTAB)	READJ 29
	EQUIVALENCE (OCCUR(00122), ALST)	READJ 30
	EQUIVALENCE (OCCUR(00188), AWREF)	READJ 31
	EQUIVALENCE (OCCUR(00823), B)	READJ 32
	EQUIVALENCE (OCCUR(00152), BETA11)	READJ 33
	EQUIVALENCE (OCCUR(00171), BETA12)	READJ 34
	EQUIVALENCE (OCCUR(00153), BETA21)	READJ 35
	EQUIVALENCE (OCCUR(00172), BETA22)	READJ 36
	EQUIVALENCE (OCCUR(00154), BETA31)	READJ 37
	EQUIVALENCE (OCCUR(00173), BETA32)	READJ 38
	EQUIVALENCE (OCCUR(00155), BETA41)	READJ 39
	EQUIVALENCE (OCCUR(00174), BETA42)	READJ 40
	EQUIVALENCE (OCCUR(00115), C)	READJ 41
	EQUIVALENCE (OCCUR(00019), CAPG)	READJ 42
	EQUIVALENCE (OCCUR(00128), CASE)	READJ 43
	EQUIVALENCE (OCCUR(03549), CDOWN)	READJ 44
	EQUIVALENCE (OCCUR(03363), COTAB)	READJ 45
	EQUIVALENCE (OCCUR(03533), CHIGH)	READJ 46
	EQUIVALENCE (OCCUR(00124), CHQIN1)	READJ 47
	EQUIVALENCE (OCCUR(00125), CHQIN2)	READJ 48
	EQUIVALENCE (OCCUR(00161), CPG1)	READJ 49
	EQUIVALENCE (OCCUR(00180), CPG2)	READJ 50
	EQUIVALENCE (OCCUR(00160), CP21)	READJ 51
	EQUIVALENCE (OCCUR(00179), CP22)	READJ 52
	EQUIVALENCE (OCCUR(00127), DATE)	READJ 53
	EQUIVALENCE (OCCUR(00166), DELHC1)	READJ 54
	EQUIVALENCE (OCCUR(00185), DELHC2)	READJ 55
	EQUIVALENCE (OCCUR(00187), DELIN)	READJ 56
	EQUIVALENCE (OCCUR(00159), DELRH1)	READJ 57
	EQUIVALENCE (OCCUR(00178), DELRH2)	READJ 58
	EQUIVALENCE (OCCUR(00219), DELY)	READJ 59

EQUIVALENCE (OCCUR(00220), DELZ)	HEAD1 60
EQUIVALENCE (OCCUR(00248), UNBNDZ)	HEAD1 61
EQUIVALENCE (OCCUR(00129), EMO)	HEAD1 62
EQUIVALENCE (OCCUR(167), EPSIL1)	HEAD1 63
EQUIVALENCE (OCCUR(00186), EPSIL2)	HEAD1 64
EQUIVALENCE (OCCUR(00157), F1)	HEAD1 65
EQUIVALENCE (OCCUR(00176), F2)	HEAD1 66
EQUIVALENCE (OCCUR(00027), G)	HEAD1 67
EQUIVALENCE (OCCUR(00105), GAMFO)	HEAD1 68
EQUIVALENCE (OCCUR(00028), GAMMA)	HEAD1 69
EQUIVALENCE (OCCUR(00156), HREF1)	HEAD1 70
EQUIVALENCE (OCCUR(00175), HREF2)	HEAD1 71
EQUIVALENCE (OCCUR(03233), HTAB)	HEAD1 72
EQUIVALENCE (OCCUR(00222), ISP)	HEAD1 73
EQUIVALENCE (OCCUR(00137), LAMDA1)	HEAD1 74
EQUIVALENCE (OCCUR(00143), LAMDA2)	HEAD1 75
EQUIVALENCE (OCCUR(00138), LA1)	HEAD1 76
EQUIVALENCE (OCCUR(00144), LA2)	HEAD1 77
EQUIVALENCE (OCCUR(00117), MW)	HEAD1 78
EQUIVALENCE (OCCUR(00164), NGL1)	HEAD1 79
EQUIVALENCE (OCCUR(00183), NGL2)	HEAD1 80
EQUIVALENCE (OCCUR(00165), NGT1)	HEAD1 81
EQUIVALENCE (OCCUR(00184), NGT2)	HEAD1 82
EQUIVALENCE (OCCUR(00162), NSL1)	HEAD1 83
EQUIVALENCE (OCCUR(00181), NSL2)	HEAD1 84
EQUIVALENCE (OCCUR(00163), NST1)	HEAD1 85
EQUIVALENCE (OCCUR(00182), NST2)	HEAD1 86
EQUIVALENCE (OCCUR(00112), PH10)	HEAD1 87
EQUIVALENCE (OCCUR(00223), PSIZET)	HEAD1 88
EQUIVALENCE (OCCUR(00114), PSID)	HEAD1 89
EQUIVALENCE (OCCUR(00109), PO)	HEAD1 90
EQUIVALENCE (OCCUR(00110), QO)	HEAD1 91
EQUIVALENCE (OCCUR(00057), K)	HEAD1 92
EQUIVALENCE (OCCUR(00156), KB1)	HEAD1 93
EQUIVALENCE (OCCUR(00142), KB2)	HEAD1 94
EQUIVALENCE (OCCUR(00063), KE)	HEAD1 95
EQUIVALENCE (OCCUR(00158), KHDZ1)	HEAD1 96
EQUIVALENCE (OCCUR(00177), KHDZ2)	HEAD1 97
EQUIVALENCE (OCCUR(00135), KN1)	HEAD1 98
EQUIVALENCE (OCCUR(00141), KN2)	HEAD1 99
EQUIVALENCE (OCCUR(00116), SIG)	HEAD1 100
EQUIVALENCE (OCCUR(00111), SMRO)	HEAD1 101
EQUIVALENCE (OCCUR(03033), TABIX1)	HEAD1 102
EQUIVALENCE (OCCUR(03083), TABIX2)	HEAD1 103
EQUIVALENCE (OCCUR(02933), TABI1)	HEAD1 104
EQUIVALENCE (OCCUR(02983), TABI2)	HEAD1 105
EQUIVALENCE (OCCUR(03771), TABRHO)	HEAD1 106
EQUIVALENCE (OCCUR(03821), TABSND)	HEAD1 107
EQUIVALENCE (OCCUR(03133), TABZ1)	HEAD1 108
EQUIVALENCE (OCCUR(03183), TABZ2)	HEAD1 109
EQUIVALENCE (OCCUR(03721), TBATHZ)	HEAD1 110
EQUIVALENCE (OCCUR(00077), TCRIT)	HEAD1 111
EQUIVALENCE (OCCUR(00078), TECUN)	HEAD1 112
EQUIVALENCE (OCCUR(03618), THDEL1)	HEAD1 113
EQUIVALENCE (OCCUR(03593), THDEL2)	HEAD1 114
EQUIVALENCE (OCCUR(00113), THEALO)	HEAD1 115
EQUIVALENCE (OCCUR(00134), THETA1)	HEAD1 116
EQUIVALENCE (OCCUR(00140), THETA2)	HEAD1 117
EQUIVALENCE (OCCUR(00224), THEZET)	HEAD1 118
EQUIVALENCE (OCCUR(03568), THHNO)	HEAD1 119

EQUIVALENCE (OCCUR(00207), THO)	READ1120
EQUIVALENCE (OCCUR(00132), TINIT)	READ1121
EQUIVALENCE (OCCUR(00209), TOFF)	READ1122
EQUIVALENCE (OCCUR(00208), TUN)	READ1123
EQUIVALENCE (OCCUR(01644), TRAJRN)	READ1124
EQUIVALENCE (OCCUR(01344), TRAJT)	READ1125
EQUIVALENCE (OCCUR(01454), TRAJV)	READ1126
EQUIVALENCE (OCCUR(01569), TRAJW)	READ1127
EQUIVALENCE (OCCUR(01419), TRAJZ)	READ1128
EQUIVALENCE (OCCUR(01719), TRJALP)	READ1129
EQUIVALENCE (OCCUR(00243), TRZTR)	READ1130
EQUIVALENCE (OCCUR(00123), TST)	READ1131
EQUIVALENCE (OCCUR(00148), TWST)	READ1132
EQUIVALENCE (OCCUR(00149), TW1)	READ1133
EQUIVALENCE (OCCUR(00168), TW2)	READ1134
EQUIVALENCE (OCCUR(02833), TXCGD1)	READ1135
EQUIVALENCE (OCCUR(02883), TXCGD2)	READ1136
EQUIVALENCE (OCCUR(00102), TO)	READ1137
EQUIVALENCE (OCCUR(00247), UPBMDZ)	READ1138
EQUIVALENCE (OCCUR(00106), VO)	READ1139
EQUIVALENCE (OCCUR(03458), WCDTAB)	READ1140
EQUIVALENCE (OCCUR(03308), WHTAB)	READ1141
EQUIVALENCE (OCCUR(01119), WTMINF)	READ1142
EQUIVALENCE (OCCUR(01269), WPTUT)	READ1143
EQUIVALENCE (OCCUR(01194), WTRINF)	READ1144
EQUIVALENCE (OCCUR(01044), WTZ)	READ1145
EQUIVALENCE (OCCUR(00133), W1)	READ1146
EQUIVALENCE (OCCUR(00139), W2)	READ1147
EQUIVALENCE (OCCUR(00238), XLOW)	READ1148
EQUIVALENCE (OCCUR(00107), XRO)	READ1149
EQUIVALENCE (OCCUR(00237), XUP)	READ1150
EQUIVALENCE (OCCUR(00240), XLOW1)	READ1151
EQUIVALENCE (OCCUR(00239), X1UP)	READ1152
EQUIVALENCE (OCCUR(00120), ZBAR)	READ1153
EQUIVALENCE (OCCUR(00093), ZETA)	READ1154
EQUIVALENCE (OCCUR(00206), ZUFF)	READ1155
EQUIVALENCE (OCCUR(00205), ZON)	READ1156
EQUIVALENCE (OCCUR(00118), ZPR1)	READ1157
EQUIVALENCE (OCCUR(00119), ZPA2)	READ1158
EQUIVALENCE (OCCUR(00121), ZST)	READ1159
EQUIVALENCE (OCCUR(00145), ZTURN)	READ1160
EQUIVALENCE (OCCUR(00109), ZO)	READ1161
EQUIVALENCE (NOCCUR(08), IATMOS)	READ1162
EQUIVALENCE (NOCCUR(09), IKCMU)	READ1163
EQUIVALENCE (NOCCUR(30), INALPH)	READ1164
EQUIVALENCE (NOCCUR(29), ITAPE)	READ1165
EQUIVALENCE (NOCCUR(23), ITHRST)	READ1166
EQUIVALENCE (NOCCUR(07), LJPT)	READ1167
EQUIVALENCE (NOCCUR(20), MATLN1)	READ1168
EQUIVALENCE (NOCCUR(21), MATLN2)	READ1169
EQUIVALENCE (NOCCUR(06), MAXVAL)	READ1170
EQUIVALENCE (NOCCUR(10), MHEAT)	READ1171
EQUIVALENCE (NOCCUR(16), MXTAB1)	READ1172
EQUIVALENCE (NOCCUR(17), MXTAB2)	READ1173
EQUIVALENCE (NOCCUR(18), MAXCD)	READ1174
EQUIVALENCE (NOCCUR(19), MAXWCD)	READ1175
EQUIVALENCE (NOCCUR(15), NGEOM)	READ1176
EQUIVALENCE (NOCCUR(03), MUPT)	READ1177
EQUIVALENCE (NOCCUR(005), NUSEDP)	READ1178
EQUIVALENCE (NOCCUR(24), NPLDT)	READ1179

	EQUIVALENCE (NOCCUR(14), NPRINT)	READ1180
	EQUIVALENCE (NOCCUR(22), NTHKST)	READ1181
	EQUIVALENCE (IOCCUR(00001), IOP)	READ1182
C		READ1183
C		READ1184
	DATA INCDL1/0388/	READ1185
C		READ1186
C		READ1187
	GO TO (1,2),ILL111	READ1188
L	CONTINUE	READ1189
	CALL ZREADX	READ1190
	CALL SETUP(8HOCUR ,8,OCUR,4000)	READ1191
C		READ1192
C	CASE IDENTIFICATION INPUTS	READ1193
	CALL SETUP(8HCASE ,8,CASE)	READ1194
	CALL SETUP(8HDATE ,8,DATE)	READ1195
	CALL SETUP(8HMEMU ,8,EMO)	READ1196
C		READ1197
C	PRINTING AND STOP CONTROLS	READ1198
	CALL SETUP(8HNPRINT ,4,NPRINT)	READ1199
	CALL SETUP(8HZPKA ,8,ZPKA)	READ1200
	CALL SETUP(8HZBAK ,8,ZBAK)	READ1201
	CALL SETUP(8HZPK2 ,8,ZPK2)	READ1202
	CALL SETUP(8HZST ,8,ZST)	READ1203
	CALL SETUP(8HTST ,8,TST)	READ1204
	CALL SETUP(8HALST ,8,ALST)	READ1205
C		READ1206
C	INTEGRATION ACCURACY CONTROLS	READ1207
	CALL SETUP(8HDELIN ,8,DELIN)	READ1208
	CALL SETUP(8HCHIGH ,8,CHIGH ,16)	READ1209
	CALL SETUP(8HCDOWN ,8,CDOWN ,16)	READ1210
C		READ1211
C	INPUT TRANSITION ALTITUDE TRZTR	READ1212
	CALL SETUP(8HTKZTR ,8,TKZTR)	READ1213
C		READ1214
C	INPUT LIMITS ON THE FAIRING REGIONS	READ1215
	CALL SETUP(8HXIUP ,8,XIUP)	READ1216
	CALL SETUP(8HXLOW ,8,XLOW)	READ1217
	CALL SETUP(8HXUP ,8,XUP)	READ1218
	CALL SETUP(8HXLOW ,8,XLOW)	READ1219
C		READ1220
C	TRAJECTORY OPTION AND INITIAL VALUES OF TRAJECTORY PARAMETERS	READ1221
	CALL SETUP(8HLOPT ,4,LOPT)	READ1222
	CALL SETUP(8HZO ,8,ZO)	READ1223
	CALL SETUP(8HGAMFO ,8,GAMFO)	READ1224
	CALL SETUP(8HVO ,8,VO)	READ1225
	CALL SETUP(8HTO ,8,TO)	READ1226
	CALL SETUP(8HXRO ,8,XRO)	READ1227
	CALL SETUP(8HPO ,8,PO)	READ1228
	CALL SETUP(8HQO ,8,QO)	READ1229
	CALL SETUP(8HSMKO ,8,SMKO)	READ1230
	CALL SETUP(8HPHIO ,8,PHIO)	READ1231
	CALL SETUP(8HPSIO ,8,PSIO)	READ1232
	CALL SETUP(8HTHEALO ,8,THEALO)	READ1233
C		READ1234
C	INPUT TEST QUANTITIES FOR ANGLE OF ATTACK CYCLE TIME	READ1235
	CALL SETUP(8HTCRIT ,8,TCRIT)	READ1236
	CALL SETUP(8HTECON ,8,TECON)	READ1237
C		READ1238
		READ1239

C	NUMBER OF TABULAR INPUTS FOR EITHER THE INPUT WIND TUNNEL	READ1240
C	OPTION OR THE INPUT-TRAJECTORY-OPTION.	READ1241
	CALL SETUP(BHMAXVAL ,4,MAXVAL)	READ1242
C		READ1243
C	INPUTS FOR WIND TUNNEL OPTION	READ1244
	CALL SETUP(BHWTZ ,8,WTZ,75)	READ1245
	CALL SETUP(BHWTMINF ,8,WTMINF,75)	READ1246
	CALL SETUP(BHWTRINF ,8,WTRINF,75)	READ1247
	CALL SETUP(BHWTPTOT ,8,WTPTOT,75)	READ1248
C		READ1249
C	INPUTS FOR TABULAR INPUT TRAJECTORY	READ1250
	CALL SETUP(BHTRAJT ,8,TRAJT,75)	READ1251
	CALL SETUP(BHTRAJZ ,8,TRAJZ,75)	READ1252
	CALL SETUP(BHTRAJV ,8,TRAJV,75)	READ1253
	CALL SETUP(BHTRAJW ,8,TRAJW,75)	READ1254
	CALL SETUP(BHTRAJRN ,8,TRAJRN,75)	READ1255
	CALL SETUP(BHTRJALP ,8,TRJALP,75)	READ1256
C		READ1257
C	TABULAR INPUT ANGLE OF ATTACK FOR USE WITH LOPT=1 TRAJECTORY	READ1258
	CALL SETUP(BHINALPH ,4,INALPH)	READ1259
	CALL SETUP(BHALPTAB ,8,ALPTAB,75)	READ1260
C		READ1261
C	MASS LOSS OPTION INPUTS	READ1262
	CALL SETUP(BHMOPT ,4,MOPT)	READ1263
	CALL SETUP(BHMHEAT ,4,MHEAT)	READ1264
	CALL SETUP(BHNOSEOP ,4,NOSEOP)	READ1265
C		READ1266
C	TWST IS A WALL TEMPERATURE USED IN RAREFIED FLOW CALCULATIONS	READ1267
	CALL SETUP(BHTWST ,8,TWST)	READ1268
C		READ1269
C	TINIT IS INTERNAL TEMPERATURE OF VEHICLE USED TO OBTAIN	READ1270
C	TEMPERATURE GRADIENT IN MASS LOSS CALCULATIONS USING THE	READ1271
C	ITERATIVE METHOD.	READ1272
	CALL SETUP(BHTINIT ,8,TINIT)	READ1273
C		READ1274
C	OPTION INDICATING THE SET OF GEOMETRIC VARIABLES BEING INPUT	READ1275
	CALL SETUP(BHNGEOM ,4,NGEOM)	READ1276
C	OPTION TO INPUT CMQ	READ1277
	CALL SETUP(BHIKCMQ ,4,IKCMQ)	READ1278
C	GEOMETRY INPUTS FOR THE FIRST CONFIGURATION	READ1279
	CALL SETUP(BHW1 ,8,W1)	READ1280
	CALL SETUP(BHTHETA1 ,8,THETA1)	READ1281
	CALL SETUP(BHKN1 ,8,RN1)	READ1282
	CALL SETUP(BHRB1 ,8,RB1)	READ1283
	CALL SETUP(BHLAMD1 ,8,LAMD1)	READ1284
	CALL SETUP(BHLA1 ,8,LA1)	READ1285
	CALL SETUP(BHMXTAB1 ,4,MXTAB1)	READ1286
	CALL SETUP(BHCMQIN1 ,8,CMQIN1)	READ1287
	CALL SETUP(BHTABZ1 ,8,TABZ1 ,50)	READ1288
	CALL SETUP(BHTXCGD1 ,8,IXCGD1,50)	READ1289
	CALL SETUP(BHTAB11 ,8,TAB11 ,50)	READ1290
	CALL SETUP(BHTAB1X1 ,8,TAB1X1,50)	READ1291
C		READ1292
C	FIRST CONFIGURATION HEATSHIELD PROPERTIES	READ1293
	CALL SETUP(BHMATL1 ,4,MATL1)	READ1294
	CALL SETUP(BHTW1 ,8,TW1)	READ1295
	CALL SETUP(BHBETA11 ,8,BETA11)	READ1296
	CALL SETUP(BHBETA21 ,8,BETA21)	READ1297
	CALL SETUP(BHBETA31 ,8,BETA31)	READ1298
	CALL SETUP(BHBETA41 ,8,BETA41)	READ1299

CALL SETUP(BHREF1	,8,HREF1)	READ1300
CALL SETUP(BHF1	,8,F1)	READ1301
CALL SETUP(BHRHU21	,8,RHU21)	READ1302
CALL SETUP(BHDELRH1	,8,DELRH1)	READ1303
CALL SETUP(BHCP21	,8,CP21)	READ1304
CALL SETUP(BHCPG1	,8,CPG1)	READ1305
CALL SETUP(BHNSL1	,8,NSL1)	READ1306
CALL SETUP(BHNST1	,8,NST1)	READ1307
CALL SETUP(BHNGL1	,8,NGL1)	READ1308
CALL SETUP(BHNGT1	,8,NGT1)	READ1309
CALL SETUP(BHDELHC1	,8,DELHC1)	READ1310
CALL SETUP(BHEPSIL1	,8,EPSIL1)	READ1311
C		READ1312
C	ZTURN IS ALTITUDE FOR DISCONTINUOUS CONFIGURATION CHANGE	READ1313
CALL SETUP(BHZTURN	,8,ZTURN)	READ1314
C		READ1315
C	GEDMETRY INPUTS FOR THE SECOND CONFIGURATION	READ1316
CALL SETUP(BHW2	,8,W2)	READ1317
CALL SETUP(BHTHETA2	,8,THETA2)	READ1318
CALL SETUP(BHKN2	,8,KN2)	READ1319
CALL SETUP(BHRB2	,8,RB2)	READ1320
CALL SETUP(BHLAMDA2	,8,LAMDA2)	READ1321
CALL SETUP(BHLA2	,8,LA2)	READ1322
CALL SETUP(BHMXTAB2	,4,MXTAB2)	READ1323
CALL SETUP(BHCMQIN2	,8,CMQIN2)	READ1324
CALL SETUP(BHTAB22	,8,TAB22 ,50)	READ1325
CALL SETUP(BHTXCGD2	,8,TXCGD2 ,50)	READ1326
CALL SETUP(BHTAB12	,8,TAB12 ,50)	READ1327
CALL SETUP(BHTABIX2	,8,TABIX2 ,50)	READ1328
C		READ1329
C	SECOND CONFIGURATION HEATSHIELD PROPERTIES	READ1330
CALL SETUP(BHMATLN2	,4,MATLN2)	READ1331
CALL SETUP(BHTW2	,8,TW2)	READ1332
CALL SETUP(BHJLTA12	,8,BETA12)	READ1333
CALL SETUP(BHBETA22	,8,BETA22)	READ1334
CALL SETUP(BHBETA32	,8,BETA32)	READ1335
CALL SETUP(BHBETA42	,8,BETA42)	READ1336
CALL SETUP(BHREF2	,8,HREF2)	READ1337
CALL SETUP(BHF2	,8,F2)	READ1338
CALL SETUP(BHRHU22	,8,RHU22)	READ1339
CALL SETUP(BHDELRH2	,8,DELRH2)	READ1340
CALL SETUP(BHCP22	,8,CP22)	READ1341
CALL SETUP(BHCPG2	,8,CPG2)	READ1342
CALL SETUP(BHNSL2	,8,NSL2)	READ1343
CALL SETUP(BHNST2	,8,NST2)	READ1344
CALL SETUP(BHNGL2	,8,NGL2)	READ1345
CALL SETUP(BHNGT2	,8,NGT2)	READ1346
CALL SETUP(BHDELHC2	,8,DELHC2)	READ1347
CALL SETUP(BHEPSIL2	,8,EPSIL2)	READ1348
C		READ1349
C	DRAG COEFFICIENT INPUTS	READ1350
CALL SETUP(BHMAXCD	,4,MAXCD)	READ1351
CALL SETUP(BHHTAB	,8,HTAB ,75)	READ1352
CALL SETUP(BHCUTAB	,8,CUTAB ,75)	READ1353
C		READ1354
C	INCREMENT IN DRAG COEFF. INPUTS	READ1355
CALL SETUP(BHMAXWCD	,4,MAXWCD)	READ1356
CALL SETUP(BHWHTAB	,8,WHTAB ,75)	READ1357
CALL SETUP(BHWCDTAB	,8,WCDTAB ,75)	READ1358
CALL SETUP(BHAWREF	,8,AWREF)	READ1359

C			READ1360
C	THRUSTING OPTION INPUTS		READ1351
	CALL SETUP(BHNTHRST ,4,NTHRST)		READ1352
	CALL SETUP(BHTON ,8,TUN)		READ1353
	CALL SETUP(BHTOFF ,8,TUFF)		READ1354
	CALL SETUP(BHZON ,8,ZON)		READ1355
	CALL SETUP(BHZUFF ,8,ZOFF)		READ1356
	CALL SETUP(BHITHRST ,4,ITHRST)		READ1357
	CALL SETUP(BHTHMO ,8,THMO ,25)		READ1358
	CALL SETUP(BHTHDELZ ,8,THDELZ ,25)		READ1359
	CALL SETUP(BHTHDELZ ,8,THDELZ ,25)		READ1370
	CALL SETUP(BHTHO ,8,THO)		READ1371
	CALL SETUP(BHTISP ,8,ISP)		READ1372
	CALL SETUP(BHAE ,8,AE)		READ1373
	CALL SETUP(BHPSIZET ,8,PSIZET)		READ1374
	CALL SETUP(BHTHEZET ,8,THEZET)		READ1375
	CALL SETUP(BHDELY ,8,DELY)		READ1376
	CALL SETUP(BHDELZ ,8,DELZ)		READ1377
C			READ1378
C	INPUTS FOR TABULAR INPUT ATMOSPHERE		READ1379
	CALL SETUP(BHIATMOS ,4,IATMOS)		READ1380
	CALL SETUP(BHUPBNDZ ,8,UPBNDZ)		READ1381
	CALL SETUP(BHONBNDZ ,8,ONBNDZ)		READ1382
	CALL SETUP(BHTBATMZ ,8,TBATMZ ,50)		READ1383
	CALL SETUP(BHTABRHO ,8,TABRHO ,50)		READ1384
	CALL SETUP(BHTABSND ,8,TABSND ,50)		READ1385
C			READ1386
C	PLOTTING AND TAPE INPUTS		READ1387
	CALL SETUP(BHNPLUT ,4,NPLOT ,5)		READ1388
	CALL SETUP(BHITAPE ,4,ITAPE)		READ1389
C			READ1390
C	PHYSICAL CONSTANTS		READ1391
	CALL SETUP(BHC ,8,C)		READ1392
	CALL SETUP(BHR ,8,R)		READ1393
	CALL SETUP(BHGAMMA ,8,GAMMA)		READ1394
	CALL SETUP(BHRE ,8,RE)		READ1395
	CALL SETUP(BHCAPG ,8,CAPG)		READ1396
	CALL SETUP(BHG ,8,G)		READ1397
	CALL SETUP(BHZETA ,8,ZETA)		READ1398
	CALL SETUP(BHSIG ,8,SIG)		READ1399
	CALL SETUP(BHMW ,8,MW)		READ1400
C			READ1401
C	CURVE FIT CONSTANTS		READ1402
	CALL SETUP(BHA ,8,A ,516)		READ1403
	CALL SETUP(BHB ,8,B ,211)		READ1404
C			READ1405
	2 CALL READIN(INCOL1,899)		READ1406
	IOP(73) = 0		READ1407
	DO 66 I = 1, 3		READ1408
	J = I + 63		READ1409
	IOP(J) = 0		READ1410
	DO 55 K = I, 21, 3		READ1411
	IF(IOP(K).EQ.1) IOP(J) = 1		READ1412
55	CONTINUE		READ1413
	IF(IOP(J).EQ.1) IOP(73) = 1		READ1414
66	CONTINUE		READ1415
	IOP(74) = 0		READ1416
	ITEM = 0		READ1417
	DO 88 I = 22, 27		READ1418
	J = I + 45		READ1419

	IOP(J) = 0	READ1420
	IF(IOP(JG10).EQ.1) ITEM = 1	READ1421
	DO 77 K = 1, 63, 6	READ1422
	IF(IOP(K).EQ.1) IOP(J) = 1	READ1423
77	CONTINUE	READ1424
	IF(ITEM.EQ.1) IOP(74) = 1	READ1425
	IF(IOP(J).EQ.1) IOP(74) = 1	READ1426
88	CONTINUE	READ1427
	CALL HEDING(6)	READ1428
	GO TO 98	READ1429
99	IF(IMPLUT.EQ.1) CALL AVPLT(H,H,H,H,H,S,1)	READ1430
	IF(ITAPE.NE.0)END FILE 8	READ1431
	IF(ITAPE.NE.0)REWIND 8	READ1432
	CALL EXIT	READ1433
98	RETURN	READ1434
	END	READ1435

C	READY	READY	1
	SUBROUTINE READY	READY	2
	IMPLICIT REAL*8 (A-H,O-Z)	READY	3
	COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)	READY	4
	1,GS,GSP,GTP,GSS,GTT,GSB,F,FP,FB,FO,E,P,TD,RS,SL,Z,Q,A,EL,	READY	5
	2DELTA,FAC,C(40,10)	READY	6
	COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC	READY	7
	COMMON /NALTFC/F,SM,NALT	READY	8
	COMMON /END/ ITERM	READY	9
C	READY	READY	10
C		READY	11
C		READY	12
C	SUBROUTINE READY PERFORMS THE CALCULATIONS OF THE DAVIDON	READY	13
C	MINIMIZATION METHOD WHICH ESTABLISH A DIRECTION ALONG WHICH TO	READY	14
C	SEARCH FOR A RELATIVE MINIMUM AND BOX OFF AN INTERVAL IN	READY	15
C	THIS DIRECTION WITHIN WHICH A RELATIVE MINIMUM IS LOCATED.	READY	16
C		READY	17
C		READY	18
	L=L	READY	19
	GO TO (200,201),L	READY	20
	200 IT=1	READY	21
	201 CALL MATMP (M,N,H,G,S)	READY	22
C		READY	23
	202 DO 203 I=1,N	READY	24
	203 S(I)=-S(I)	READY	25
C		READY	26
	204 M=1	READY	27
	205 CALL MATMP (M,N,S,G,GS)	READY	28
	206 IF (GSGE)207,227,227	READY	29
C		READY	30
	207 TP1 = -FUSM(F/GS)	READY	31
	WRITE(6,1000) GS, E, TP1, IT	READY	32
	EL = DMIN(2.000, TP1)	READY	33
	209 SL=-GS	READY	34
C		READY	35
	210 DO 211 I=1,N	READY	36
	211 XP(I)=X(I)+EL*S(I)	READY	37
C		READY	38
	212 M1=2	READY	39
	213 CALL FCN(M,GP,FP,XP,M1)	READY	40
	IF(ITERM.EQ.0) GO TO 214	READY	41
	DO 100 I = 1,N	READY	42
100	X(I) = XP(I)	READY	43
	GO TO 222	READY	44
C		READY	45
	214 M=1	READY	46
	215 CALL MATMP (M,N,S,GP,GSP)	READY	47
	216 IF (GSP)217,229,229	READY	48
	217 IF (FP-F)218,229,229	READY	49
	218 WRITE(6,1)	READY	50
C		READY	51
	IF(NALT) 231, 231, 10	READY	52
10	EL = 2.00*EL	READY	53
	GO TO 210	READY	54
	231 F0=FP	READY	55
	232 DO 234 I=1,N	READY	56
	233 G0(I)=GP(I)	READY	57
	234 T(I)=XPE(I)	READY	58
	220 IF (EL-2.0)221,223,223	READY	59

221 L=3	READY 60
C	READY 61
222 RETURN	READY 62
C	READY 63
223 DELTA=2.0*DELTA	READY 64
224 T0=1.0/SL	READY 65
225 L=2	READY 66
226 GO TO 222	READY 67
227 L=1	READY 68
228 GO TO 222	READY 69
229 L=4	READY 70
230 GO TO 222	READY 71
C	READY 72
1. FORMAT (10HOUNDERSHOT)	READY 73
1000 FORMAT(1H 'GS ='1PE15.7,' ERR ='E15.7,' TP1 ='E15.7,' IT ='17)	READY 74
END	READY 75

C	REDUCE	REDUC 1
	IMPLICIT REAL*8(A-H,O-Z)	REDUC 2
	SUBROUTINE REDUCE(CUP,KRED,IRED,LRED,WRF)	REDUC 3
	COMMON /IXCOM/ XCOM(200), ICOM(200)	REDUC 4
	COMMON OCCUR,NOCCUR	REDUC 5
	COMMON /IDNUS/ ID1(50), ID2(50)	REDUC 6
	COMMON /IOPT/IPROC,IN,NCONS,IPNT,LEX,LIMIT,IRAND	REDUC 7
	COMMON /END/ ITEM	REDUC 8
	DIMENSION CUP(20)	REDUC 9
	DIMENSION OCCUR(4000),NOCCUR(30)	REDUC 10
C		REDUC 11
C		REDUC 12
C	SUBROUTINE REDUCE CAN MULTIPLY ANY SPECIFIED ELEMENT OF THE	REDUC 13
C	OCCUR ARRAY BY THE REDUCTION FACTOR WRF AND SET THE FIRST	REDUC 14
C	ELEMENT OF THE CUP ARRAY EQUAL TO ANY SPECIFIED ELEMENT OF THE	REDUC 15
C	OCCUR ARRAY MULTIPLIED BY THE FACTOR WRF.	REDUC 16
C		REDUC 17
C		REDUC 18
C		REDUC 19
	IZ = ID2(1)	REDUC 20
	IRED = IRED + 1	REDUC 21
	IF(IRED.EQ.LRED) GO TO 100	REDUC 22
	IF(ITEM) 10, 50, 60	REDUC 23
C		REDUC 24
C		REDUC 25
10	DO 20 I = 1, IN	REDUC 26
	IF(ID1(I).EQ.IZ) GO TO 40	REDUC 27
20	CONTINUE	REDUC 28
C		REDUC 29
	IF(ICOM(3).EQ.1) GO TO 40	REDUC 30
C		REDUC 31
30	OCCUR(IZ) = WRF*OCCUR(IZ)	REDUC 32
	KRED = 0	REDUC 33
	GO TO 200	REDUC 34
C		REDUC 35
40	CUP(I) = WRF*OCCUR(IZ)	REDUC 36
	KRED = 0	REDUC 37
	GO TO 200	REDUC 38
C		REDUC 39
50	KRED = 1	REDUC 40
C	50 MAY BE USED FOR LATER IMPLEMENTATION	REDUC 41
C		REDUC 42
C		REDUC 43
	GO TO 200	REDUC 44
C		REDUC 45
60	KRED = 1	REDUC 46
	GO TO 200	REDUC 47
C		REDUC 48
C		REDUC 49
100	KRED = -1	REDUC 50
C		REDUC 51
200	RETURN	REDUC 52
	END	REDUC 53

CRITOUT	RITOU 1
IMPLICIT REAL*8 (A-H,O-Z)	RITOU 2
SUBROUTINE RITOUT	RITOU 3
C	RITOU 4
DIMENSION ALMAX(200),ALMIN(200),TAMAX(200),TAMIN(200),ZMAX(200),	RITOU 5
1ZMIN(200),FMAX(200),FMIN(200),OCCUR(4000),NOCCUR(30)	RITOU 6
C	RITOU 7
EQUIVALENCE	RITOU 8
1(OCCUR(1044),ALMAX(1)),(OCCUR(1244),ALMIN(1))	RITOU 9
2(OCCUR(1444),TAMAX(1)),(OCCUR(1644),TAMIN(1))	RITOU 10
3(OCCUR(1844),ZMAX(1)),(OCCUR(2044),ZMIN(1))	RITOU 11
4(OCCUR(2244),FMAX(1)),(OCCUR(2444),FMIN(1))	RITOU 12
EQUIVALENCE	RITOU 13
1(NOCCUR(11),IKMAX),(NOCCUR(12),IKMIN),(NOCCUR(14),NPRINT)	RITOU 14
C	RITOU 15
COMMON OCCUR,NOCCUR	RITOU 16
C	RITOU 17
C	RITOU 18
C	RITOU 19
C	RITOU 20
C	RITOU 21
C	RITOU 22
C	RITOU 23
IF(NPRINT.EQ.0)RETURN	RITOU 24
51 WRITE(6,1051)	RITOU 25
1051 FORMAT(1H0 50X,23HMAXIMUMS IN ALPHA PRIME/40X,4HTIME,9X,1HZ,12X,	RITOU 26
13HSMF,6X,11HALPHA PRIME)	RITOU 27
C	RITOU 28
DO 52 J=1,IKMAX	RITOU 29
ALMAX(J)=57.29578DO*ALMAX(J)	RITOU 30
WRITE(6,1052)TAMAX(J),ZMAX(J),FMAX(J),ALMAX(J)	RITOU 31
1052 FORMAT(1H 38X,F7.2,2E14.5,4X,F8.3)	RITOU 32
52 CONTINUE	RITOU 33
WRITE(6,1053)	RITOU 34
1053 FORMAT(1H0 50X,23HMINIMUMS IN ALPHA PRIME/40X,4HTIME,9X,1HZ,12X,	RITOU 35
13HSMF,6X,11HALPHA PRIME)	RITOU 36
C	RITOU 37
DO 53 J=1,IKMIN	RITOU 38
ALMIN(J)=57.29578DO*ALMIN(J)	RITOU 39
WRITE(6,1052)TAMIN(J),ZMIN(J),FMIN(J),ALMIN(J)	RITOU 40
53 CONTINUE	RITOU 41
RETURN	RITOU 42
END	

C	ROSBRK	ROSBRK 1
	SUBROUTINE ROSBRK	ROSBRK 2
	IMPLICIT REAL*8 (A-H,O-Z)	ROSBRK 3
	COMMON OCCUR, NOCCUR	ROSBRK 4
	COMMON/BLK0/HH(1600),P(40),D(40),E(40),DP(40),A(40),	ROSBRK 5
	1 B(40),GS,GSP,GTP,GSS,GTT,GSB,FZZ,FP,FB,FO,Z,PZZ,TJ,RS,SL,ZZZ,	ROSBRK 6
	1QZZ,AXX,EL	ROSBRK 7
	2DELXX,FAK,CXX(40,10)	ROSBRK 8
	COMMON/DOPT/EO(20)	ROSBRK 9
	COMMON/END/ITERM	ROSBRK 10
	COMMON/IOPT/IPROC,NMAX,NCUNS,IPNT,LEX,LIMIT,IRAND	ROSBRK 11
	COMMON /XCOM/ XCOM(200), ICOM(200)	ROSBRK 12
	DIMENSION C(20,20)	ROSBRK 13
	DIMENSION OCCUR(4000), NOCCUR(30)	ROSBRK 14
	EQUIVALENCE (HH(401),C(1))	ROSBRK 15
	EQUIVALENCE (XCOM(1),ALPHA)	ROSBRK 16
	EQUIVALENCE (XCOM(2),BETA)	ROSBRK 17
	EQUIVALENCE (XCOM(3),GAMMA)	ROSBRK 18
	EQUIVALENCE (XCOM(4),DEL)	ROSBRK 19
	EQUIVALENCE (XCOM(5),RATU)	ROSBRK 20
	EQUIVALENCE (XCOM(6),TOL)	ROSBRK 21
C		ROSBRK 22
C		ROSBRK 23
C	SUBROUTINE ROSBRK ATTEMPTS TO MINIMIZE THE VALUE OF A FUNC-	ROSBRK 24
C	TION U(P1,P2,...,P(NMAX)) WHERE THE P'S ARE ADJUSTABLE	ROSBRK 25
C	DESIGN PARAMETERS BY THE ROSENBRCK ROTATING COORDINATE MINI-	ROSBRK 26
C	MIZATION TECHNIQUE.	ROSBRK 27
C		ROSBRK 28
C	NMAX IS THE NUMBER OF PARAMETERS TO BE ADJUSTED. ITS MAXIMUM	ROSBRK 29
C	VALUE IS 20.	ROSBRK 30
C		ROSBRK 31
C		ROSBRK 32
	NTRIA = 0	ROSBRK 33
	NSTAG = 0	ROSBRK 34
	INDEX = 0	ROSBRK 35
C		ROSBRK 36
123	CALL FEV(NMAX,P,U)	ROSBRK 37
	IF(ITERM.NE.0) GO TO 600	ROSBRK 38
C		ROSBRK 39
	IF(NTRIA.LE.LIMIT) GO TO 610	ROSBRK 40
	WRITE(6,611)	ROSBRK 41
611	FORMAT(1H0 34HNTKIA GREATER THAN LIMIT IN ROSBRK)	ROSBRK 42
	ITERM = 1	ROSBRK 43
	GO TO 120	ROSBRK 44
610	INDEX = INDEX+1	ROSBRK 45
C		ROSBRK 46
	IF(NTRIA.NE.0) GO TO 103	ROSBRK 47
C		ROSBRK 48
	ULAST = U	ROSBRK 49
	USTAG = U	ROSBRK 50
	UPREV = U	ROSBRK 51
	NSUCC = 0	ROSBRK 52
	NTRIA = 1	ROSBRK 53
	DU 104 I=1,NMAX	ROSBRK 54
104	E(I) = EO(I)	ROSBRK 55
C		ROSBRK 56
C	C MATRIX IS SET TO A UNIT MATRIX	ROSBRK 57
C		ROSBRK 58
	DU 105 I=1,NMAX	ROSBRK 59

DO 105 J=1,NMAX	RUSBR 60
IF(I-J) 106,107,106	RUSBR 61
106 C(I,J) = 0.00	RUSBR 62
GO TO 105	RUSBR 63
107 C(I,J) = 1.00	RUSBR 64
105 CONTINUE	RUSBR 65
N=1	RUSBR 66
D(1) = 0.00	RUSBR 67
GO TO 108	RUSBR 68
C	RUSBR 69
C SECOND AND SUBSEQUENT TRIALS START HERE	RUSBR 70
C	RUSBR 71
C	RUSBR 72
103 NTRIA = NTRIA & 1	RUSBR 73
IF(U-ULAST*(1.00&TOL)) 109,109,110	RUSBR 74
C	RUSBR 75
C IF TRIAL WAS A SUCCESS, START HERE	RUSBR 76
C	RUSBR 77
109 ULAST = U	RUSBR 78
NSUCC = NSUCC & 1	RUSBR 79
D(N) = D(N) & E(N)	RUSBR 80
IF(NSUCC.GE.20) GO TO 50	RUSBR 81
E(N) = ALPHA*E(N)	RUSBR 82
GO TO 108	RUSBR 83
C	RUSBR 84
C IF TRIAL WAS A FAILURE, START HERE	RUSBR 85
110 CONTINUE	RUSBR 86
WRITE(6,306)NTRIA,N,NSTAG,NSUCC,U,NMAX,(P(I),I=1,NMAX)	RUSBR 87
306 FORMAT(1H024H NTRIA N NSTAG NSUCC19X1HU/ 1H 416,F20.10/1H	RUSBR 88
1 13HP(I) I = 1 TO 13/(1H 6F20.10))	RUSBR 89
DO 111 I=1,NMAX	RUSBR 90
111 P(I) = P(I) - DP(I)	RUSBR 91
U = ULAST	RUSBR 92
IF(NSUCC) 112,112,113	RUSBR 93
112 E(N) = -BETA*E(N)	RUSBR 94
GO TO 125	RUSBR 95
50 WRITE(6,922) NSUCC	RUSBR 96
922 FORMAT(1H0 15HWARNING-NSUCC= 16,1X-22HGOING TO NEXT VARIABLE)	RUSBR 97
C	RUSBR 98
C SEARCH IN ONE DIRECTION	RUSBR 99
C	RUSBR100
113 E(N) = GAMMA*D(N)	RUSBR101
C IF N = NMAX, THIS STAGE IS COMPLETE.	RUSBR102
IF(N.EQ.NMAX)GO TO 115	RUSBR103
N=N&1	RUSBR104
D(N) = 0.00	RUSBR105
NSUCC = 0	RUSBR106
GO TO 125	RUSBR107
C	RUSBR108
C STOPPING CRITERIA IS APPLIED AFTER 2ND AND SUBSEQUENT STAGES	RUSBR109
C	RUSBR110
115 IF(NSTAG) 116,116,117	RUSBR111
117 ULIM = UPREV - DEL*UPREV	RUSBR112
IF((UPREV-USTAG).EQ.0.00) GO TO 121	RUSBR113
URAT = (USTAG-ULAST)/(UPREV-USTAG)	RUSBR114
IF(ULAST = ULIM)121,121,118	RUSBR115
118 IF(USTAG = ULIM)121,121,119	RUSBR116
119 IF(URAT = RATU)120,121,121	RUSBR117
C	RUSBR118
120 WRITE(6,301)	RUSBR119

301	FORMAT(1H0 26HROSBK SEARCH IS COMPLETE.)	ROSBR120
	WRITE(6,302) ULAST,(P(I),I=1,NMAX)	ROSBR121
302	FORMAT(1H037HFINAL PERFORMANCE FUNCTION. ULAST = E20.10/1H	RUSBR122
	120HFINAL PARAMETERS ARE /(1H 6E20.10))	RUSBR123
	WRITE(6,303)NTRIA,NSTAG	RUSBR124
303	FORMAT(1H018HNUMBER OF TRIALS = 15/1H 18HNUMBER OF STAGES = 15)	RUSBR125
	GO TO 600	RUSBR126
C		RUSBR127
C	SEARCH IS COMPLETE	RUSBR128
C		RUSBR129
C		RUSBR130
C	SEARCH WILL CONTINUE IN A NEW COORDINATE SYSTEM.	RUSBR131
	121 UPREV = USTAG	RUSBR132
	110 USTAG = ULAST	RUSBR133
	NSTAG = NSTAG&1	RUSBR134
C		RUSBR135
C	DEFINE NEW COORDINATE SYSTEM.	RUSBR136
	CALL GRAM	RUSBR137
	WRITE(6,304)	RUSBR138
304	FORMAT(1H0 20HC MATRIX FROM ROSBRK/)	RUSBR139
	DU 200 I=1,NMAX	RUSBR140
200	WRITE(6,305) (C(I,J),J=1,NMAX)	RUSBR141
305	FORMAT(12(1X,1PE9.2))	RUSBR142
129	N=1	RUSBR143
	D(1)= 0.00	RUSBR144
	NSUCC=0	RUSBR145
108	CONTINUE	RUSBR146
	WRITE(6,306)NTRIA,N,NSTAG,NSUCC,U,NMAX,(P(I),I=1,NMAX)	RUSBR147
C		RUSBR148
C	DEFINE COORDINATES FOR NEXT TRIAL.	RUSBR149
	125 DO 122 I=1,NMAX	RUSBR150
	DP(I)=C(I,N)*E(N)	RUSBR151
	122 P(I)=P(I)&DP(I)	RUSBR152
C		RUSBR153
	WRITE(6,500)(DP(I),I=1,NMAX)	RUSBR154
	WRITE(6,501)NMAX,(P(I),I=1,NMAX)	RUSBR155
	WRITE(6,503) N,E(N)	RUSBR156
500	FORMAT(1H014HIN ROSBRK DP =/(12H 1P6E20.7))	RUSBR157
501	FORMAT(1H013HP(I) I = 2 TO 13/(1H 1P6E20.7))	RUSBR158
503	FORMAT(1H0 13HIN ROSBRK N = 13,2X6HE(N) =E20.10)	RUSBR159
	GO TO 123	RUSBR160
600	RETURN	RUSBR161
	END	RUSBR162

CROTATE	ROTAT 1
IMPLICIT REAL*8 (A-H,O-Z)	ROTAT 2
SUBROUTINE ROTATE(DERIV,LP)	ROTAT 3
C	ROTAT 4
REAL*8 MX,MY,MZ	ROTAT 5
REAL*8 NUMER1,NUMER2	ROTAT 6
C	ROTAT 7
DIMENSION TABIX(50),TABI(50),TABZ(50),OCCUR(4000),NOCCUR(30),	ROTAT 8
1 DERIV(16),XJR(25),XJI(25)	ROTAT 9
DIMENSION XNR(25),XNI(25)	ROTAT 10
C	ROTAT 11
EQUIVALENCE	ROTAT 12
1(OCCUR(001),AREF),(OCCUR(002),ALPHA),(OCCUR(003),ALPRIM),	ROTAT 13
2(OCCUR(012),CMALP),(OCCUR(013),CMALP),(OCCUR(015),CD),	ROTAT 14
3(OCCUR(020),CMQ),(OCCUR(021),D),(OCCUR(027),G),	ROTAT 15
4(OCCUR(042),PI),(OCCUR(043),P),(OCCUR(044),PHI),	ROTAT 16
5(OCCUR(045),PSI),(OCCUR(050),Q),(OCCUR(051),QD),	ROTAT 17
6(OCCUR(055),RHUINF),(OCCUR(056),RHOINI),	ROTAT 18
7(OCCUR(065),SMR),(OCCUR(071),THEALP),(OCCUR(075),T),	ROTAT 19
8(OCCUR(077),TCRIT),(OCCUR(078),TECON),(OCCUR(081),SMF),	ROTAT 20
9(OCCUR(082),V),(OCCUR(084),W),(OCCUR(090),XBAR),	ROTAT 21
X(OCCUR(091),Z)	ROTAT 22
EQUIVALENCE	ROTAT 23
1(OCCUR(108),ZO),(OCCUR(113),THEALO),(OCCUR(122),ALST),	ROTAT 24
2(OCCUR(130),ALWIG2),(OCCUR(131),ALBARP),(OCCUR(202),CM),	ROTAT 25
3(OCCUR(203),CN),(OCCUR(210),MX),(OCCUR(211),MY),	ROTAT 26
4(OCCUR(212),MZ),(OCCUR(221),SINGO),(OCCUR(229),ALALI),	ROTAT 27
5(OCCUR(237),XUP),	ROTAT 28
6(OCCUR(894),TABI(1)),(OCCUR(944),TABIX(1)),(OCCUR(994),TABZ(1))	ROTAT 29
C	ROTAT 30
EQUIVALENCE	ROTAT 31
1(NOCCUR(04),MAXTAB),(NOCCUR(07),LOPT)	ROTAT 32
C	ROTAT 33
COMMON OCCUR,NOCCUR	ROTAT 34
C	ROTAT 35
C	ROTAT 36
C	ROTAT 37
C	ROTAT 38
C	ROTAT 39
C	ROTAT 40
C	ROTAT 41
C	ROTAT 42
C	ROTAT 43
C	ROTAT 44
C	ROTAT 45
C	ROTAT 46
C	ROTAT 47
C	ROTAT 48
C	ROTAT 49
C	ROTAT 50
C	ROTAT 51
C	ROTAT 52
C	ROTAT 53
C	ROTAT 54
C	ROTAT 55
C	ROTAT 56
C	ROTAT 57
C	ROTAT 58
C	ROTAT 59

C		ROTAT 60
C	FOR LOPT=0 USING A THREE DEGREE OF FREEDOM IN ROTATION METHOD	ROTAT 61
C	WHICH CALCULATES THE DERIVATIVES WITH RESPECT TO TIME OF THE	ROTAT 62
C	EULER ANGLES - PSI, THETA SUB ALPHA, AND PHI - AND OF THE	ROTAT 63
C	ANGULAR VELOCITIES - P,Q,SMR.	ROTAT 64
C	4 CONTINUE	ROTAT 65
C	TRIGONOMETRIC FUNCTIONS OF THE EULER ANGLES.	ROTAT 66
C	3 SINP=SIN(PHI)	ROTAT 67
C	COSP=COS(PHI)	ROTAT 68
C	SINI=SIN(PSI)	ROTAT 69
C	COSI=COS(PSI)	ROTAT 70
C	SINL=SIN(THALP)	ROTAT 71
C	COSL=COS(THALP)	ROTAT 72
C	CALCULATION OF THE DERIVATIVES OF THE EULER ANGLES, WHERE	ROTAT 73
C	DERIV(8)=PSI DOT, DERIV(9)=THETA SUB ALPHA DOT, DERIV(10)=	ROTAT 74
C	PHI DOT.	ROTAT 75
C	DERIV(8)=(SMR*COSP*Q*SINP)/COSL	ROTAT 76
C	DERIV(9)=Q*COSP-SMR*SINP	ROTAT 77
C	DERIV(10)=P*DERIV(8)*SINL	ROTAT 78
C	COSLP=COSL*COSI	ROTAT 79
C	DETERMINATION OF THE ANGLE OF ATTACK FROM THE ARCCOSINE OF	ROTAT 80
C	THE PRODUCT - COS(THETA SUB ALPHA)*COS(PSI).	ROTAT 81
C	IF(COSLP.LT.0.00)GO TO 100	ROTAT 82
C	ALPRM=ACOSK(COSLP)	ROTAT 83
C	CN=CNALP*ALPRM	ROTAT 84
C	GO TO 4	ROTAT 85
C	100 WRITE(6,101)	ROTAT 86
C	101 FORMAT(1H0 10X,76H THE ARGUMENT OF ALPRM IN SUBROUTINE ROTATE IS	ROTAT 87
C	NEGATIVE. GOING TO NEXT CASE.)	ROTAT 88
C	LP=6	ROTAT 89
C	RETURN	ROTAT 90
C	MOMENTS OF INERTIA ARE DETERMINED FROM THE INPUT TABLES.	ROTAT 91
C	4 CONTINUE	ROTAT 92
C	LL=-1	ROTAT 93
C	UI=TABLE(Z,TABZ,TAB1,MAXTAB,LL)	ROTAT 94
C	UIX=TABLE(Z,TABZ,TABIX,MAXTAB,LL)	ROTAT 95
C	IF(UI.EQ.0.00)GO TO 13	ROTAT 96
C	IF(UIX.EQ.0.00)GO TO 14	ROTAT 97
C	6 CM=CNALP*ALPRM	ROTAT 98
C		ROTAT 99
C		ROTAT 100
C	CALCULATION OF DERIVATIVES OF THE ANGULAR VELOCITIES, WHERE	ROTAT 101
C	DERIV(11)=Q DOT, DERIV(12)=SMR DOT, AND DERIV(13)=P DOT.	ROTAT 102
C	TEM1=Q*AREF*Q/UI	ROTAT 103
C	TEM2=0.00	ROTAT 104
C	IF(ALPRM.NE.0.00)TEM2=CM/SIN(ALPRM)	ROTAT 105
C	TEM3=SINL*COSI	ROTAT 106
C	TEM4=0.500*Q*CMQ/V	ROTAT 107
C	TEM5=(UI-UIX)*P/UI	ROTAT 108
C	DERIV(11)=TEM1*(TEM2*(TEM3*COSP*SINL*SINP)&TEM4*Q)&TEM5*SMR&MY/JI	ROTAT 109
C	DERIV(12)=TEM1*(TEM2*(SINI*COSP-TEM3*SINP)&TEM4*SMR)-TEM5*Q&MZ/JI	ROTAT 110
C	DERIV(13)=MX/UIX	ROTAT 111
C		ROTAT 112
C	IF(XBAR.LT.XUP)GO TO 10	ROTAT 113
C	11 ALPHA=ALPRM	ROTAT 114
C	RETURN	ROTAT 115
C		ROTAT 116
C	IN NON-CONTINUUM FLOW REGIME, PERIOD OF THE CYCLE IS TESTED	ROTAT 117
C	TO DETERMINE APPROPRIATE ANGLE OF ATTACK FOR USE IN DRAGCO.	ROTAT 118
C	10 IF(T.EQ.0.00)GO TO 11	ROTAT 119

	IF(T.GE.TCRIT)GO TO 11	ROTAT120
	IF(T.GE.TECON)GO TO 12	ROTAT121
	IF(P.EQ.0.00)ALPHA=2.00*ALWIG2/PI	ROTAT122
	IF(P.NE.0.00)ALPHA=0.500*(ALWIG2&ALBARP)	ROTAT123
	GO TO 12	ROTAT124
C		ROTAT125
C		ROTAT126
C		ROTAT127
C	SIMPLIFIED ANGLE OF ATTACK SOLUTION WHICH REQUIRES NO	ROTAT128
C	CALCULATION OF DERIVATIVES OR SUBSEQUENT INTEGRATION.	ROTAT129
	15 LL=-1	ROTAT130
	IF(Z.LE.100.000)GO TO 90	ROTAT131
	UI=TABLE(Z,TABZ,TABI,MAXTAB,LL)	ROTAT132
	IF(UI.EQ.0.000)GO TO 13	ROTAT133
	BETAZ=-DLOG(RHOINI/0.07647400)	ROTAT134
	IF(BETAZ.LE.0.000100)BETAZ=0.000100	ROTAT135
	IF(Z.EQ.Z0)CU=0.8	ROTAT136
	AK1=AREF*G*Z*(2.000*CD-CNALP&CMU*D*D*W/(UI*G))/	ROTAT137
	1(4.000*BETAZ*W*DABS(SINGO))	ROTAT138
	AK2=AREF*G*Z*(CNALP-LL)*BETAZ*DABS(SINGO)/Z-CNALP*D*W/(UI*G)/	ROTAT139
	1(2.000*W*(BETAZ*SINGO)**2)	ROTAT140
	17 CONTINUE	ROTAT141
	TBAR=-2.000*PI*Z/(BETAZ*V*SINGO*DSQRT(AK2*RHOINF))	ROTAT142
	IF(TBAR.LE.TECON)GO TO 16	ROTAT143
	IF(TBAR.LT.1.00-10)GO TO 16	ROTAT144
C		ROTAT145
C	OSCILLATORY SOLUTION FOR CYCLE TIMES, TBAR, GREATER THAN	ROTAT146
C	TECON.	ROTAT147
	SMF=1.000/TBAR	ROTAT148
	TEM=2.000*DSQRT((AK1&AK2)*RHOINF)	ROTAT149
	CALL BESSEL(TEM,0.000,1.00-5,10,XJR,XJI,1)	ROTAT150
	CALL NEUMAY(1.00-5,TEM,0.000,10,1.001,XNR,XNI,XJR(1),XJI(1),XJR(2)	ROTAT151
	1,XJI(2),1)	ROTAT152
	IF(Z.NE.Z0)GO TO 2	ROTAT153
	UTEMDZ=-0.500*TEM*BETAZ/Z	ROTAT154
	DENAC2=UTEMDZ*(-XNR(2)*XJR(1)&XJR(2)*XNR(1))	ROTAT155
	NUMER1=(XJR(1)*AK1*THEALO*2.37690-3*BETAZ)/(Z*DEXP(AK1*RHOINF&	ROTAT156
	1BETAZ))	ROTAT157
	NUMER2=THEALO*UTEMDZ*(-XJR(2))*DEXP(-AK1*RHOINF)	ROTAT158
	AC2=(NUMER1-NUMER2)/DENAC2	ROTAT159
	AC1=(THEALO*DEXP(-AK1*RHOINF)-AC2*XNR(1))/XJR(1)	ROTAT160
	ALALI=DEXP(AK1*RHOINF)/DSQRT(PI*DSQRT(AK2*RHOINF))	ROTAT161
	2 CONTINUE	ROTAT162
	ALPKIM=DEXP(AK1*RHOINF)*(AC1*XJR(1)&AC2*XNR(1))	ROTAT163
	CN=CNALP*ALPRIM	ROTAT164
	CM=CMALP*ALPRIM	ROTAT165
	ALPHA=ALPRIM	ROTAT166
	RETURN	ROTAT167
C		ROTAT168
C	ENVELOPE ANGLE OF ATTACK CALCULATION FOR TBAR LESS THAN OR	ROTAT169
C	EQUAL TO TECON.	ROTAT170
	16 ALALO=DEXP(AK1*RHOINF)/DSQRT(PI*DSQRT(AK2*RHOINF))	ROTAT171
	IF(Z.EQ.Z0)ALALI=ALALO	ROTAT172
	ALPKIM=0.6366100*THEALC*ALALO	ROTAT173
	ALPRIM=ALPKIM/ALALI	ROTAT174
	SMF=0.000	ROTAT175
C		ROTAT176
C	IF ANGLE OF ATTACK ENVELOPE VALUE IS LESS THAN OR EQUAL TO	ROTAT177
C	ALST, INPUT ALPHA STOP VALUE, THE TRAJECTORY IS SWITCHED OVER	ROTAT178
C	TO A PARTICLE TRAJECTORY, LUPT=1, AND ANGLE OF ATTACK IS SET	ROTAT179

C	TO ZERD.	ROTAT180
	IF(ALPRIM.GT.ALST)GO TO 92	ROTAT181
90	ALPRIM=0.000	ROTAT182
	LOPT=1	ROTAT183
92	CONTINUE	ROTAT184
	CN=CNALP*ALPRIM	ROTAT185
	CM=CMALP*ALPRIM	ROTAT186
	ALPHA=ALPRIM	ROTAT187
12	RETURN	ROTAT188
13	WRITE(6,1013)Z	ROTAT189
	LP=0	ROTAT190
	RETURN	ROTAT191
14	WRITE(6,1014)Z	ROTAT192
	LP=0	ROTAT193
	RETURN	ROTAT194
1013	FORMAT(1H0 10X,37HRESULT OF TABI LOOKUP IS 0.000 FOR Z=E12.5)	ROTAT195
1014	FORMAT(1H0 10X,38HRESULT OF TABIX LOOKUP IS 0.000 FOR Z=E12.5)	ROTAT196
	END	ROTAT197

CSAVEDV	SAVED 1
	IMPLICIT REAL*8(A-H,O-Z)	SAVED 2
	SUBROUTINE SAVEDV(K,DVAR,OCCUR)	SAVED 3
	DIMENSION DVAR(50),OCCUR(4000)	SAVED 4
C	SAVED 5
C	SAVED 6
C	SUBROUTINE SAVEDV STORES THE VALUES OF OCCUR(133-145),	SAVED 7
C	OCCUR(205-209), AND OCCUR(222) IN LOCATIONS DVAR(1-19)	SAVED 8
C	RESPECTIVELY AND LATER RESTORES THESE VALUES TO THE OCCUR	SAVED 9
C	ARRAY.	SAVED 10
C	SAVED 11
C	SAVED 12
	K = K	SAVED 13
	GO TO (100,200), K	SAVED 14
100	DO 111 I = 1, 13	SAVED 15
111	DVAR(I) = OCCUR(I&132)	SAVED 16
	DO 112 I = 14, 18	SAVED 17
112	DVAR(I) = OCCUR(I&191)	SAVED 18
	DVAR(19) = OCCUR(222)	SAVED 19
	RETURN	SAVED 20
200	DO 222 I = 1, 13	SAVED 21
222	OCCUR(I&132) = DVAR(I)	SAVED 22
	DO 223 I = 14, 18	SAVED 23
223	OCCUR(I&191) = DVAR(I)	SAVED 24
	OCCUR(222) = DVAR(19)	SAVED 25
	RETURN	SAVED 26
	END	SAVED 27

```

CSCREEN-----
SUBROUTINE SCREEN(N,X,ISUC,VAL,D)
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 LA1,LA2,LA1F,LAMDA1,LAMDA2,LAMD1F
COMMON OCCUR,NOCCUR
DIMENSION PENLTY(25),-----A(25),OCCUR(4000),-----
INOCCUR(30)
EQUIVALENCE (OCCUR(42),P1),(OCCUR(102),T0),(OCCUR(108),Z0),
1(OCCUR(121),ZST),(OCCUR(123),TST),(NOCCUR(15),NGEOM),
2(OCCUR(22),NTHRST)
EQUIVALENCE
1(OCCUR(170),W1F),(OCCUR(150),THET1F),(OCCUR(159),RN1F),
2(OCCUR(147),R31F),(OCCUR(151),LAMD1F),(OCCUR(146),LA1F)
REAL*8DLL(25)/4*0.00,2*3.00,2*4.00,-10.00,14*0.00/
EQUIVALENCE
1(OCCUR(133),W1 ),(OCCUR(134),THETA1),(OCCUR(135),RN1 ),
2(OCCUR(136),R31 ),(OCCUR(137),LAMDA1),(OCCUR(138),LA1 ),
3(OCCUR(139),W2 ),(OCCUR(140),THETA2),(OCCUR(141),RN2 ),
4(OCCUR(142),R32 ),(OCCUR(143),LAMDA2),(OCCUR(144),LA2 ),
5(OCCUR(145),ZTURN)
EQUIVALENCE(OCCUR(222),ISP)
EQUIVALENCE
3(OCCUR(205),ZON ),(OCCUR(206),ZOFF ),(OCCUR(207),THO ),
4(OCCUR(208),TON ),(OCCUR(209),TOFF ),(OCCUR(210),MX )
REAL*8DUL(25)/2*10000.00,2*.600,2*168.00,40.00,40.00,17*0.00/
REAL*4M(25)/135,141,137,143,138,144,134,140,145,139,222,205,208,
1206,209,10*1/
-----
SUBROUTINE SCREEN TESTS THE VALUES OF UP TO TWENTY-FIVE
ELEMENTS OF THE OCCUR ARRAY TO DETERMINE WHETHER THESE VALUES
LIE WITHIN PRESCRIBED LIMITS. IF ANY VALUES LIE OUTSIDE THE
PRESCRIBED LIMITS, A QUANTITY E IS CALCULATED AS A MEASURE OF
ERROR.
-----
DO 2 I = 1,25
PENLTY(I)=0.00
2 A(I)=0.00
A(I)=1.00
A(3)=1.00
A(5)=1.00
A(7)=1.00
A(9)=1.00
DLL(12)=ZOFF
DLL(13)=T0
DLL(14)=ZST
DLL(15)=TON
DUL(9)=Z0
DUL(10)=W1
DUL(11)=10000.00
DUL(12)=Z0
DUL(13)=TOFF
DUL(14)=ZON
DUL(15)=TST
IF(ZTURN.LT.0.00)GJ TO 10
DO 8 I = 2,10,2
8 A(I)=1.00
-----

```

```

SCREE 1
SCREE 2
SCREE 3
SCREE 4
SCREE 5
SCREE 6
SCREE 7
SCREE 8
SCREE 9
SCREE 10
SCREE 11
SCREE 12
SCREE 13
SCREE 14
SCREE 15
SCREE 16
SCREE 17
SCREE 18
SCREE 19
SCREE 20
SCREE 21
SCREE 22
SCREE 23
SCREE 24
SCREE 25
SCREE 26
SCREE 27
SCREE 28
SCREE 29
SCREE 30
SCREE 31
SCREE 32
SCREE 33
SCREE 34
SCREE 35
SCREE 36
SCREE 37
SCREE 38
SCREE 39
SCREE 40
SCREE 41
SCREE 42
SCREE 43
SCREE 44
SCREE 45
SCREE 46
SCREE 47
SCREE 48
SCREE 49
SCREE 50
SCREE 51
SCREE 52
SCREE 53
SCREE 54
SCREE 55
SCREE 56
SCREE 57
SCREE 58
SCREE 59

```

C
C
C
C
C
C
C

10	IF(INTHRST.EQ.0)GO TO 15	SCREE 60
	A(11)=1.00	SCREE 61
	IF(INTHRST.EQ.2)GO TO 12	SCREE 62
	A(12)=1.00	SCREE 63
	A(14)=1.00	SCREE 64
	GO TO 15	SCREE 65
12	A(13) = 1.000	SCREE 66
	A(15)=1.00	SCREE 67
C		SCREE 68
C		SCREE 69
C		SCREE 70
15	IF(NGEOM=2) 18, 20, 40	SCREE 71
18	LAMDA1 = RN1/RB1	SCREE 72
	IF(ZTURN.GE.0.00)-LAMDA2 = RN2/RB2	SCREE 73
	GO TO 25	SCREE 74
20	RN1 = LAMCA1*RB1	SCREE 75
25	IF(ZTURN.LT.0.00)GO TO 30	SCREE 76
	KN2=LAMDA2*RB2	SCREE 77
	TR=0.0174532900*THETA2	SCREE 78
	ST2=DSIN(TR)	SCREE 79
	CT2=DCOS(TR)	SCREE 80
	LA2=(KN2*ST2-RN2&RB2*CT2)/ST2	SCREE 81
30	TR=0.0174532900*THETA1	SCREE 82
	ST1=DSIN(TR)	SCREE 83
	CT1=DCOS(TR)	SCREE 84
	LA1=(RN1*ST1-RN1&RB1*CT1)/ST1	SCREE 85
	GO TO 200	SCREE 86
40	LAMLA1 = RN1/RB1	SCREE 87
	TEM=LAI-RN1	SCREE 88
	TEST=TEM*TEM-RN1*RN1&RB1*RB1	SCREE 89
	IF(TEST.LT.0.000)GO TO 50	SCREE 90
	THETA1=180.00*ASINR((- (LAI-RN1)*RN1&RB1*DSQRT(TEST))/(TEST&RN1*RN1	SCREE 91
	1))/PI	SCREE 92
	GO TO 60	SCREE 93
50	THETA1=0.00	SCREE 94
60	IF(ZTURN.LT.0.00)GO TO 200	SCREE 95
	LAMDA2 = RN2/RB2	SCREE 96
	TEM=LA2-RN2	SCREE 97
	TEST=TEM*TEM-RN2*RN2&RB2*RB2	SCREE 98
	IF(TEST.LT.0.00)GO TO 70	SCREE 99
	THETA2=180.00*ASINR((- (LA2-RN2)*RN2&RB2*DSQRT(TEST))/(TEST&RN2*RN2	SCREE 100
	1))/PI	SCREE 101
	GO TO 200	SCREE 102
70	THETA2=0.00	SCREE 103
C		SCREE 104
C		SCREE 105
C		SCREE 106
200	E=0.00	SCREE 107
	DO 500 I=1,25	SCREE 108
	IZ=M(I)	SCREE 109
	IF(OCCUR(IZ).GT.DLL(I))GO TO 250	SCREE 110
	PENLTY(I)=A(I)*SMULT(I)*DABS(OCCUR(IZ)-DLL(I))*2	SCREE 111
	GO TO 500	SCREE 112
250	IF(OCCUR(IZ).LT.DUL(I))GO TO 500	SCREE 113
	PENLTY(I)=A(I)*SMULT(I)*DABS(OCCUR(IZ)-DUL(I))*2	SCREE 114
500	E=E+PENLTY(I)	SCREE 115
C		SCREE 116
C		SCREE 117
	IF(E.NE.0.00)GO TO 600	SCREE 118
	ISUC=0	SCREE 119

	GO TO 700	SCREEN120
C		SCREEN121
600	ISUC=-1	SCREEN122
	WRITE(6,450)PI,TO,ZO,ZST,TST,NGEOM,NTHRST,W1F,THET1F,RN1F,RB1F,	SCREEN123
	1LAMD1F,LA1F,W1,THETA1,RN1,RB1,LAMDA1,LA1,W2,THETA2,RN2,RB2,LAMDA2,	SCREEN124
	2LA2,ZTURN,ISP,ZON,ZOFF,THO,TON,TOFF,(PENLTY(I),I=1,25)	SCREEN125
	VAL=E&D	SCREEN126
C		SCREEN127
700	CUNTINUE	SCREEN128
C		SCREEN129
	RETURN	SCREEN130
C		SCREEN131
450	FORMAT(1H1 13X'PI',13X'TO',13X'ZO',12X'ZST',12X'TST',10X'NGEOM',	SCREEN132
	1 9X'NTHRST',12X'W1F'/1H 1PBE15.7,2115,E15.7/	SCREEN133
	21H0 9X'THET1F',11X'RN1F',11X'RB1F', 9X'LAMD1F',11X'LA1F',13X'W1',	SCREEN134
	3 9X'THETA1',12X'RN1'/1H 1PBE15.7/	SCREEN135
	41H0 12X'RB1', 9X'LAMDA1',12X'LA1',13X'W2', 9X'THETA2',12X'RN2',	SCREEN136
	512X'RN2', 9X'LAMDA2'/1H 1PBE15.7/	SCREEN137
	61H0 12X'LA2',10X'ZTURN',12X'ISP',12X'ZON',11X'ZOFF',12X'THO',	SCREEN138
	712X'TON',11X'TOFF'/1H 1PBE15.7/1H0PENLTY(I),I=1,25/	SCREEN139
	1(1H 1PBE15.7))	SCREEN140
	END	SCREEN141

CSR2490

SUBROUTINE SR2490	SR249 1
IMPLICIT REAL*8 (A-H,O-Z)	SR249 2
COMMON OCCUR,NOCCUR	SR249 3
COMMON /CPCUR/ PCCUR(11770)	SR249 4
COMMON /CICUR/ IOCCUR(320)	SR249 5
REAL*8 ISP,MW	SR249 6
DIMENSION A(514), B(21), AA(27), CDOWN(16), CHIGH(16), IOP(90)	SR249 7
DIMENSION NCOMDV(50), NUCCUR(30), OCCUR(4000)	SR249 8
DIMENSION NPV(160)	SR249 9
DIMENSION TAB1(50),TAB2(50),TABIX1(50),TABIX2(50)	SR249 10
EQUIVALENCE (OCCUR(301),A(1))	SR249 11
EQUIVALENCE (OCCUR(122),ALST)	SR249 12
EQUIVALENCE (OCCUR(823),B(1))	SR249 13
EQUIVALENCE (OCCUR(115),C)	SR249 14
EQUIVALENCE (OCCUR(19),CAPG)	SR249 15
EQUIVALENCE (OCCUR(3549),CDOWN(1))	SR249 16
EQUIVALENCE (OCCUR(3533),CHIGH(1))	SR249 17
EQUIVALENCE (OCCUR(167),DELIN)	SR249 18
EQUIVALENCE (OCCUR(189),FACTR1)	SR249 19
EQUIVALENCE (OCCUR(27),G)	SR249 20
EQUIVALENCE (OCCUR(28),GAMMA)	SR249 21
EQUIVALENCE (OCCUR(222),ISP)	SR249 22
EQUIVALENCE (OCCUR(4000),LP)	SR249 23
EQUIVALENCE (OCCUR(117),MW)	SR249 24
EQUIVALENCE (OCCUR(42),PI)	SR249 25
EQUIVALENCE (OCCUR(57),K)	SR249 26
EQUIVALENCE (OCCUR(63),RE)	SR249 27
EQUIVALENCE (OCCUR(116),SIG)	SR249 28
EQUIVALENCE (OCCUR(3033),TABIX1(1))	SR249 29
EQUIVALENCE (OCCUR(3083),TABIX2(1))	SR249 30
EQUIVALENCE (OCCUR(2933),TAB1(1))	SR249 31
EQUIVALENCE (OCCUR(2983),TAB2(1))	SR249 32
EQUIVALENCE (OCCUR(78),TECUN)	SR249 33
EQUIVALENCE (OCCUR(132),TINIT)	SR249 34
EQUIVALENCE (OCCUR(123),TST)	SR249 35
EQUIVALENCE (OCCUR(148),TWST)	SR249 36
EQUIVALENCE (OCCUR(149),TW1)	SR249 37
EQUIVALENCE (OCCUR(108),T42)	SR249 38
EQUIVALENCE (OCCUR(230),XLOW)	SR249 39
EQUIVALENCE (OCCUR(237),XUP)	SR249 40
EQUIVALENCE (OCCUR(240),X1LJW)	SR249 41
EQUIVALENCE (OCCUR(239),X1UP)	SR249 42
EQUIVALENCE (OCCUR(120),ZBAR)	SR249 43
EQUIVALENCE (OCCUR(93),ZETA)	SR249 44
EQUIVALENCE (OCCUR(118),ZPRI)	SR249 45
EQUIVALENCE (OCCUR(145),ZTURN)	SR249 46
EQUIVALENCE (NOCCUR(7),LUPT)	SR249 47
EQUIVALENCE (NOCCUR(20),MATLN1)	SR249 48
EQUIVALENCE (NOCCUR(21),MATLN2)	SR249 49
EQUIVALENCE (NOCCUR(26),MXTAB1)	SR249 50
EQUIVALENCE (NOCCUR(17),MXTAB2)	SR249 51
EQUIVALENCE (NOCCUR(15),NGEUM)	SR249 52
EQUIVALENCE (NOCCUR(14),NPRINT)	SR249 53
EQUIVALENCE (NOCCUR(22),NTHKST)	SR249 54
EQUIVALENCE (PCCUR(11571), AA)	SR249 55
EQUIVALENCE (IOCCUR(309), IMPLDT)	SR249 56
EQUIVALENCE (IOCCUR(00001), IOP)	SR249 57
EQUIVALENCE (IOCCUR(00301), IREF)	SR249 58
	SR249 59

EQUIVALENCE (IOCCUR(00302), LPLOT)	SR249 60
EQUIVALENCE (IOCCUR(00304), NCP)	SR249 61
EQUIVALENCE (IOCCUR(00091), NCOMDV)	SR249 62
EQUIVALENCE (IOCCUR(00305), NDECOY)	SR249 63
EQUIVALENCE (IOCCUR(00306), NDCVCH)	SR249 64
EQUIVALENCE (IOCCUR(000307), NPA)	SR249 65
EQUIVALENCE (IOCCUR(00141), NPV)	SR249 66
C	SR249 67
C	SR249 68
C	SR249 69
C	SR249 70
C	SR249 71
C	SR249 72
DO 11 I = 1, 4000	SR249 73
OCCUR(I) = 0.00	SR249 74
CONTINUE	SR249 75
C	SR249 76
DO 88 I = 1, 11770	SR249 77
PCCUR(I) = 0.000	SR249 78
CONTINUE	SR249 79
C	SR249 80
DO 44 I = 1, 30	SR249 81
NOCCUR(I) = 0	SR249 82
CONTINUE	SR249 83
C	SR249 84
DO 66 I = 1, 320	SR249 85
IOCCUR(I) = 0	SR249 86
CONTINUE	SR249 87
C	SR249 88
C	SR249 89
C	SR249 90
C	SR249 91
ALST=0.200	SR249 92
C=1.00	SR249 93
CAPG=32.2185200	SR249 94
DELIN=-2000.000	SR249 95
FACTR1=0.00237500**0.800	SR249 96
G=32.17400	SR249 97
GAMMA=1.400	SR249 98
IMPL0T = 0	SR249 99
IREF = 1	SR249 100
ISP=1.000	SR249 101
LOPT=1	SR249 102
LP = 1	SR249 103
LPLOT = 1	SR249 104
MATLN1=1	SR249 105
MATLN2=1	SR249 106
MW=28.900	SR249 107
MXTAB1=1	SR249 108
MXTAB2=1	SR249 109
NCP = 1	SR249 110
NDECOY = 1	SR249 111
NDCVCH = 1	SR249 112
NGEUM=1	SR249 113
NPA = 1	SR249 114
NPINT=1	SR249 115
PI = 3.14159265358979300	SR249 116
R=53.500	SR249 117
RE=2.0902290607	SR249 118
SIG=3.500	SR249 119
TABIX1(1)=1.000	SR249 120

	TAB1A2(1)=1.000	SK249120
	TAB1(1)=1.000	SR249121
	TAB2(1)=1.000	SK249122
	TECON=2.000	SK249123
	TINIT=500.000	SK249124
	TST=100.000	SK249125
	TWST=580.000	SR249126
	TW1=1200.000	SR249127
	TW2=1200.000	SK249128
	XL0W=4.000	SK249129
	XUP=6.000	SK249130
	XLOW=0.200	SK249131
	XUP=0.400	SK249132
	ZBAR=-10000.000	SK249133
	ZETA=0.900	SK249134
	ZPK1=10000.000	SK249135
	ZTUKN = -1.00	SK249136
C		SK249137
	DO 22 I = 1, 160	SK249138
	NPV(I) = 1	SK249139
22	CONTINUE	SK249140
C		SK249141
	DO 33 I = 1, 50	SK249142
	NCOMDV(I) = 133	SK249143
33	CONTINUE	SR249144
C		SK249145
	DO 99 I = 1, 90	SK249146
	IOP(I) = 1	SK249147
99	CONTINUE	SK249148
C		SK249149
	DO 4 J=1,16	SK249150
	CHIGH(J)=1.00-4	SK249151
4	CDOWN(J)=1.00-5	SK249152
C		SK249153
	DO 77 I = 3, 27, 3	SK249154
	AA(I) = 1.00	SK249155
77	CONTINUE	SK249156
C		SK249157
	CALL ZPRS(A,8)	SK249158
C		SK249159
	RETURN	SK249160
C		SK249161
	END	SR249162

CSTUFF	SUBROUTINE STUFF	STUFF 1
	IMPLICIT REAL*8 (A-H,O-Z)	STUFF 2
	COMMON/BLKD/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)	STUFF 3
	1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FO,E, P,TJ,RS,SL, Z,D,A,EL,	STUFF 4
	2DELTA,FAC,C(40,10)	STUFF 5
	COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC	STUFF 6
C		STUFF 7
C		STUFF 8
C	SUBROUTINE STUFF, PART OF THE DAVIDON MINIMIZATION METHOD,	STUFF 9
C	TESTS HOW WELL THE FUNCTION HAS BEEN MINIMIZED AND HOW WELL	STUFF 10
C	THE MATRIX H APPROXIMATES THE INVERSE OF THE MATRIX OF SECOND	STUFF 11
C	PARTIAL DERIVATIVES AT THE MINIMUM.	STUFF 12
C		STUFF 13
C		STUFF 14
C	STUFF	STUFF 15
C	600 K=K-1	STUFF 16
	601 IF (K)617,602,602	STUFF 17
	602 MS=MS&1	STUFF 18
	620 WRITE (6,1)MS,GS	STUFF 19
	603 DO 604 I=1,N	STUFF 20
C		STUFF 21
C		STUFF 22
C	604 T(I)=RANDOM(DUMMY)-0.5	STUFF 23
C		STUFF 24
C	605 CALL MATMP (N,N,H,T,S)	STUFF 25
	606 M=1	STUFF 26
	607 CALL MATMP (M,N,S,T,TP1)	STUFF 27
	608 TP1 = DSQRT(TP1)	STUFF 28
	609 FL=P/TP1	STUFF 29
	610 DO 611 I=1,N	STUFF 30
	611 X(I)=X(I)&EL*S(I)	STUFF 31
	612 M1=2	STUFF 32
	614 L=1	STUFF 33
	616 RETURN	STUFF 34
	617 L=2	STUFF 35
	618 MS=0.	STUFF 36
	619 GO TO 616	STUFF 37
	1 FORMAT (13HORANDOM,STEP 14,5H GS=1PE14.5)	STUFF 38
	END	STUFF 39
		STUFF 40

CTABLE	IMPLICIT REAL*8 (A-M,O-Z)	TABLE	1
	FUNCTION TABLE(X,XTAB,YTAB,KMAX,L)	TABLE	2
	DIMENSION XTAB(2),YTAB(2)	TABLE	3
C		TABLE	4
C	FUNCTION TABLE PERFORMS A ONE DIMENSIONAL TABLE LOOK UP	TABLE	5
C	USING LINEAR INTERPOLATION.	TABLE	6
C		TABLE	7
	IF(KMAX.EQ.1)GO TO 6	TABLE	8
	IF(L.GT.0)GO TO 2	TABLE	9
	IGD=1	TABLE	10
	IF(XTAB(1).GT.XTAB(2))IGD=2	TABLE	11
C		TABLE	12
	DO 1 K=2,KMAX	TABLE	13
	GO TO(3,4),IGD	TABLE	14
	3 IF(X.GT.XTAB(K))GO TO 1	TABLE	15
	GO TO 5	TABLE	16
	4 IF(X.LT.XTAB(K))GO TO 1	TABLE	17
	5 L=K	TABLE	18
	GO TO 2	TABLE	19
	1 CONTINUE	TABLE	20
C		TABLE	21
	L=KMAX	TABLE	22
	2 ANS=YTAB(L-1)+(X-XTAB(L-1))*(YTAB(L)-YTAB(L-1))/	TABLE	23
	(XTAB(L)-XTAB(L-1))	TABLE	24
	TABLE=ANS	TABLE	25
	RETURN	TABLE	26
C		TABLE	27
	6 TABLE=YTAB(1)	TABLE	28
	RETURN	TABLE	29
	END	TABLE	30
		TABLE	31

NOT REPRODUCIBLE

C	TEQUAT	TEQUA 1
C	IMPLICIT REAL*8 (A-H,O-Z)	TEQUA 2
C	SUBROUTINE TEQUAT(DERIV)	TEQUA 3
C	REAL*8 ISP,M	TEQUA 4
C		TEQUA 5
C		TEQUA 6
C	DIMENSION DERIV(16),CMAT(3,3),TTMAT(3),TBMAT(3),FBMAT(3),FTMAT(3)	TEQUA 7
C	DIMENSION OCCUR(1000),NOCCUR(30)	TEQUA 8
C		TEQUA 9
C	EQUIVALENCE	TEQUA 10
C	1(OCCUR(100)),ARFF),(OCCUR(1016),CD),(OCCUR(1019),CAPG),	TEQUA 11
C	2(OCCUR(1026),GAMP),(OCCUR(1037),M),(OCCUR(1051),OD),	TEQUA 12
C	3(OCCUR(1063),RF),(OCCUR(1072),V),(OCCUR(1091),Z)	TEQUA 13
C	EQUIVALENCE	TEQUA 14
C	1(OCCUR(1003),ALPRM),(OCCUR(1044),PHI),(OCCUR(1045),PSI),	TEQUA 15
C	2(OCCUR(1071),THEALP),(OCCUR(1200),PSIALP),(OCCUR(1203),CN),	TEQUA 16
C	3(OCCUR(1213),THINF),(OCCUR(1222),ISP),	TEQUA 17
C	4(OCCUR(13565),TTMAT(1)),(OCCUR(13643),TBMAT(1))	TEQUA 18
C		TEQUA 19
C	COMMON OCCUR,NOCCUR	TEQUA 20
C		TEQUA 21
C		TEQUA 22
C		TEQUA 23
C	SUBROUTINE TEQUAT CALCULATES THE DERIVATIVES OF THE PARTICLE	TEQUA 24
C	TRAJECTORY AND THRUSTING PARAMETERS.	TEQUA 25
C	THE FOLLOWING DERIVATIVES ARE CALCULATED IN THIS SUBROUTINE	TEQUA 26
C	DERIV(1)=V DOT	TEQUA 27
C	DERIV(2)=GAMP DOT	TEQUA 28
C	DERIV(3)=D(TIME)/D(ALTITUDE)	TEQUA 29
C	DERIV(4)=XR DOT	TEQUA 30
C	DERIV(14)=YR DOT	TEQUA 31
C	DERIV(15)=PSIALP DOT	TEQUA 32
C	DERIV(16)=KTH DOT	TEQUA 33
C		TEQUA 34
C		TEQUA 35
C	TRIGONOMETRIC FUNCTIONS OF THE EULPR ANGLES.	TEQUA 36
C	CTHEL=COS(THEALP)	TEQUA 37
C	STHEL=SIN(THEALP)	TEQUA 38
C	CPSI=COS(PSI)	TEQUA 39
C	SPSI=SIN(PSI)	TEQUA 40
C	CPHI=COS(PHI)	TEQUA 41
C	SPHI=SIN(PHI)	TEQUA 42
C		TEQUA 43
C	COMPONENTS OF THE CONVERSION MATRIX WHICH TRANSFORMS FROM A	TEQUA 44
C	BODY ORIENTED TO A TRAJECTORY ORIENTED COORDINATE SYSTEM.	TEQUA 45
C	CMAT(1,1)=CTHEL*CPSI	TEQUA 46
C	CMAT(2,1)=CTHEL*SPSI	TEQUA 47
C	CMAT(3,1)=-STHEL	TEQUA 48
C	CMAT(1,2)=CPSI*STHEL*SPHI-SPSI*CPHI	TEQUA 49
C	CMAT(2,2)=SPSI*STHEL*SPHI+CPSI*CPHI	TEQUA 50
C	CMAT(3,2)=CTHEL*SPHI	TEQUA 51
C	CMAT(1,3)=CPSI*STHEL*CPHI+SPSI*SPHI	TEQUA 52
C	CMAT(2,3)=SPSI*STHEL*CPHI-CPSI*SPHI	TEQUA 53
C	CMAT(3,3)=CTHEL*CPHI	TEQUA 54
C		TEQUA 55
C	RESOLVING THE THRUST VECTOR INTO TRAJECTORY COORDINATES.	TEQUA 56
C	CALL MATMPY(TTMAT,CMAT,TBMAT)	TEQUA 57
C	XROOT=V*CMAT(1,1)	TEQUA 58
C	YROOT=V*CMAT(1,2)	TEQUA 59

```

ZROOT=V*CMAT(1,2) TEQUA 60
C TEQUA 61
NORMAL FORCE CALCULATION. TEQUA 62
FN=CN*QD*AREF TEQUA 63
AXIAL FORCE COEFFICIENT. TEQUA 64
CX=(1-CCDCN*SIN(ALPRM))/COS(ALPRM) TEQUA 65
C TEQUA 66
CALCULATION OF BODY FORCES. TEQUA 67
FMAT(1)=QD*AREF*CX TEQUA 68
ROOT=SQRT(YROOT**2&ZROOT**2) TEQUA 69
IF(ROOT,LT,1.0D-10)GO TO 1 TEQUA 70
FMAT(2)=-FN*YROOT/ROOT TEQUA 71
FMAT(3)=-FN*ZROOT/ROOT TEQUA 72
GO TO 2 TEQUA 73
1 FMAT(2)=0.000 TEQUA 74
FMAT(3)=0.000 TEQUA 75
C RESOLVING THE BODY FORCES INTO TRAJECTORY COORDINATE SYSTEM. TEQUA 76
2 CALL MATMPY(FMAT,CMAT,FMAT) TEQUA 77
C TEQUA 78
CALCULATION OF DERIVATIVES. TEQUA 79
TEM=RE/(P*GZ) TEQUA 80
TEM2=CARG*TEM*TEM TEQUA 81
SING=SIN(GAME) TEQUA 82
COSG=COS(GAME) TEQUA 83
DERIV(1)=-QD*CD*AREF/M-TEM2*SING&TTMAT(1)/M TEQUA 84
DERIV(2)=COSG*(V*V*TEM/RE-TEM2)/V-(FTMAT(3)&TTMAT(3))/(M*V) TEQUA 85
DERIV(3)=V*SING TEQUA 86
DERIV(4)=COS(PSIALP)*V*COSG*TEM TEQUA 87
DERIV(4A)=SIN(PSIALP)*V*COSG*TEM TEQUA 88
DERIV(1F)=- (FTMAT(2)&TTMAT(2))/(M*V*CCSG) TEQUA 89
DERIV(1F)=THINF/ISP TEQUA 90
DERIV(3)=1.000/DERIV(3) TEQUA 91
RETURN TEQUA 92
END TEQUA 93

```

C	TIMERS	TIMER 1
C	IMPLICIT REAL*8(A-H,O-Z)	TIMER 2
C	SUBROUTINE TIMERS(IN, TABLE)	TIMER 3
C	DIMENSION TABLE(2)	TIMER 4
C		TIMER 5
C		TIMER 6
C	SUBROUTINE TIMERS IS A DUMMY SUBROUTINE.	TIMER 7
C		TIMER 8
C		TIMER 9
C	RETURN	TIMER 10
C	END	TIMER 11

CTOMALO	TOMAL 1
IMPLICIT REAL*8 (A-H,O-Z)	TOMAL 2
SUBROUTINE TOMALO	TOMAL 3
REAL*8 LAMDA,LA,MDOT,MC,MT,MN	TOMAL 4
	TOMAL 5
DIMENSION OCCUR(4000),NOCCUR(30),MDOT(4,8),XLA(8)	TOMAL 6
DIMENSION MC(6)	TOMAL 7
	TOMAL 8
EQUIVALENCE	TOMAL 9
1(OCCUR(008),COST),(OCCUR(032),LAMDA),(OCCUR(033),LA),	TOMAL 10
2(OCCUR(042),PI),(OCCUR(052),RN),(OCCUR(064),SINT),	TOMAL 11
3(OCCUR(070),TANT),(OCCUR(086),WDOCT),(OCCUR(091),Z),	TOMAL 12
4(OCCUR(092),ZTR),(OCCUR(0915),XLA(1)),(OCCUR(2708),MCOT(1,1))	TOMAL 13
	TOMAL 14
COMMON OCCUR,NOCCUR	TOMAL 15
	TOMAL 16
	TOMAL 17
	TOMAL 18
	TOMAL 19
	TOMAL 20
SUBROUTINE TOMALO INTEGRATES THE MASS LOSS RATE ALONG THE	TOMAL 21
BODY TO OBTAIN THE RATE OF CHANGE IN WEIGHT DUE TO ABLATION.	TOMAL 22
	TOMAL 23
INTEGRATION ALONG SHARP CONE FOR LAMINAR FLOW.	TOMAL 24
IF(LAMDA.GT.1.00-3160 TO 2	TOMAL 25
IF(Z.GE.ZTR)MT=2.071800*SQRT(2.000)*TANT*MDOT(2,8)*LA**2/COST	TOMAL 26
	TOMAL 27
INTEGRATION ALONG SHARP CONE FOR TURBULENT FLOW.	TOMAL 28
IF(Z.LT.ZTR)MT=3.0400*((2.000)**0.200)*TANT*MDOT(3,8)*LA**2/COST	TOMAL 29
GO TO 3	TOMAL 30
	TOMAL 31
TEM=PI*LA**2*LA**2/(COST*COST)	TOMAL 32
TEM1=RN*(1.00-SINT)/LA	TOMAL 33
TEM2=0.500*SINT	TOMAL 34
TEM3=(1.74600-SINT)/.23400	TOMAL 35
	TOMAL 36
BLUNT CONE LAMINAR MASS LOSS INTEGRATION.	TOMAL 37
IF(Z.LT.ZTR)GO TO 4	TOMAL 38
INTEGRATION AROUND THE NOSECAP.	TOMAL 39
MN=PI*RN*(1.00-SINT)*(MDOT(1,1)&MDOT(2,1))*RN	TOMAL 40
INTEGRATION ALONG CONICAL FRUSTUM USING TRAPEZOIDAL RULE.	TOMAL 41
DO 5 I=1,6	TOMAL 42
MC(I)=TEM*(XLA(I&1)-XLA(I))*((TEM1&TEM2*(XLA(I&1)&XLA(I)))	TOMAL 43
MC(I)=MC(I)*(MDOT(2,I)&MDOT(2,I&1))	TOMAL 44
GO TO 7	TOMAL 45
	TOMAL 46
	TOMAL 47
BLUNT CONE TURBULENT MASS LOSS INTEGRATION.	TOMAL 48
INTEGRATION AROUND THE NOSECAP.	TOMAL 49
MN=0.725100*RN*RN*(MDOT(1,1)&MDOT(4,1)&(MDOT(4,1)&MDOT(3,1)))	TOMAL 50
I*TEM3)	TOMAL 51
INTEGRATION ALONG CONICAL FRUSTUM USING TRAPEZOIDAL RULE.	TOMAL 52
DO 5 I=1,6	TOMAL 53
MC(I)=TEM*(XLA(I&1)-XLA(I))*((TEM1&TEM2*(XLA(I&1)&XLA(I)))	TOMAL 54
MC(I)=MC(I)*(MDOT(3,I)&MDOT(3,I&1))	TOMAL 55
	TOMAL 56
	TOMAL 57
SUMMATION OF CONTRIBUTIONS FROM NOSECAP AND CONICAL FRUSTUM.	TOMAL 58
MT=MN	TOMAL 59

DO B (=),A
A MT=MTMC(1)
WDDY= MY
RETURN
END

TOTAL 60
TOTAL 61
TOTAL 62
TOTAL 63
TOTAL 64

NOT REPRODUCIBLE

CVIXEN	IMPLICIT REAL*8 (A-H,U-Z)	VIXFN	1
	SUBROUTINE VIXEN	VIXFN	2
		VIXFN	3
	REAL*4 TIXM,TIXT,TIXZ,T1,T2,T3,T4,	VIXFN	4
	T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16,T17,T18,T19,T20,T21,T22	VIXFN	5
	REAL*4 T23,T24,T25,T26,T27,T28	VIXFN	6
	REAL*8 LAMDA,LA,MINE,MOUT,LAMSTO	VIXFN	7
	COMMON /COCUR/ COCUR(11770)	VIXFN	8
	COMMON /CICUR/ ICOCUR(1320)	VIXFN	9
	COMMON /IXCOM/ XCOM(200), ICUM(200)	VIXFN	10
	COMMON COCUR,NOCCUR	VIXFN	11
		VIXFN	12
		VIXFN	13
	DIMENSION A(514),COFINE(2,2,2),QDOT(4,8),MDET(4,8),PEPSU(8),	VIXFN	14
1	COCUR(11770),NOCCUR(1320),CDJNV(16),TTMAT(3),TITLE(2),	VIXFN	15
2	ALMAX(200),ALMIN(200),TAMAX(200),TAMIN(200),	VIXFN	16
3	ZMAX(200),ZMIN(200),FMAX(200),FMIN(200)	VIXFN	17
	DIMENSION DVALUE(16),DERIV(16),LPRNO(16),DNRAO(16),CHIGH(16)	VIXFN	18
	DIMENSION QINT(4),QLAST(4)	VIXFN	19
	DIMENSION NPLOT(4)	VIXFN	20
	DIMENSION WTZ(75),WTMINE(75),WTRINE(75),WTPICT(75),TRAJZ(75),	VIXFN	21
1	TRAJZ(75),TRAJZV(75),TRAJZV(75),TRAJRN(75),TRJALP(75),	VIXFN	22
2	ALPTAB(75)	VIXFN	23
	DIMENSION TIXM(10),TIXT(10),TIXZ(10),T1(10),T2(10),	VIXFN	24
	T3(10),T4(10),T5(10),T6(10),T7(10),T8(10),T9(10),T10(10),T11(10),	VIXFN	25
	T12(10),T13(10),T14(10),T15(10),T16(10),T17(10),T18(10),T19(10),T19(10),	VIXFN	26
	T20(10),T21(10),T22(10),	VIXFN	27
	3ZPLT(160),TPLUT(160),PLMINE(160),COPLT(160),COFPLT(160),	VIXFN	28
4	COFPLT(160),COIPLT(160),WTOIPL(160),METAPL(160),VPLCT(160),	VIXFN	29
5	WGPLT(160),OPLUT(160),PSPLOT(160),SRTPL(160),COLLPL(160),	VIXFN	30
6	3D4PL(160),PEIPL(160),PE7PL(160),PE8PL(160),CO7PLT(160),	VIXFN	31
7	CO8PLT(160),W2PLT(160),COPLT(160)	VIXFN	32
	DIMENSION WR1P(160),WR2P(160),WR3P(160)	VIXFN	33
	DIMENSION WL1P(160),WL2P(160),WL3P(160)	VIXFN	34
	DIMENSION T23(10),T24(10),T25(10),T26(10),T27(10),T28(10)	VIXFN	35
	DIMENSION IOP(50)	VIXFN	36
		VIXFN	37
	EQUIVALENCE	VIXFN	38
1	(COCUR(011),APFF),(COCUR(002),ALPHA),(COCUR(003),ALPRMI),	VIXFN	39
2	(COCUR(010),CAPL),(COCUR(016),CD),(COCUR(070),CMC),	VIXFN	40
3	(COCUR(026),GAMF),(COCUR(027),G)	VIXFN	41
	EQUIVALENCE	VIXFN	42
1	(COCUR(029),HSR TC),(COCUR(032),LAMDA),(COCUR(033),LA),	VIXFN	43
2	(COCUR(035),MINE),(COCUR(042),PI),(COCUR(043),P),	VIXFN	44
3	(COCUR(044),PHI),(COCUR(045),PSI),(COCUR(047),PS),	VIXFN	45
4	(COCUR(050),Q),(COCUR(051),QO),(COCUR(052),RN)	VIXFN	46
	EQUIVALENCE	VIXFN	47
1	(COCUR(062),REYL),(COCUR(065),SMR),	VIXFN	48
2	(COCUR(069),THE TAU),(COCUR(071),THE ALP),(COCUR(075),T),	VIXFN	49
3	(COCUR(076),THE TA),(COCUR(080),TIMER),(COCUR(091),SMF)	VIXFN	50
	EQUIVALENCE	VIXFN	51
1	(COCUR(092),V),(COCUR(097),XR),	VIXFN	52
2	(COCUR(093),XBAR),(COCUR(091),Z),(COCUR(092),ZTK),	VIXFN	53
3	(COCUR(095),GRATE),(COCUR(096),ALPENV),(COCUR(097),BELM),	VIXFN	54
4	(COCUR(098),COP),(COCUR(099),COB),(COCUR(100),COI),	VIXFN	55
5	(COCUR(101),COPD),(COCUR(102),TU),(COCUR(105),GAPF),	VIXFN	56
6	(COCUR(106),VO),(COCUR(107),XRY),(COCUR(108),ZC)	VIXFN	57
	EQUIVALENCE	VIXFN	58
1	(COCUR(109),PO),(COCUR(110),QO),(COCUR(111),SMFO),	VIXFN	59



NOT REPRODUCIBLE

21(OCCUR(112),PHI0),(OCCUR(113),THEALO),(OCCUR(114),PSIG),	VIXFN 60
3(OCCUR(118),ZPR1),(OCCUR(119),ZPR2),(OCCUR(120),ZPAR),	VIXFN 61
4(OCCUR(121),ZST),(OCCUR(122),ALST),(OCCUR(123),TST),	VIXFN 62
5(OCCUR(126),XRARI),	VIXFN 63
6(OCCUR(127),DATE),(OCCUR(128),CASE),(OCCUR(129),EMC),	VIXFN 64
7(OCCUR(130),ALWIG?), (OCCUR(131),ALBARD), (OCCUR(133),W1),	VIXFN 65
8(OCCUR(137),DELIN),	VIXFN 66
EQUIVALENCE	VIXFN 67
1(OCCUR(199),YK),(OCCUR(200),PSTALP),(OCCUR(201),TH),	VIXFN 69
2(OCCUR(202),CM),(OCCUR(203),CN),(OCCUR(226),DFLW2),	VIXFN 69
EQUIVALENCE	VIXFN 70
1(OCCUR(227),DFLW3),(OCCUR(228),WOTAL),(OCCUR(230),RETAP),	VIXFN 71
2(OCCUR(231),BCTA),(OCCUR(232),VGGDT),(OCCUR(233),DOW),	VIXFN 72
3(OCCUR(234),DELOP),(OCCUR(235),DEDFP),(OCCUR(236),DEFFC),	VIXFN 73
4(OCCUR(237),XUP),(OCCUR(244),REYINF),	VIXFN 74
5(OCCUR(245),PTOTAL),(OCCUR(247),UPBNDZ),(OCCUR(248),DNBNDZ)	VIXFN 75
EQUIVALENCE	VIXFN 76
1(OCCUR(244), A(1)),(OCCUR(1344), ALMAX(1)),	VIXFN 77
2(OCCUR(1644), ALMIN(1)),(OCCUR(1444), TA4AX(1)),	VIXFN 78
3(OCCUR(1644), TMIN(1)),(OCCUR(1344), ZMAX(1)),	VIXFN 79
4(OCCUR(244), ZMIN(1)),(OCCUR(244), FMAX(1)),	VIXFN 80
5(OCCUR(2444), FMIN(1))	VIXFN 81
EQUIVALENCE	VIXFN 82
1(OCCUR(1194),WTZ(1)),(OCCUR(1119),WTMINF(1)),	VIXFN 83
2(OCCUR(1194),WTRINF(1)),(OCCUR(1269),WTPTCT(1)),	VIXFN 84
3(OCCUR(1344),TRAJT(1)),(OCCUR(1419),TRAJ7(1)),	VIXFN 85
4(OCCUR(1494),TRAJV(1)),(OCCUR(1569),TRAJW(1)),	VIXFN 86
5(OCCUR(1644),TRAJRN(1)),(OCCUR(1719),TRJALP(1))	VIXFN 87
EQUIVALENCE	VIXFN 88
1(OCCUR(2676),QDGT(1,1)),(OCCUR(2709),MDGT(1,1)),	VIXFN 89
2(OCCUR(2793),CDFINF(1,1,1)),(OCCUR(2801),PEPSH(1)),	VIXFN 90
3(OCCUR(3233),HTAB(1))	VIXFN 91
EQUIVALENCE	VIXFN 92
1(OCCUR(3533),CHIGH(1)),(OCCUR(3549),CDJWA(1)),	VIXFN 93
2(OCCUR(3565),TTHAT(1)),(OCCUR(3646),ALPTAB(1))	VIXFN 94
EQUIVALENCE	VIXFN 95
1(OCCUR(101),JJHOLD),(NOCCUR(03),MOPT),	VIXFN 96
2(OCCUR(06),MAXVAL),(NOCCUR(07),LOPT),(NOCCUR(10),MHEAT),	VIXFN 97
3(OCCUR(11),IKMAX),(NOCCUR(12),IKMIN),	VIXFN 98
4(OCCUR(14),NPRINT),(NOCCUR(24),NPLGT(1)),(NOCCUR(29),ITAPE),	VIXFN 99
5(OCCUR(30),INALPH)	VIXFN 100
EQUIVALENCE (NOCCUR(0000),TATMS)	VIXFN 101
EQUIVALENCE(OCCUR(4000),LPI)	VIXFN 102
EQUIVALENCE (OCCUR(00001), IOP)	VIXFN 103
EQUIVALENCE (OCCUR(00314), IORL)	VIXFN 104
EQUIVALENCE (OCCUR(00302), LPLCT)	VIXFN 105
EQUIVALENCE (OCCUR(00641), BTAPL)	VIXFN 106
EQUIVALENCE (OCCUR(00001), TPLGT)	VIXFN 107
EQUIVALENCE (OCCUR(00481), WGPLT)	VIXFN 108
EQUIVALENCE (OCCUR(00321), VPLGT)	VIXFN 109
EQUIVALENCE (OCCUR(012P1), WLIP)	VIXFN 110
EQUIVALENCE (OCCUR(01441), WL2P)	VIXFN 111
EQUIVALENCE (OCCUR(01641), WL3P)	VIXFN 112
EQUIVALENCE (OCCUR(00801), WR1P)	VIXFN 113
EQUIVALENCE (OCCUR(00561), WR2P)	VIXFN 114
EQUIVALENCE (OCCUR(01121), WR3P)	VIXFN 115
EQUIVALENCE (OCCUR(01161), ZPLGT)	VIXFN 116
	VIXFN 117
	VIXFN 118
	VIXFN 119

NOT REPRODUCIBLE

C			VIXFN120
C			VIXFN121
C			VIXFN122
C			VIXFN123
C			VIXFN124
C			VIXFN125
C			VIXFN126
C			VIXFN127
C			VIXFN128
C			VIXFN129
C			VIXFN130
C			VIXFN131
C			VIXFN132
C			VIXFN133
C			VIXFN134
C			VIXFN135
C			VIXFN136
C			VIXFN137
C			VIXFN138
C			VIXFN139
C			VIXFN140
C			VIXFN141
C			VIXFN142
C			VIXFN143
C			VIXFN144
C			VIXFN145
C			VIXFN146
C			VIXFN147
C			VIXFN148
C			VIXFN149
C			VIXFN150
C			VIXFN151
C			VIXFN152
C			VIXFN153
C			VIXFN154
C			VIXFN155
C			VIXFN156
C			VIXFN157
C			VIXFN158
C			VIXFN159
C			VIXFN160
C			VIXFN161
C			VIXFN162
C			VIXFN163
C			VIXFN164
C			VIXFN165
C			VIXFN166
C			VIXFN167
C			VIXFN168
C			VIXFN169
C			VIXFN170
C			VIXFN171
C			VIXFN172
C			VIXFN173
C			VIXFN174
C			VIXFN175
C			VIXFN176
C			VIXFN177
C			VIXFN178
C			VIXFN179

	SUBROUTINE VIXEN PERFORMS THE FOLLOWING FUNCTIONS -			
	SETS INITIAL VALUES OF CERTAIN QUANTITIES.			
	CALLS IN SUBROUTINE CHNTHL, WHICH ASSIGNS APPROPRIATE GEOMETRIC			
	AND MATERIAL PROPERTIES.			
	ASSIGNS THE ACCURACY LIMITS FOR INTEGRATION.			
	CALLS IN THE INTEGRATION SUBROUTINE, ADM4RK, FOR THOSE OPTIONS			
	WHERE A TRAJECTORY IS BEING CALCULATED, OR SETS PROGRAM			
	VARIABLES EQUAL TO TABULAR INPUT VALUES OF THE WIND TUNNEL OR			
	INPUT TRAJECTORY OPTIONS, AS WELL AS CALLING IN THE SUBROUTINES			
	WHICH PERFORM CALCULATIONS NECESSARY TO OBTAIN DRAG VALUES			
	WHEN NO INTEGRATION IS DONE.			
	CALCULATES THE MAXIMUMS AND MINIMUMS IN ANGLE OF ATTACK WHEN			
	THE THREE DEGREE OF ROTATIONAL FREEDOM OPTION IS USED.			
	CONTROLS THE PRINTOUT.			
	CALLS IN THE TAPE AND THE PLOTTING SUBROUTINES.			

	TITLES FOR THE PLOTTING OPTIONS			
	DATA TITLE(1)/4HTUPR/			
	DATA TITLE(2)/4H LAM/			
	DATA T1X1/4CH	TIME IN SECONDS	/	
	DATA T1X2/4CH	ALTITUDE IN FEET	/	
	DATA T1X4/4CH	MINF	/	
	DATA T1/4CH	CD TOTAL	/	
	DATA T2/4CH	CD8 & CD9	/	
	DATA T2/4CH	CDFINE(BLUNT WITH BLOWING)	/	
	DATA T4/4CH	CDI	/	
	DATA T5/4CH	WTOTAL/WI	/	
	DATA T6/4CH	BETA	/	
	DATA T7/4CH	VELOCITY(FT/SEC)	/	
	DATA T8/4CH	MINF	/	
	DATA T9/4CH	QDOT/G	/	
	DATA T10/4CH	ALTTIME(FT)	/	
	DATA T11/4CH	DYNAMIC PRESSURE(LB/FT2)	/	
	DATA T12/4CH	PS(ATMOS)	/	
	DATA T13/4CH	HSRTC	/	
	DATA T14/4CH	QDOT(STAG)(RTU/FT2-SEC)	/	
	DATA T15/4CH	QDOT(SONIC)(RTU/FT2-SEC)	/	
	DATA T16/4CH	QDOT(7)(RTU/FT2-SEC)	/	
	DATA T17/4CH	QDOT(8)(RTU/FT2-SEC)	/	
	DATA T18/4CH	PEPSH(TANG. PT)	/	
	DATA T19/4CH	PEPS3(7)	/	
	DATA T20/4CH	PEPS3(8)	/	
	DATA T21/4CH	ALPHA PRIME (IN DEGREES)	/	
	DATA T22/4CH	CCCUR(NPLCT5)	/	
	DATA T23/4CH	WAKE L1	/	
	DATA T24/4CH	WAKE L2	/	
	DATA T25/4CH	WAKE L3	/	
	DATA T26/4CH	WAKE R1	/	
	DATA T27/4CH	WAKE R2	/	
	DATA T28/4CH	WAKE R3	/	

	SETTING INITIAL VALUES			
	WAKE=0			
	V=V0			
	(WAKE=GAMFU			
	TIME=T0)			

NOT REPRODUCIBLE

Z=Z0	V IX FN 180
XR=XR0	V IX FN 181
P=PO	V IX FN 182
U=00	V IX FN 183
SNP=SMRO	V IX FN 184
PSI=PSIC	V IX FN 185
THEALP=THEALO	V IX FN 186
PHI=PHIO	V IX FN 187
YR=U.CDO	V IX FN 188
PSIALP=0.000	V IX FN 189
ZUSF=Z	V IX FN 190
CVALUE(1)=V	V IX FN 191
CVALUE(2)=GAMF	V IX FN 192
CVALUE(3)=TIME	V IX FN 193
CVALUE(4)=XR	V IX FN 194
CVALUE(8)=PSI	V IX FN 195
CVALUE(9)=THEALP	V IX FN 196
CVALUE(10)=PHI	V IX FN 197
CVALUE(11)=0	V IX FN 198
CVALUE(12)=SMR	V IX FN 199
CVALUE(13)=P	V IX FN 200
CVALUE(14)=YR	V IX FN 201
CVALUE(15)=PSIALP	V IX FN 202
HETA=0.00	V IX FN 203
HETAP=0.00	V IX FN 204
DELDET=0.00	V IX FN 205
T=0.00	V IX FN 206
SMF=C.CDO	V IX FN 207
ALPRIM=C.CDO	V IX FN 208
ALPHA=C.CDO	V IX FN 209
ALPHIV=C.CDO	V IX FN 210
IALP=1	V IX FN 211
IKMAX=0	V IX FN 212
IKMIN=0	V IX FN 213
PRINTZ=C.CDO	V IX FN 214
ZSTO=C.CDO	V IX FN 215
DELPR1=ZPR1	V IX FN 216
FREQ=-DELPR1	V IX FN 217
DEL=-ZPR1/1.CDO	V IX FN 218
LPLOT=0	V IX FN 219
NTAPE=0	V IX FN 220
C	V IX FN 221
C SELECTION OF DELTA OF INTEGRATION	V IX FN 222
C IF(DABS(DELIN),.LT,DABS(DEL))DEL=DELIN	V IX FN 223
C	V IX FN 224
C SUBROUTINE CHNTOL ASSIGNS THE APPROPRIATE GEOMETRIC PARA-	V IX FN 225
C METERS AND MATERIAL PROPERTIES AT ENTRY AND WHEN CONFIGURATION	V IX FN 226
C CHANGES AT ALTITUDE ZTURN.	V IX FN 227
C LCHANGE=1	V IX FN 228
60 CALL CHNTOL(IVALUF,ZTURN,LCHANGE)	V IX FN 229
L=1	V IX FN 230
IF(JJHOLD.NF.L=1)GO TO 50	V IX FN 231
LP=6	V IX FN 232
WRITE(6,1005)	V IX FN 233
GO TO 100	V IX FN 234
50 IF(L/USF.GT.ZTURNX)GO TO 61	V IX FN 235
LCHANGE=2	V IX FN 236
WRITE(6,1005)	V IX FN 237
1005 FORMAT(1H0,20X,'CONFIGURATION IS CHANGING DISCONTINUOUSLY AT Z=',	V IX FN 238
1E15.6)	V IX FN 239

NOT REPRODUCIBLE

	CO TO 60	VIX FN 240
61	CONTINUE	VIX FN 241
C		VIX FN 242
C	ASSIGNING OF BOUNDS ON THE ABSOLUTE DIFFERENCE BETWEEN THE	VIX FN 243
C	EXTRAPOLATED AND INTERPOLATED VALUES IN THE INTEGRATION	VIX FN 244
	CO 3 J=1, 16	VIX FN 245
	TEM=ABS(DVALUE(J))	VIX FN 246
	IF(TEM.LE.1.00)GO TO 4	VIX FN 247
	UPBND(J)=CHIGH(J)*TEM	VIX FN 248
	LNBNB(J)=CDOWN(J)*TEM	VIX FN 249
	CO TO 3	VIX FN 250
4	UPBND(J)=CHIGH(J)	VIX FN 251
	LNBNB(J)=CDOWN(J)	VIX FN 252
3	CONTINUE	VIX FN 253
C		VIX FN 254
	IF(LDPT.LT.3)GO TO 31	VIX FN 255
	IF(LDPT.EQ.4)GO TO 41	VIX FN 256
C		VIX FN 257
C	ASSIGNING TABULAR TRAJECTORY INPUT TO INTERNAL PROGRAM	VIX FN 258
C	SYMBOLS.	VIX FN 259
	TIMER=TRAJT(NK&1)	VIX FN 260
	TIME=TIMER	VIX FN 261
	TQ=TRAJT(1)	VIX FN 262
	Z=TRAJZ(NK&1)	VIX FN 263
	V=TRAJV(NK&1)	VIX FN 264
	IF(V.NE.0.00)GO TO 1016	VIX FN 265
	WRITE(6,1015)	VIX FN 266
1015	FORMAT(1H,10X,'*****WARNING***** CANNOT ACCEPT A ZERO VELOCITY,	VIX FN 267
	GOING TO NEXT CASE')	VIX FN 268
	RETURN	VIX FN 269
1016	CONTINUE	VIX FN 270
	WTAL=TRAJW(NK&1)	VIX FN 271
	KQ=TRAJRN(NK&1)/12.000	VIX FN 272
	ALPHA=TRAJALP(NK&1)/57.2597800	VIX FN 273
	ALPHIM=ALPHA	VIX FN 274
	IVALPH=1	VIX FN 275
	CO 74 KK=1, 8	VIX FN 276
	DDOT(2, KK)=0.000	VIX FN 277
	DDOT(3, KK)=0.000	VIX FN 278
	DDOT(2, KK)=0.000	VIX FN 279
74	DDOT(3, KK)=0.000	VIX FN 280
	DDOT(1, 1)=0.000	VIX FN 281
	DDOT(4, 1)=0.000	VIX FN 282
	DDOT(1, 1)=0.000	VIX FN 283
	DDOT(4, 1)=0.000	VIX FN 284
	CO TO 40	VIX FN 285
C		VIX FN 286
C	ASSIGNING TABULAR WIND TUNNEL INPUTS TO INTERNAL PROGRAM	VIX FN 287
C	SYMBOLS.	VIX FN 288
41	CONTINUE	VIX FN 289
	Z=WTZ(NK&1)	VIX FN 290
	WTP=WTPI(NK&1)	VIX FN 291
	IF(MIN.F.GE.5.00)GO TO 1046	VIX FN 292
	WRITE(6,1045)	VIX FN 293
1045	FORMAT(1H,10X,'*****WARNING***** CANNOT ACCEPT A MACH NUMBER LESS	VIX FN 294
	1 THAN 5, GOING TO NEXT CASE')	VIX FN 295
	RETURN	VIX FN 296
1046	CONTINUE	VIX FN 297
	WTPI=WTPI(NK&1)	VIX FN 298
	WTAL=WTPI(NK&1)	VIX FN 299

	TIMER=NK	V IX FN 300
	TIME=TIMER	V IX FN 301
	TC=0.000	V IX FN 302
	INALPH=1	V IX FN 303
	CO 7? KK=1,0	V IX FN 304
	MDOT(2, KK)=0.000	V IX FN 305
	MDOT(3, KK)=0.000	V IX FN 306
	QDOT(2, KK)=0.000	V IX FN 307
7?	QDOT(3, KK)=0.000	V IX FN 308
	MDOT(1, 1)=0.000	V IX FN 309
	MDOT(4, 1)=0.000	V IX FN 310
	QDOT(1, 1)=0.000	V IX FN 311
	QDOT(4, 1)=0.000	V IX FN 312
C	INPUT ALPHA TO BE USED WITH WIND TUNNEL CONDITIONS.	V IX FN 313
	ALPHA=ALPTAB(NK,1)/57.2957000	V IX FN 314
	ALPKIN=ALPHA	V IX FN 315
C		V IX FN 316
C	WHEN TRAJECTORY INPUT OR WIND TUNNEL INPUT TABLES ARE USED	V IX FN 317
C	THE INTEGRATION SUBROUTINE IS BYPASSED AND PRELIMINARY	V IX FN 318
C	CALCULATIONS, AERODYNAMIC HEATING, MASS LOSS, AND DRAG	V IX FN 319
C	CALCULATIONS ARE DONE FOR THE INPUT ALTITUDES ONLY.	V IX FN 320
4	CONTINUE	V IX FN 321
	LL=1	V IX FN 322
	CALL PRELIM1	V IX FN 323
	IF(MINF.LT.5.00)WRITE(6,32)	V IX FN 324
32	FORMAT(1NO,25X,57)***WARNING*** PROGRAM IS NOT VALID FOR MINF LESS	V IX FN 325
	1 THAN 5.0)	V IX FN 326
	IF(MDPT.EQ.0) GO TO 35	V IX FN 327
	IF(KBA? .GT. XUP)GO TO 35	V IX FN 328
	CALL AERODY	V IX FN 329
	IF(MHEAT.EQ.0) GO TO 35	V IX FN 330
	CALL MASSLO	V IX FN 331
	CALL TQALO	V IX FN 332
39	CONTINUE	V IX FN 333
	CALL DRAGCO	V IX FN 334
	GO TO 101	V IX FN 335
C		V IX FN 336
C		V IX FN 337
C	VARIABLES ARE INTEGRATED AND THE TRAJECTORY CALCULATED WHEN	V IX FN 338
C	LOPT IS LESS THAN OR EQUAL TO 2. THE QUANTITIES BEING	V IX FN 339
C	INTEGRATED IN ADMARK ARE AS FOLLOWS, WHERE DVALUE IS THE	V IX FN 340
C	INTEGRATED VALUE AND DEPIV IS THE CORRESPONDING DERIVATIVE	V IX FN 341
C	WITH ALTITUDE.	V IX FN 342
C	DVALUE(1)=V	V IX FN 343
C	DVALUE(2)=GMF	V IX FN 344
C	DVALUE(3)=TIME	V IX FN 345
C	DVALUE(4)=XR	V IX FN 346
C	DVALUE(5)=W	V IX FN 347
C	DVALUE(6)=RN	V IX FN 348
C	DVALUE(7)=RB	V IX FN 349
C	DVALUE(8)=P SI	V IX FN 350
C	DVALUE(9)=THEALP	V IX FN 351
C	DVALUE(10)=PHI	V IX FN 352
C	DVALUE(11)=Q	V IX FN 353
C	DVALUE(12)=SHR	V IX FN 354
C	DVALUE(13)=P	V IX FN 355
C	DVALUE(14)=YR	V IX FN 356
C	DVALUE(15)=PSIALP	V IX FN 357
C	DVALUE(16)=WTH	V IX FN 358
11	CONTINUE	V IX FN 359

	CALL ADM4PRK(16,DEL,DVALUF,OPRIV,UPRAD,DNRAD,0,200,FREQ,ZST,LP,	VIXEN360
	IZUSE,1,DDU)	VIXEN361
C		VIXEN362
	TIME=DVALUF(3)	VIXEN363
	Z=ZUSE	VIXEN364
	IF(LEP.NE.6)GO TO 7	VIXEN365
	WRITE(6,1005)	VIXEN366
1005	FORMAT(10X,34HPROGRAM ERROR, GOING TO NEXT CASE.)	VIXEN367
	GO TO 100	VIXEN368
C		VIXEN369
C	FOR THE LOPT=0 ROTATIONAL TRAJECTORY TESTS ARE DONE TO OBTAIN	VIXEN370
C	MAXIMUMS AND MINIMUMS IN ALPRIM.	VIXEN371
	7 IF(LOPT.NE.0)GO TO 30	VIXEN372
	IF(IKMAX.EQ.200)GO TO 51	VIXEN373
	IF(IKMIN.EQ.200)GO TO 51	VIXEN374
	CO TO(9,9,10),IALP	VIXEN375
	8 IALP=2	VIXEN376
	SUMA2=0.00	VIXEN377
	SUMA3=0.00	VIXEN378
	ALSTO1=ALPRIM	VIXEN379
	TSTO1=TIME	VIXEN380
	ZSTO1=Z	VIXEN381
	FSTO1=SMF	VIXEN382
	CO TO 20	VIXEN383
	9 IALP=3	VIXEN384
	ALSTO2=ALPRIM	VIXEN385
	TSTO2=TIME	VIXEN386
	ZSTO2=Z	VIXEN387
	FSTO2=SMF	VIXEN388
	CO TO 20	VIXEN389
10	TEMMAX=ALSTO1	VIXEN390
	IF(TEMMAX.LT.ALSTO2)TEMMAX=ALSTO2	VIXEN391
	IF(TEMMAX.LT.ALPRIM)TEMMAX=ALPRIM	VIXEN392
	TEMMIN=ALSTO1	VIXEN393
	IF(TEMMIN.GT.ALSTO2)TEMMIN=ALSTO2	VIXEN394
	IF(TEMMIN.GT.ALPRIM)TEMMIN=ALPRIM	VIXEN395
	IF(IKMAX.EQ.0)GO TO 21	VIXEN396
	TEMA=ALSTO1*ALSTO2	VIXEN397
	TEMT=TSTO2-TSTO1	VIXEN398
	SUMA2=SUMA2+C.5DC*TEMA*TEMT	VIXEN399
	SUMA3=SUMA3+C.5DC*TEMA*TEMT	VIXEN400
21	CONTINUE	VIXEN401
	ALSTO1=ALSTO2	VIXEN402
	ALSTO2=ALPRIM	VIXEN403
	TSTO1=TSTO2	VIXEN404
	TSTO2=TIME	VIXEN405
	ZSTO1=ZSTO2	VIXEN406
	ZSTO2=Z	VIXEN407
	FSTO1=FSTO2	VIXEN408
	FSTO2=SMF	VIXEN409
	IF(TEMMAX.EQ.ALSTO1)GO TO 11	VIXEN410
	IF(TEMMIN.EQ.ALSTO1)GO TO 12	VIXEN411
	CO TO 20	VIXEN412
11	IKMAX=IKMAX+1	VIXEN413
	IF(IKMAX.GE.200)WRITE(6,2000)Z	VIXEN414
2000	FORMAT(10X,46H*****WARNING***** ALPHA MAX TABLE FILLED AT Z=,	VIXEN415
	1F17.6,44H NO MORE MAXIMUMS OR MINIMUMS WILL BE STORED)	VIXEN416
	IF(IKMAX=200,14,15	VIXEN417
12	ALSTO1=ALSTO1	VIXEN418
	TEMT=TSTO1	VIXEN419

ALMAX(IKMAX)=ALWIG1	V IX FN 420
TAMAX(IKMAX)=TWIG1	V IX FN 421
FMAX(IKMAX)=FST01	V IX FN 422
ZMAX(IKMAX)=ZST01	V IX FN 423
CO TO 20	V IX FN 424
14 ALWIG2=ALST01	V IX FN 425
TWIG2=TST01	V IX FN 426
ALMAX(IKMAX)=ALWIG2	V IX FN 427
TAMAX(IKMAX)=TWIG2	V IX FN 428
FMAX(IKMAX)=FST01	V IX FN 429
ZMAX(IKMAX)=ZST01	V IX FN 430
CO TO 16	V IX FN 431
15 ALWIG1=ALWIG2	V IX FN 432
ALWIG2=ALST01	V IX FN 433
TWIG1=TWIG2	V IX FN 434
TWIG2=TST01	V IX FN 435
ALMAX(IKMAX)=ALWIG2	V IX FN 436
TAMAX(IKMAX)=TWIG2	V IX FN 437
ZMAX(IKMAX)=ZST01	V IX FN 438
FMAX(IKMAX)=FST01	V IX FN 439
CO TO 16	V IX FN 440
12 TBARP=TST01	V IX FN 441
ALBARP=ALST01	V IX FN 442
IKMIN=IKMIN61	V IX FN 443
IF(IKMIN.EQ.20)WRITE(6,2001)Z	V IX FN 444
2001 FORMAT(1H 20K,4EH****WARNING**** ALPHA MIN TABLE FILLED AT Z=,	V IX FN 445
IF(12.6,44H NO MORE MINIMUMS OR MAXIMUMS WILL BE STORED)	V IX FN 446
ALMIN(IKMIN)=ALBARP	V IX FN 447
TAMIN(IKMIN)=TBARP	V IX FN 448
FMIN(IKMIN)=FST01	V IX FN 449
ZMIN(IKMIN)=ZST01	V IX FN 450
CO TO 20	V IX FN 451
16 T=TWIG2-TWIG1	V IX FN 452
SUF=1.00/T	V IX FN 453
ABAR 2=0.00	V IX FN 454
ABAR 3=0.00	V IX FN 455
CG 22 I=1.3	V IX FN 456
II=I-1	V IX FN 457
IF(I=METAD**II	V IX FN 458
ISUB 11=120**II	V IX FN 459
ISUB 12=ISUB 1161	V IX FN 460
CG 22 J=1.3	V IX FN 461
JJ=J-1	V IX FN 462
TEMPJ=1.00	V IX FN 463
IF(JJ.NE.0)TEMJ=LAMSTO**JJ	V IX FN 464
ISUB 1=ISUB 1109**JJ	V IX FN 465
ISUB 2=ISUB 1209**JJ	V IX FN 466
ABAR 2=ABAR 26A(I SUB 1)*TEMJ*TEMPJ	V IX FN 467
22 ABAR 3=ABAR 36A(I SUB 2)*TEMJ*TEMPJ	V IX FN 468
TEMP I=100.00/PI	V IX FN 469
CPAT I=TEMP I*(ABAR 2*SUMA 2&TF MPI*ABAR 3*SUMA 3)	V IX FN 470
SUMA 2=0.00	V IX FN 471
SUMA 3=0.00	V IX FN 472
IF(IALWIG2.GE.ALST00)GO TO 70	V IX FN 473
LEPT=1	V IX FN 474
LP=1	V IX FN 475
T=0.00	V IX FN 476
ALPHA=0.00	V IX FN 477
ALPR IG=0.00	V IX FN 478
CO TO 30	V IX FN 479

2)	LAMST0=LAMDA	VIXFN480
	ALPENV=ALPRIM	VIXFN481
	IF(T.EQ.0)GO TO 3C	VIXFN482
	ALPENV=ALWIG1*(TWIG1-TIME)*(ALWIG2-ALWIG1)/(TWIG1-TWIG2)	VIXFN483
3)	CONTINUE	VIXFN484
C		VIXFN485
C	FOR THE SIMPLIFIED ANGLE OF ATTACK CALCULATION, LOPT=2, ALPHA	VIXFN486
C	ENVELOPE IS SET EQUAL TO INSTANTANEOUS VALUE OF ALPHA.	VIXFN487
	IF(LOPT.NE.2)GO TO 51	VIXFN488
	ALPENV=ALPRIM	VIXFN489
9)	CONTINUE	VIXFN490
	IF(LOPT.EQ.1)ALPENV=ALPRIM	VIXFN491
C		VIXFN492
C	BETA, VDDT/G, AND D(BETA)/Z CALCULATION	VIXFN493
	CDW=CD*AREF*CD/WTOTAL	VIXFN494
	BETA=BTATA	VIXFN495
	BETA=WTOTAL/(CD*AREF)	VIXFN496
	IF(IGM(10).EQ.1) BETA = WTOTAL/(CD*AREF-TH/QD)	VIXFN497
	IF(IGM(10).EQ.2)BETA=1.0/(WTOTAL/(CD*AREF-TH/QD))	VIXFN498
	DENOM=7-ZSTO	VIXFN499
	IF(DENM.EQ.0)DENOM=1.00-10	VIXFN500
	BETAD=(BETA-BETA ST)/DENOM	VIXFN501
	ZSTO=Z	VIXFN502
	VDDOT=DERIV(1)/(G*DERIV(3))	VIXFN503
C		VIXFN504
C	TESTS FOR PRINTING IF LOPT IS LESS THAN OR EQUAL TO 2	VIXFN505
	IF(Z.LE.(ZSTO1.CD))GO TO 54	VIXFN506
	IF(ABS(PRINTZ-Z).LT.DELPPI)GO TO 50	VIXFN507
54	PRINTZ=Z	VIXFN508
	IF(PRINTZ.LE.(ZBAPE.C.IDC))DELPPI=7*PR2	VIXFN509
	FREQ=-DELPPI	VIXFN510
C		VIXFN511
C	SETTING PRINTOUT VALUES FOR LOPT=3 OR 4 OPTIONS	VIXFN512
101	CONTINUE	VIXFN513
	IF(LOPT.LT.3)GO TO 100	VIXFN514
	BETA=WTOTAL/(CD*AREF)	VIXFN515
	VDDOT=C.CDO	VIXFN516
	DVALUE(1)=V	VIXFN517
	CVALUE(4)=C.CDO	VIXFN518
	IVALUE(6)=RN	VIXFN519
	CVALUE(12)=0.CDO	VIXFN520
	PSIALP=C.CDO	VIXFN521
	GO TO 80	VIXFN522
C		VIXFN523
C	STORAGE FOR THE TAPE OPTION	VIXFN524
10	CONTINUE	VIXFN525
	NNTAPE=NNTAPE&1	VIXFN526
C		VIXFN527
C	STORAGE OF VALUES FOR PLOTTING	VIXFN528
	IF(LPLOT.EQ.16)GO TO 80	VIXFN529
	LPLOT=LPLOT&1	VIXFN530
	IF(LPLOT.EQ.16)WRITE(6,100)Z,TIME	VIXFN531
105	FORBIDDING 10X,4TH*****WARNING***** PLOTS HAVE BEEN CUT OFF AT Z=	VIXFN532
	1E13.6,100 AND TIME=7.2)	VIXFN533
	ZPLOT(LPLOT)=Z	VIXFN534
	TPLOT(LPLOT)=TIME	VIXFN535
	PLINPL(LPLOT)=TIME	VIXFN536
	CPLOT(LPLOT)=CD	VIXFN537
	CDPLT(LPLOT)=CD&CDP	VIXFN538
	I(Z.GT.ZTR)CPPLT(LPLOT)=CDFINF(2,2,1)	VIXFN539

```

IFIZ,LT,ZTR)COEPLT(LPLOT)=CDFINF(2,1,1)
COEPLT(LPLOT)=CDI
VIXFN540
VIXFN541
VIXFN542
VIXFN543
VIXFN544
VIXFN545
VIXFN546
VIXFN547
VIXFN548
VIXFN549
VIXFN550
VIXFN551
VIXFN552
VIXFN553
VIXFN554
VIXFN555
VIXFN556
VIXFN557
VIXFN558
VIXFN559
VIXFN560
VIXFN561
VIXFN562
VIXFN563
VIXFN564
VIXFN565
VIXFN566
VIXFN567
VIXFN568
VIXFN569
VIXFN570
VIXFN571
VIXFN572
VIXFN573
VIXFN574
VIXFN575
VIXFN576
VIXFN577
VIXFN578
VIXFN579
VIXFN580
VIXFN581
VIXFN582
VIXFN583
VIXFN584
VIXFN585
VIXFN586
VIXFN587
VIXFN588
VIXFN589
VIXFN590
VIXFN591
VIXFN592
VIXFN593
VIXFN594
VIXFN595
VIXFN596
VIXFN597
VIXFN598
VIXFN599

```

RC CONTINUE

IF (IOP(74),EQ,1) CALL WAKE

IF NPRINT = C, PRINTOUT IS OMITTED OUT TAPES AND PLOTS MAY
STILL BEING GENERATED.
IF NPRINT, EQ, 0 GO TO 8C

BEGINNING OF PRINTOUT

PRINTOUT OF TRANSLATIONAL QUANTITIES
ALL ANGLES ARE CHANGED INTO UNITS OF DEGREES BEFORE BEING
PRINTED OUT.

PSALP=57.2957800*DMVALUE(15)
GAMFP=57.2957800*DMVALUE(2)
PSPG=PS/2116.000
UNIT(1),1032)TIME,Z ,DMVALUE(1),GAMFP,DMVALUE(4),BETA,ZTR,
100,NIIF,VGGOUT,BETAP,TH,TTMATE(1),DMVALUE(14),PSALP,ICW
1032 FURNATI(1)C,777,54X,24)TRANSLATIONAL QUANTITIES/10X,9)TIME=F7.2,
15H Z=E15.6,5H V=E15.7,8H GAMF=E12.4,6H XR=E15.6,8H BETA=VIXFN585
7E11.4/25X,4HZTR=C13.6,6H DD=F14.4,4H *INF=E12.4,10H VDOT,6=EVIXFN586
201.4,9H BETAP=F11.4/25X,3HTH=E14.4,7H TXT=E13.4,6H YK=E14.4,VIXFN587
410H PSALP=C11.4,7H D/W=F13.4)
1075 FURNATI(1)C,777,54X,24)INPUT ATMOSPHERE BEING USED WHEN Z LESS THAN *,
2E14.6,*, GREATER THAN OR EQUAL TO*,E14.6/)
75 CONTINUE

UPAC COEFFICIENT PRINTOUT
IFIZ,LT,ZTR)J=1
IFIZ,GF,ZTR)J=2

```

IF(J.EQ.1)JJ=3
IF(J.EQ.2)JJ=2
REYLA=(REYL*LA)/CAPL
WRITE(6,1034)CD,CDP,TITLE(J),CDFINF(2,J,1),CDR,CDPU,TITLE(J),
VIXEN600
VIXEN601
VIXEN602
VIXEN603
VIXEN604
1034 FORMAT(1H0,5X,15HPRAG QUANTITIES//25X,3HCDF=E14.4,7H CDP=E13.4,
VIXEN605
113H CDFINF(HI,A4,5H,NB)=E12.4,10X,4HCDF=E13.4/45X,5HCLPU=E12.4,
VIXEN606
213H CDFINF(BL,A4,5H,NB)=E12.4,10X,4HCDF=E13.4/45X,5HKEAR=E12.4,
VIXEN607
212H REYINFLA=,10X,E12.4,10X,6HXBAR1=E11.4)
VIXEN608
IF(J.EQ.1)GO TO 1040
WRITE(6,1039)DFLDDP,DECDFF,DECFIC
VIXEN609
VIXEN610
1039 FORMAT(54X,2PHLAMINAR COT COMPONENTS/54X,6HCDF/P=E11.4,10H CDF/SV
VIXEN611
IF=,12X,E12.4,10X,7HCDF/TC=E11.4)
VIXEN612
1040 CONTINUE
VIXEN613
C
VIXEN614
C PPINTOUT OF CONFIGURATION PARAMETERS
VIXEN615
LA=LA*12.CDG
VIXEN616
DVALUE(6)=DVALUE(6)*12.CDG
VIXEN617
WRITE(6,1036)DVALUE(6),THE TAD,LA,LAMDA,ARFF,WTCTAL,DELW,DELW2,
VIXEN618
DELW?
VIXEN619
1036 FORMAT(1H0,5X,13HCONFIGURATION//25X,3HRN=E14.4,9H THETA=E11.4,
VIXEN620
16H LA=E14.4,10H LAMDA=E11.4,8H ARFF=E12.4/25X,2HW=E15.4,
VIXEN621
20H DELW=E12.4,8H WABL=E12.4,10H WTHRST=E11.4)
VIXEN622
LA=LA/12.CDG
VIXEN623
DVALUE(6)=DVALUE(6)/12.CDG
VIXEN624
C
VIXEN625
C IF THE MASS LOSS OPTIONS ARE NOT BEING USED, THE PRINTOUT IS
VIXEN626
BYPASSED.
VIXEN627
C
IF(MOPT.EQ.0)GO TO 88
VIXEN628
C
VIXEN629
C HEATING AND MASS LOSS PRINTOUT
VIXEN630
IF(TIMER.EQ.0)TLAST=C.DDG
VIXEN631
DO 24 I=1,8
VIXEN632
IF(TLAST.EQ.0)GO TO 24
VIXEN633
QINT(I)=QINT(I)&C.500*(QDOT(JJ,I)&LAST(I))*(TIMER-TLAST)
VIXEN634
24 QLAST(I)=QDOT(JJ,I)
VIXEN635
IF(TLAST.EQ.0)GO TO 26
VIXEN636
QINST=QINST&C.500*(QDOT(1,1)&QTLST)* (TIMER-TLAST)
VIXEN637
QINSON=QINSON&C.500*(QDOT(4,1)&QLTSON)* (TIMER-TLAST)
VIXEN638
GO TO 28
VIXEN639
26 DO 27 I=1,8
VIXEN640
27 QINT(I)=0.CDG
VIXEN641
QINST=0.CDG
VIXEN642
QINSON=0.CDG
VIXEN643
28 QTLST=QDOT(1,1)
VIXEN644
QLTSON=QDOT(4,1)
VIXEN645
TLAST=TIMER
VIXEN646
WRITE(6,1035)QDOT(1,1),QDOT(4,1),HSRTO,PSPO,(PFRS(LJ),LJ=1,8),
VIXEN647
(QDOT(LJ,LJ),LJ=1,8),(QDOT(LJ,LJ),LJ=1,8),(QINT(LJ),LJ=1,8),QINST,
VIXEN648
QINSON
VIXEN649
1035 FORMAT(1H0,61X,9HMASS LOSS//22X,11HQDOT(5TAG)=E11.4,7X,12HQDOT(SUNV
VIXEN650
11C)=E11.4,7X,6HHSRTO=E11.4,7X,5HPSPO=E12.4/12X,46HX STATIONS ARE PV
VIXEN651
PERCENTAGES OF UNADLATED LENGTH//22X,9HTANG. PT.,5X,8HX/LA=0.2,5X,
VIXEN652
8HX/LA=0.4,5X,8HX/LA=0.6,5X,8HX/LA=0.75,4X,8HX/LA=0.9,5X,8HX/LA=1.0
VIXEN653
4X,5X,9HCONE R=RH/12X,7HPFRS= RE13.4/12X,7HCOT= RE13.4/12X,7HDDUV
VIXEN654
ST= ,RE13.4/12X,7HINT= ,RE13.4/22X,13HINT(5TAG)= E11.4,24X,15H
VIXEN655
6 QINT(SUNIC)= E11.4,16H TPRALLET ONLY )
VIXEN656
C
VIXEN657
C CONTINUE
VIXEN658
IF(IALPH.GT.C100)GO TO 82
VIXEN659

```

	IF(LOPT.EQ.1)GO TO 52	VIXEN660
C		VIXEN661
C	ROTATIONAL QUANTITIES PRINTOUT FOR LOPT = 0 CR 2	VIXEN662
	PSIP=57.29578D(*DVALUE(8))	VIXEN663
	THLP=57.29578D(*DVALUE(9))	VIXEN664
	PHIP=57.29578D(*DVALUE(10))	VIXEN665
	ALPRIP=57.29578D(*ALPRIM)	VIXEN666
	ALENVP=57.29578D(*ALPENV)	VIXEN667
	IF(SMF.FQ.C.CD())TRAP=C.CD()	VIXEN668
	IF(SMF.NE.O.JDD)TRAP=1.C/O/SMF	VIXEN669
	AAA=PHIP/360.DC	VIXEN670
	NNN=AAA	VIXEN671
	HBB=NNN	VIXEN672
	PHIP=PHIP-360.DC*RRR	VIXEN673
	IF(PHIP.LT.O.DC)PHIP=360.DC&PHIP	VIXEN674
	WRITE(6,1033)PSIP,THLP,PHIP,ALPRIP,DVALUE(13),DVALUE(11),DVALUE(12)	VIXEN675
	11,ALENVP,CMO,CM,CN,SMF,TRAP	VIXEN676
1033	FORMAT(1H0,5X,21HROTATIONAL QUANTITIES//25X,4HPSI=E13.4,10H THLVIXEN677	
	1ALP=E11.4,6H PHI=E13.4,10H ALPRM=E11.4/25X,3HPR=E14.4,5H J=,VIXEN678	
	2E14.4,8H SMR=E13.4,10H ALPENV=E11.4/25X,4HCNC=E13.4,6H CM=EVIXEN679	
	3E13.4,7H CN=E14.4,7H SMF=E14.4,8H TRAP=E12.4)	VIXEN680
	GO TO 50	VIXEN681
C		VIXEN682
C	INPUT ANGLE OF ATTACK PRINTOUT	VIXEN683
	52 CONTINUE	VIXEN684
	ALPHID = ALPHA * 57.29578DC	VIXEN685
	WRITE(6,1042) ALPHID	VIXEN686
1042	FORMAT(1H0,4X,37HTABLLAR INPUT ANGLE OF ATTACK ALPHA =,E12.4)	VIXEN687
	IF(LOPT.LE.2)GO TO 50	VIXEN688
		VIXEN689
C		VIXEN690
C	SEQUENCING FOR LOPT=3 OR 4 WHICH CAUSES REPEAT OF CALCULATION	VIXEN691
C	FOR NEXT SET OF VALUES IN TABLES, UNLESS FINAL SET OF VALUES	VIXEN692
C	HAS ALREADY BEEN USED - THEN CAUSES RETURN TO MAIN PROGRAM.	VIXEN693
C	PLOTTING ROUTINES ARE BYPASSED WHEN LOPT = 3 OR 4.	VIXEN694
	49 CONTINUE	VIXEN695
	IF(MAXVAL.EQ.O)WRITE(6,1025)	VIXEN696
	IF(MAXVAL.EQ.O)MAXVAL=1	VIXEN697
1025	FORMAT(1H0,10X,*****WARNING***** YOU FORGCT TO INPUT MAXVAL FOR LVIXEN698	
	LOPT=3 OR 4 CASE*)	VIXEN699
	IF(NK.EQ.(MAXVAL-1))GO TO 834	VIXEN700
	NK=NK&1	VIXEN701
	GO TO 2	VIXEN702
C		VIXEN703
C		VIXEN704
C	STOPPING TESTS FOR LOPT LESS THAN OR EQUAL TO 2.	VIXEN705
90	CONTINUE	VIXEN706
	IF(TIMF.GT.T3T)GO TO 51	VIXEN707
	IF(Z.LE.(ZST&1.DC))GO TO 51	VIXEN708
	IF(LP.GE.3)GO TO 51	VIXEN709
	GO TO 50	VIXEN710
51	IF(NPRINT.NE.O)CALL RITOUT	VIXEN711
C		VIXEN712
	CO 81 J=1,5	VIXEN713
	IF(NPLOT(J).GT.C)GO TO 82	VIXEN714
81	CONTINUE	VIXEN715
	GO TO 833	VIXEN716
C		VIXEN717
C	PLOTTING OPTIONS	VIXEN718
82	CONTINUE	VIXEN719
	IF(NPLOT(1).LE.C)GO TO 84	

CALL AVPLT(TIXZ,T1,ZPLOT,CDPLCT,ZPLCT,3,1)	VIXEN720
CALL AVPLT(TIXM,T2,PLMINF,CDPL1,ZPLOT,3,1)	VIXEN721
CALL AVPLT(TIXZ,T3,ZPLOT,CDPL1,ZPLCT,3,1)	VIXEN722
CALL AVPLT(TIXZ,T4,ZPLOT,CDPL1,ZPLOT,3,1)	VIXEN723
84 CONTINUE	VIXEN724
IF(NPLOT(2).LE.0)GO TO 85	VIXEN725
CALL AVPLT(TIXZ,T6,ZPLOT,HETAPL,ZPLOT,3,1)	VIXEN726
CALL AVPLT(TIXT,T7,TPLCT,VPLCT,ZPLOT,3,1)	VIXEN727
CALL AVPLT(TIXT,T8,TPLCT,PLMINF,ZPLOT,3,1)	VIXEN728
CALL AVPLT(TIXT,T9,TPLCT,VCGPL1,ZPLOT,3,1)	VIXEN729
CALL AVPLT(TIXT,T10,TPLCT,ZPLCT,ZPLOT,3,1)	VIXEN730
CALL AVPLT(TIXT,T11,TPLCT,QPLCT,ZPLOT,3,1)	VIXEN731
85 CONTINUE	VIXEN732
IF(NPLOT(3).LE.0)GO TO 86	VIXEN733
CALL AVPLT(TIXZ,T5,ZPLOT,WOTPL,ZPLOT,3,1)	VIXEN734
CALL AVPLT(TIXT,T12,TPLCT,PSPLCT,ZPLOT,3,1)	VIXEN735
CALL AVPLT(TIXT,T13,TPLCT,HSP1PL,ZPLOT,3,1)	VIXEN736
IF(LAMDA.NE.C.0)GO TO 851	VIXEN737
CALL AVPLT(TIXT,T14,TPLCT,QD11PL,ZPLOT,3,1)	VIXEN738
CALL AVPLT(TIXT,T17,TPLCT,QD8PL1,ZPLOT,3,1)	VIXEN739
CALL AVPLT(TIXT,T1E,TPLCT,PE1PL1,ZPLOT,3,1)	VIXEN740
CALL AVPLT(TIXT,T2C,TPLCT,PE8PL1,ZPLOT,3,1)	VIXEN741
GO TO 86	VIXEN742
851 CALL AVPLT(TIXT,T14,TPLCT,QD11PL,ZPLOT,3,1)	VIXEN743
CALL AVPLT(TIXT,T15,TPLCT,QD41PL,ZPLOT,3,1)	VIXEN744
CALL AVPLT(TIXT,T16,TPLCT,QD7PL1,ZPLOT,3,1)	VIXEN745
CALL AVPLT(TIXT,T17,TPLCT,QD8PL1,ZPLOT,3,1)	VIXEN746
CALL AVPLT(TIXT,T18,TPLCT,PE1PL1,ZPLOT,3,1)	VIXEN747
CALL AVPLT(TIXT,T19,TPLCT,PE7PL1,ZPLOT,3,1)	VIXEN748
CALL AVPLT(TIXT,T2C,TPLCT,PE8PL1,ZPLOT,3,1)	VIXEN749
86 CONTINUE	VIXEN750
IF(NPLOT(4).LE.0)GO TO 87	VIXEN751
CALL AVPLT(TIXT,T21,TPLCT,ALPLOT,ZPLOT,3,1)	VIXEN752
87 CONTINUE	VIXEN753
IF(NPLOT(5).LE.0)GO TO 833	VIXEN754
CALL AVPLT(TIXT,T22,TPLCT,OCPLCT,ZPLOT,3,1)	VIXEN755
833 CONTINUE	VIXEN756
IF(IOP(74).EQ.0) GO TO 837	VIXEN757
IF(IOP(77).EQ.1) CALL AVPLT(TIXZ,T23,ZPLOT,WL1P,ZPLCT,ICBL,1)	VIXEN758
IF(IOP(78).EQ.1) CALL AVPLT(TIXZ,T24,ZPLOT,WL2P,ZPLCT,ICBL,1)	VIXEN759
IF(IOP(79).EQ.1) CALL AVPLT(TIXZ,T25,ZPLOT,WL3P,ZPLCT,ICBL,1)	VIXEN760
IF(IOP(80).EQ.1) CALL AVPLT(TIXZ,T26,ZPLOT,WR1P,ZPLCT,ICBL,0)	VIXEN761
IF(IOP(81).EQ.1) CALL AVPLT(TIXZ,T27,ZPLOT,WR2P,ZPLCT,ICBL,0)	VIXEN762
IF(IOP(82).EQ.1) CALL AVPLT(TIXZ,T28,ZPLOT,WR3P,ZPLCT,ICBL,0)	VIXEN763
837 CONTINUE	VIXEN764
C	VIXEN765
C OPTION FOR TAPE OF V, BETA, Z	VIXEN766
IF(ITAPE.EQ.0)GO TO 834	VIXEN767
VVAR=0.000	VIXEN768
WRITE(P)DATE,END,CASE,NNTAPE	VIXEN769
WRITE(S)(VVAR,ZPLOT(I),VPLCT(I),VVAR,RETAPL(I),VVAR,VVAR,VVAR,	VIXEN770
IVVAR,I=1,NNTAPE)	VIXEN771
C	VIXEN772
C CHANGING ANGLES BACK TO UNITS OF DEGREES TO PREPARE FOR NEW	VIXEN773
C INPUT QUANTITIES.	VIXEN774
834 CONTINUE	VIXEN775
THETA=THETA/0.0174532900	VIXEN776
CANFC=GAMFC/0.0174532900	VIXEN777
PHIC=PHIC/0.0174532900	VIXEN778
PSIC=PSIC/0.0174532900	VIXEN779

34. CUBI LNU
THETA=THETA/O.O1745325DC
GAMFC=GAMFC/O.O1745325DC
PHIO=PHIO/O.O1745325DC
PSIO=PSIO/O.O1745325DC

V IX FN 775
V IX FN 776
V IX FN 777
V IX FN 778
V IX FN 779

THEALC=THEALC/O.O1745325DC
RETURN
END

V IX FN 780
V IX FN 781
V IX FN 782

WAKE		WAKE	1
SUBROUTINE WAKE		WAKE	2
IMPLICIT REAL*(A-H,O-Z)		WAKE	3
REAL *8 MUINF, MTNF, ME		WAKE	4
COMMON /CPCCUR/ PCCUR(11770)		WAKE	5
COMMON /CICCUR/ ICOCUR(320)		WAKE	6
COMMON OCCUR, NOCCUR		WAKE	7
COMMON /INPUT/ AKW, BHOW, CPHOW, DELWH, TABL, CHCHEM, RHCSL,		WAKE	8
1 BTO, BCON, ACON, R21, R22, R23, C(169)		WAKE	9
COMMON /CHAKE/ BETA7, PRO1, PRO2, PRO3, PHI1, PHI2, PHI3, SIGNAL1, SIGNAL2,		WAKE	10
1 SIGNAL3, TAU1, TAU2, TAU3, WKALT, WSTALT, IND, IWAKE, IWPRNT		WAKE	11
COMMON /CPROSEC/ T, TAU, SIGMDS, PHI, XEZ, ZME, HH, CDW, C1, C2, C3, C4, C5		WAKE	12
COMMON /FRLTNK/ R7, RN, THETAC, DR, GAMMAE, CDTA, CDVA, PICIPU, PU,		WAKE	13
1 AMU, PHOU, UU, HU, AMUJ, AMINFC, RHOC, HTNFC, MUINFC, AMULSD,		WAKE	14
2 AMABLD, Z, SC, ZBLT		WAKE	15
DIMENSION OCCUR(1000)		WAKE	16
DIMENSION NOCCUR(30)		WAKE	17
DIMENSION WL1P(160), WL2P(160), WL3P(160), WR1P(160), WR2P(160)		WAKE	18
DIMENSION WP3P(160), ZPLOT(160), A(514), TOP(90)		WAKE	19
DIMENSION BETA7(10), PHI(10), PHI2(10), PHI3(10), WKALT(10)		WAKE	20
EQUIVALENCE(OCCUR(001), ARFF)		WAKE	21
EQUIVALENCE(OCCUR(015), CFI)		WAKE	22
EQUIVALENCE(OCCUR(017), CPF)		WAKE	23
EQUIVALENCE(OCCUR(021), O)		WAKE	24
EQUIVALENCE(OCCUR(021), GAME)		WAKE	25
EQUIVALENCE(OCCUR(034), MUINF)		WAKE	26
EQUIVALENCE(OCCUR(035), MTNF)		WAKE	27
EQUIVALENCE(OCCUR(037), ME)		WAKE	28
EQUIVALENCE(OCCUR(049), PF)		WAKE	29
EQUIVALENCE(OCCUR(049), PINF)		WAKE	30
EQUIVALENCE(OCCUR(052), RNMN)		WAKE	31
EQUIVALENCE(OCCUR(056), RHONIT)		WAKE	32
EQUIVALENCE(OCCUR(061), RHOF)		WAKE	33
EQUIVALENCE(OCCUR(072), TINF)		WAKE	34
EQUIVALENCE(OCCUR(074), THETA)		WAKE	35
EQUIVALENCE(OCCUR(079), TE)		WAKE	36
EQUIVALENCE(OCCUR(082), V)		WAKE	37
EQUIVALENCE(OCCUR(093), VE)		WAKE	38
EQUIVALENCE(OCCUR(094), WOOT)		WAKE	39
EQUIVALENCE(OCCUR(090), XBAR)		WAKE	40
EQUIVALENCE(OCCUR(091), ZNK)		WAKE	41
EQUIVALENCE(OCCUR(092), ZTR)		WAKE	42
EQUIVALENCE(OCCUR(099), COP)		WAKE	43
EQUIVALENCE(OCCUR(099), COB)		WAKE	44
EQUIVALENCE(OCCUR(100), COT)		WAKE	45
EQUIVALENCE(OCCUR(239), XLCM)		WAKE	46
EQUIVALENCE(OCCUR(301), A)		WAKE	47
EQUIVALENCE(OCCUR(00001), TOP)		WAKE	48
EQUIVALENCE(OCCUR(00316), IDPL)		WAKE	49
EQUIVALENCE(OCCUR(00302), ZPLOT)		WAKE	50
EQUIVALENCE(OCCUR(01291), WL1P)		WAKE	51
EQUIVALENCE(OCCUR(01441), WL2P)		WAKE	52
EQUIVALENCE(OCCUR(01401), WL3P)		WAKE	53
EQUIVALENCE(OCCUR(00901), WR1P)		WAKE	54
EQUIVALENCE(OCCUR(00941), WR2P)		WAKE	55
EQUIVALENCE(OCCUR(01121), WP3P)		WAKE	56
EQUIVALENCE(OCCUR(00161), ZPLOT)		WAKE	57
EQUIVALENCE(NOCCUR(03), MOPT)		WAKE	58
NAMELIST/NAME/ HZ, RN, THETAC, DR, GAMMAE, CDTA, CDVA, PICIPU, PL,		WAKE	59

1	AMU,PHOU,TH,PU,AMUJ,APINFC,PHOIC,HINFC,UIINFC,AMBLSO,	WAKE 60
2	AMARLO,Z,SC,ZBLT	WAKE 61
C		WAKE 62
C		WAKE 63
C	SUBROUTINE WAKE CALCULATES SOME QUANTITIES NEEDED BY SLO-	WAKE 64
C	ROUTINES FLOWE AND RCSEC, THEN CALLS THESE SUBROUTINES TO	WAKE 65
C	CALCULATE THE VEHICLE WAKE LENGTH AND CROSS SECTION AT EACH OF	WAKE 66
C	THREE RADAR FREQUENCIES	WAKE 67
C		WAKE 68
C		WAKE 69
C	IF(INOINT,50.0) GO TO 10	WAKE 70
C	WRITE(5,1000) (A(I),I=105,112),ARFF,ZTR,ZNX,D,RNX,THETA,GAMF,CC,	WAKE 71
C	1 COP,COR,COT,PR,PINF,MINF,PHOINI,V,TINF,MUTNF,ME,RHOE,CPC,TE,	WAKE 72
C	3 VE,WOT,XHAP,XLON	WAKE 73
C		WAKE 74
10	IF(ZNX,LE,WSTAL T,AND,XBAR,LT,XLON) GO TO 30	WAKE 75
C	WL1P(PLOT) = 0.00	WAKE 76
C	WL2P(PLOT) = 0.00	WAKE 77
C	WL3P(PLOT) = 0.00	WAKE 78
C	WP1P(PLOT) = 1.00	WAKE 79
C	WP2P(PLOT) = 1.00	WAKE 80
C	WP3P(PLOT) = 1.00	WAKE 81
C	RETURN	WAKE 82
C		WAKE 83
30	CALL INTERP(10,(WAKE,ZPLCT(LPLOT),WKALT,BETAZ,BZ)	WAKE 84
C	FUN1 = 1.03	WAKE 85
C	FUN2 = 778.00*Z2.17400	WAKE 86
C		WAKE 87
C	THE FOLLOWING CALCULATIONS ARE FOR RCSEC	WAKE 88
C	X07 = R7	WAKE 89
C	ZND = MINE	WAKE 90
C	MM = ZNX / FUN1	WAKE 91
C	COV = 0.1,0.100	WAKE 92
C	C1 = C(1)	WAKE 93
C	C2 = C(2)	WAKE 94
C	C3 = C(3)	WAKE 95
C	C4 = C(4)	WAKE 96
C	C5 = C(5)	WAKE 97
C		WAKE 98
C	THE FOLLOWING CALCULATIONS ARE FOR PLOMF.	WAKE 99
C	RM = RMX	WAKE 100
C	THETAC = THETA	WAKE 101
C	OR = 0	WAKE 102
C	GAMFAP = GASSIGAMF)	WAKE 103
C	COTA = COTARCF	WAKE 104
C	COVA = (CO - (COR & COT)) * ARFF	WAKE 105
C	PICINF = PR/PINF	WAKE 106
C	PI = PINF	WAKE 107
C	MHI = MINE	WAKE 108
C	PHOI = PHOINI	WAKE 109
C	UI = V / FUN1	WAKE 110
C	IF(TINF,GT,0.03) GO TO 6A	WAKE 111
C	IF(TINF,LT,7.02) GO TO 7C	WAKE 112
C	CRINF = 0.00	WAKE 113
C	GO TO 1 = 1, *	WAKE 114
71	CRINF = CRINF & A11C104) * TINF**([1-1)	WAKE 115
C	GO TO 72	WAKE 116
6A	CRINF = A1111) & A1112) * TINF	WAKE 117
C	GO TO 72	WAKE 118
70	CRINF = .235800	WAKE 119

72	MU = C*INF*TIME*FUG2	WAKE 120
	AMIU = MU*INF	WAKE 121
	AMINFC = ME	WAKE 122
	PHOIC = RHOE	WAKE 123
	HINFC = C*P*TE*FUG2	WAKE 124
	HINFC = VC/FUG1	WAKE 125
	IF (RN.GT..49500*DR) GO TO 78	WAKE 126
	FUG3 = PIC(PH*AMINFC/AMU	WAKE 127
	FUG4 = 0.003*PHCU *UU*AMUU/(PHOSL*OSIN(THETAC))	WAKE 128
	AMBLSD = 1005.00 * (HINFC / MU)**.2500 * CSQRT(FUG3 * FUG4)	WAKE 129
	GO TO 79	WAKE 130
79	FUG5 = 1.02*PHOL*UU*DB/ AMLU	WAKE 131
	AMBLSD = C(.75) * PHOU * UU *AREF* FUG5**C(.77)	WAKE 132
79	AMBLSD = -WOUT	WAKE 133
	IF (MORT.FO.0) AMBLSD = 0.000	WAKE 134
	Z = ZNX / FUG1	WAKE 135
	IF (.49770*DR.GT.RN) GO TO 101	WAKE 136
	SC = 0.00	WAKE 137
	GO TO 102	WAKE 138
101	SC = (.000 * DR - RN * [COS(THETAC)] / OSIN(THETAC)	WAKE 139
102	ZPLT = ZTR/FUG1	WAKE 140
		WAKE 141
		WAKE 142
		WAKE 143
	CALL FLOWF(FNET,R1,IND)	WAKE 144
	IF (IWRPNT.FO.1) WRITE(6,2000) FNET, R1	WAKE 145
	IF (IWRPNT.FO.1) WRITE(6,2001)	WAKE 146
		WAKE 147
	IF ((TOP(67) & TOP(70)).FC.0) GO TO 40	WAKE 148
	CALL INTERP(10, IWAKE,ZPLCT(LPLCT),WKALT,PHI1,PHI)	WAKE 149
	F = FRO1	WAKE 150
	TAU = TAU1	WAKE 151
	SIGMS = SIGNL1	WAKE 152
	CALL RCSEFC(R1,FACT,WL1P(LPLOT),WR1P(LPLOT),TOP(67))	WAKE 153
	IF (WR1P(LPLOT).GT.0.00) GO TO 35	WAKE 154
	WR1P(LPLOT) = 0.00	WAKE 155
	IF (ORL.FO.4) WR1P(LPLOT) = 1.00	WAKE 156
	GO TO 40	WAKE 157
		WAKE 158
35	IF (ORL.FO.3) WR1P(LPLOT) = 10.00*DLCG10(WR1P(LPLOT))	WAKE 159
		WAKE 160
		WAKE 161
		WAKE 162
40	IF ((TOP(67) & TOP(71)).FC.0) GO TO 50	WAKE 163
	CALL INTERP(10, IWAKE,ZPLCT(LPLOT),WKALT,PHI2,PHI)	WAKE 164
	F = FRO2	WAKE 165
	TAU = TAU2	WAKE 166
	SIGMS = SIGNL2	WAKE 167
	CALL RCSEFC(R1,FACT,WL2P(LPLOT),WR2P(LPLOT),TOP(68))	WAKE 168
	IF (WR2P(LPLOT).GT.0.00) GO TO 45	WAKE 169
	WR2P(LPLOT) = 0.00	WAKE 170
	IF (ORL.FO.4) WR2P(LPLOT) = 1.00	WAKE 171
	GO TO 50	WAKE 172
		WAKE 173
45	IF (ORL.FO.3) WR2P(LPLOT) = 10.00*DLCG10(WR2P(LPLOT))	WAKE 174
		WAKE 175
		WAKE 176
		WAKE 177
50	IF ((TOP(67) & TOP(72)).FC.0) GO TO 60	WAKE 178
	CALL INTERP(10, IWAKE,ZPLCT(LPLOT),WKALT,PHI3,PHI)	WAKE 179
	F = FRO3	WAKE 180
	TAU = TAU3	WAKE 181

SIGMAS = SIGNL3	WAKE 180
CALL 20500(R1,FNET,WL3P(LPLOT),WR3P(LPLOT),TOP(69))	WAKE 181
IF(WR3P(LPLOT)GT,0.00) GO TO 182	WAKE 182
WR3P(LPLOT) = 0.00	WAKE 183
IF(TOP(69.4)-WR3P(LPLOT) = 1.00	WAKE 184
GO TO 180	WAKE 185
IF(TOP(69.3) WR3P(LPLOT)=10.00*LOG10(WR3P(LPLOT))	WAKE 186
	WAKE 187
	WAKE 188
RETURN	WAKE 189
	WAKE 190
	WAKE 191
2000 FORMAT(1H0 11HWAKE FNET = 1PF14.7; 1X4HP1 = E15.7)	WAKE 192
1000 FORMAT(1H03X4HA(105)8X4HA(106)8X4HA(107)8X6HA(108)8X6HA(109)8X6HA(WAKE 193
1110)8X4HA(111)8X4HA(112)10X4HRCF/1H 1PPE14.7/1H 11X3HZT11X3HZNXWAKE	WAKE 194
2 13X1H010X4HRNMXXCFHTHETA10X4HCAF12X2HCO11X3HCO11X3HCCR/1H	WAKE 195
3 1PPE14.7/1H 11X3HCO112X2HPE10X4HP1NF10X4HMINF9X6HPCIN112X1HV10X	WAKE 196
1 4HTINF0X5HMHTNF12X2HME/1H 1PPE14.7/1H 10X4HRHOF11X3HCFE12X2HTE	WAKE 197
2 12X2HVF 10X4HWCT1CX4HX9A1CX4HXLCW/1H 1PPE14.7)	WAKE 198
END	WAKE 199

CIBLE



A

C	WRITEM	WRITE	1
C	WRITEM PROGRAM 25420	WRITE	2
C	IMPLICIT REAL*(A-H,O-Z)	WRITE	3
C	SUBROUTINE WRITEM(X,Z,T,Q,R,S,DA,DB, SLOPE,N)	WRITE	4
C		WRITE	5
C		WRITE	6
C	EACH TIME SUBROUTINE WRITEM IS CALLED, A SINGLE LINE MAY BE	WRITE	7
C	PRINTED IN THE CORRIDOR INTEGRAL TABLE.	WRITE	8
C		WRITE	9
C		WRITE	10
C	IF(N.EQ.0) RETURN	WRITE	11
C	IX = IX/10	WRITE	12
C	GO TO (10, 20, 30, 40, 50, 60), IX	WRITE	13
10	WRITE(6,1000)Z,T,Q,R,S,DA	WRITE	14
	RETURN	WRITE	15
20	WRITE(6,1100)Z,T,Q,R,S,DA,DB	WRITE	16
	RETURN	WRITE	17
30	WRITE(6,1400)Z,T,Q,R,S,DA,DB	WRITE	18
	RETURN	WRITE	19
40	WRITE(6,1300)Z,T,Q,R,S,DA,SLOPE	WRITE	20
	RETURN	WRITE	21
50	WRITE(6,1500)Z,T,Q,R,S,DA,DB,SLOPE	WRITE	22
	RETURN	WRITE	23
60	WRITE(6,1400)Z,T,Q,R,S,DA,DB,SLOPE	WRITE	24
	RETURN	WRITE	25
1000	FORMAT(1H F14.1,/F14.2,1PF14.7,2F14.1,1PF14.7)	WRITE	26
1100	FORMAT(1H F14.1,/F14.2,1PF14.7,14XDPF14.1)	WRITE	27
1300	FORMAT(1H F14.1,/F14.2,1PF14.7,2RX1PF14.7)	WRITE	28
1400	FORMAT(1H F14.1,/F14.2,1PF14.7,DPF14.1,14X1PF14.7)	WRITE	29
1500	FORMAT(1H F14.1,/F14.2,1PF14.7,14X,DPF14.1,1PF14.7)	WRITE	30
	END	WRITE	31

NOT REPRODUCIBLE

C/ZPRM		ZPRM	1
IMPLICIT REAL*8 (A-H,O-Z)		ZPRM	2
SUBROUTINE ZPRM		ZPRM	3
COMMON /IGDHL/ IGDH1(20), TGDH1(20)		ZPRM	4
COMMON /IXCOM/ XCOM(200), ICOM(200)		ZPRM	5
COMMON /SENSE/ ISFN1, ISFN2		ZPRM	6
COMMON /MIN/ALOW(20), UP(20), AMULT(20), CALOW(20), CTP(20)		ZPRM	7
COMMON /IOPT/IPROC, IN, NCONS, IPNT, IFX, LIMIT, IRAND		ZPRM	8
COMMON /EORT/ERR, PRAND, FAC, DELTA		ZPRM	9
COMMON /QWL/QVECT(20), WRF, LREQ		ZPRM	10
COMMON /NALTQ/EGSN, NALT		ZPRM	11
COMMON /DOPT/DELX(20)		ZPRM	12
COMMON /MULT/ SMULT(25)		ZPRM	13
EQUIVALENCE (XCOM(1), ALPHA)		ZPRM	14
EQUIVALENCE (XCOM(2), BETA)		ZPRM	15
EQUIVALENCE (XCOM(3), GAMMA)		ZPRM	16
EQUIVALENCE (XCOM(4), DEL1)		ZPRM	17
EQUIVALENCE (XCOM(5), RATL)		ZPRM	18
EQUIVALENCE (XCOM(6), TOL)		ZPRM	19
C		ZPRM	20
C	SUBROUTINE ZPRM PRESETS THE VALUES OF SOME OF THE VARIABLES	ZPRM	21
C	USED IN THE PROGRAM.	ZPRM	22
C		ZPRM	23
C		ZPRM	24
C		ZPRM	25
DO 16 I = 1, 200		ZPRM	26
ICOM(I) = 0		ZPRM	27
16 CONTINUE		ZPRM	28
C		ZPRM	29
DO 11 I = 1, 20		ZPRM	30
IGDHI(I) = 0		ZPRM	31
IGDHT(I) = 0		ZPRM	32
ALOW(I) = 0.00		ZPRM	33
AMULT(I) = 1.00		ZPRM	34
CALOW(I) = 0.00		ZPRM	35
CTP(I) = 0.00		ZPRM	36
DELX(I) = .00100		ZPRM	37
QVECT(I) = -5.00		ZPRM	38
UP(I) = 0.00		ZPRM	39
11 CONTINUE		ZPRM	40
C		ZPRM	41
DO 22 I = 1, 25		ZPRM	42
SMULT(I) = 1.00		ZPRM	43
22 CONTINUE		ZPRM	44
C		ZPRM	45
ERR = 0.0100		ZPRM	46
EGSN = 4.00		ZPRM	47
IFX = 2		ZPRM	48
IN = 1		ZPRM	49
IPROC = 1		ZPRM	50
IRAND = 0		ZPRM	51
ISFN1 = 0		ZPRM	52
ISFN2 = 0		ZPRM	53
LIMIT = 30		ZPRM	54
NALT = 0		ZPRM	55
NCONS = 1		ZPRM	56
PRAND = 0.00		ZPRM	57
WRF = .000		ZPRM	58
XCOM(1) = 3.00		ZPRM	59

XCOM(2) = .500
XCOM(3) = .500
XCOM(4) = .0100
XCOM(5) = .500
XCOM(6) = 1.000

ZPRN 60
ZPRN 61
ZPRN 62
ZPRN 63
ZPRN 64
ZPRN 65
ZPRN 66
ZPRN 67

RETURN
END

ZPRS		ZPRS	1
	SUBROUTINE ZPRS(A,7)	ZPRS	2
	IMPLICIT REAL*8(A-H,O-Z)	ZPRS	3
	DIMENSION A(21), R(21)	ZPRS	4
		ZPRS	5
		ZPRS	6
	SUBROUTINE ZPRS DEFINES THE ELEMENTS OF THE A AND B ARRAYS	ZPRS	7
	WHICH ARE COEFFICIENTS OF THE CURVE FITS CONTAINED IN VARIOUS	ZPRS	8
	OTHER SUBROUTINES.	ZPRS	9
		ZPRS	10
		ZPRS	11
		ZPRS	12
	COEFFICIENTS FOR THE PROBABILITY DISTRIBUTION BETWEEN FREE	ZPRS	13
	MOLECULE AND CONTINUUM FLOW REGIMES, USED IN SUBROUTINE CRAGCO	ZPRS	14
	R(01)=0.9309200	ZPRS	15
	R(02)=0.1184700	ZPRS	16
	R(03)=-0.222400-1	ZPRS	17
	R(04)=-0.256340-2	ZPRS	18
	R(05)=-0.303250-2	ZPRS	19
	R(06)=-0.271750-2	ZPRS	20
	R(07)=0.421630-2	ZPRS	21
	R(08)=0.303650-3	ZPRS	22
	R(09)=-0.134120-2	ZPRS	23
	R(10)=-0.737200-4	ZPRS	24
	R(11)=0.221050-3	ZPRS	25
	R(12)=-0.104390-5	ZPRS	26
	R(13)=-0.213730-4	ZPRS	27
	R(14)=0.996020-4	ZPRS	28
	R(15)=0.123410-5	ZPRS	29
	R(16)=-0.781430-7	ZPRS	30
	R(17)=-0.404310-7	ZPRS	31
	R(18)=0.777600-8	ZPRS	32
	R(19)=0.643060-5	ZPRS	33
	R(20)=-0.434740-10	ZPRS	34
	R(21)=-0.295370-11	ZPRS	35
		ZPRS	36
	COEFFICIENTS USED IN FINDING MAXIMUMS AND MINIMUMS OF ALPHA	ZPRS	37
	IN SUBROUTINE VIXEN.	ZPRS	38
	A(11)=4.0669170-01	ZPRS	39
	A(12)=1.0929900-01	ZPRS	40
	A(13)=7.5266630-02	ZPRS	41
	A(14)=9.9279700-02	ZPRS	42
	A(15)=-1.8320020-02	ZPRS	43
	A(16)=-0.1525070-03	ZPRS	44
	A(17)=-4.3270470-03	ZPRS	45
	A(18)=-0.0990920-04	ZPRS	46
	A(19)=2.0097070-04	ZPRS	47
	A(20)=-1.4334700001	ZPRS	48
	A(21)=-4.5001000000	ZPRS	49
	A(22)=-1.0704100-01	ZPRS	50
	A(23)=-2.0774080000	ZPRS	51
	A(24)=-0.2144320-01	ZPRS	52
	A(25)=-5.0619010-02	ZPRS	53
	A(26)=-1.2946220-01	ZPRS	54
	A(27)=-3.0747740-02	ZPRS	55
	A(28)=-7.0047100-02	ZPRS	56
	A(29)=-5.3514900001	ZPRS	57
	A(30)=-1.5175500000	ZPRS	58
	A(31)=-2.1007020-01	ZPRS	59

A(32)=9.747159000	ZPRS 60
A(33)=-2.77304700	ZPRS 61
A(34)=9.1575960-02	ZPRS 62
A(35)=-4.2004070-01	ZPRS 63
A(36)=1.0571490-01	ZPRS 64
A(37)=-3.7212170-3	ZPRS 65

COEFFICIENTS USED IN CALCULATING BLUNT CONE PRESSURE DISTRIBUTIONS FOR CONE HALF ANGLES GREATER THAN OR EQUAL TO 20 DEGREES IN AERODY.

A(41)=1.491575270-1	ZPRS 66
A(42)=-9.684050500-1	ZPRS 67
A(43)=1.795244590-1	ZPRS 68
A(44)=-3.441965900	ZPRS 69
A(45)=-5.0979922300	ZPRS 70
A(46)=8.092023390-1	ZPRS 71
A(47)=1.851701600-1	ZPRS 72
A(48)=1.231025550	ZPRS 73
A(49)=-1.061593760-1	ZPRS 74
A(50)=3.9213215300	ZPRS 75
A(51)=5.7309227900	ZPRS 76
A(52)=-9.009915720-01	ZPRS 77
A(53)=6.0447217100	ZPRS 78
A(54)=-4.406655010-1	ZPRS 79
A(55)=5.363261360	ZPRS 80
A(56)=-1.0490477500	ZPRS 81
A(57)=-1.508684400	ZPRS 82
A(58)=2.497980700-1	ZPRS 83

COEFFICIENTS USED IN CALCULATING EDGE TEMPERATURE IN PRELIM.

A(90)=2.30983700	ZPRS 84
A(91)=1.62701700	ZPRS 85
A(92)=-2.3562520-01	ZPRS 86
A(93)=2.4184420-02	ZPRS 87
A(94)=-7.1430150-04	ZPRS 88
A(95)=-6.1245330002	ZPRS 89
A(96)=9.091090050-01	ZPRS 90
A(97)=-1.67294200	ZPRS 91
A(98)=2.5912010-01	ZPRS 92
A(99)=61.2414150-02	ZPRS 93
A(100)=2.1759430003	ZPRS 94
A(101)=2.6479300802	ZPRS 95
A(102)=-5.493559700-01	ZPRS 96
A(103)=6.16434200	ZPRS 97
A(104)=-1.8469720-01	ZPRS 98

COEFFICIENTS USED IN CALCULATING SPECIFIC HEATS AT CONSTANT PRESSURE WHEN TEMPERATURE IS IN THE RANGE 700 TO 5000 DEGREES RANKINE INCLUSIVE. USED IN SUBROUTINE PRELIM.

A(105)=0.213100	ZPRS 99
A(106)=0.457300-04	ZPRS 100
A(107)=0.772290-09	ZPRS 101
A(108)=0.478690-12	ZPRS 102
A(109)=0.000	ZPRS 103
A(110)=0.000	ZPRS 104

COEFFICIENTS USED IN CALCULATING SPECIFIC HEATS AT CONSTANT PRESSURE WHEN TEMPERATURE IS GREATER THAN 5000 DEGREES RANKINE. USED IN SUBROUTINE PRELIM.

A(111)=-0.042200	ZPRS 105
------------------	----------

NOT REPRODUCIBLE

A(112)=7.00 05

ZPRS 120

ZPRS 121

COEFFICIENTS USED IN BLUNT CONE PRESSURE DISTRIBUTIONS FOR CONE HALF ANGLE LESS THAN 20 DEGREES IN AREA.

ZPRS 122

ZPRS 123

A(135)=3.747663901

ZPRS 124

A(136)=-1.592449401

ZPRS 125

A(137)=2.1399929200

ZPRS 126

A(138)=4.5075261400

ZPRS 127

A(139)=-3.1997916800

ZPRS 128

A(140)=1.1761663000

ZPRS 129

A(141)=-4.0249657500

ZPRS 130

A(142)=1.3371717900

ZPRS 131

A(143)=-2.267489140-1

ZPRS 132

A(144)=-4.7419249301

ZPRS 133

A(145)=1.9959879401

ZPRS 134

A(146)=-2.7538929300

ZPRS 135

A(147)=-8.2600169500

ZPRS 136

A(148)=5.0601641500

ZPRS 137

A(149)=-1.4893057400

ZPRS 138

A(150)=5.98009064500

ZPRS 139

A(151)=-1.7966908500

ZPRS 140

A(152)=2.562378350-1

ZPRS 141

A(153)=1.4426872001

ZPRS 142

A(154)=-6.1270301300

ZPRS 143

A(155)=9.736578810-1

ZPRS 144

A(156)=3.3902952900

ZPRS 145

A(157)=-1.9090029200

ZPRS 146

A(158)=4.719003870-1

ZPRS 147

A(159)=-1.8490124300

ZPRS 148

A(160)=5.695376290-1

ZPRS 149

A(161)=-6.904477490-2

ZPRS 150

COEFFICIENTS USED IN CALCULATING TSTAR IN SUBROUTINE DRAGCC.

ZPRS 151

A(162)= 1.5276955073

ZPRS 152

A(163)= 2.1399929200

ZPRS 153

A(164)=-2.071067630-4

ZPRS 154

A(165)= 5.521092210-9

ZPRS 155

A(166)=-1.3149173703

ZPRS 156

A(167)= 2.2819449200

ZPRS 157

A(168)=-0.953427970-4

ZPRS 158

A(169)= 1.364804780-7

ZPRS 159

A(170)= 1.0995150502

ZPRS 160

A(171)=-1.013401660-1

ZPRS 161

A(172)= 9.648595920-5

ZPRS 162

A(173)=-1.207949180-8

ZPRS 163

ZPRS 164

COEFFICIENTS USED IN CALCULATING THE FREE MOLECULE DRAG COEFFICIENT ON THE SPHERICAL NOSE, USED IN SUBROUTINE DRAGCC.

ZPRS 165

A(174)= 0.9736600

ZPRS 166

A(175)=-0.907320-1

ZPRS 167

A(176)=-0.406000-1

ZPRS 168

A(177)= 0.124270-2

ZPRS 169

A(178)= 0.118630-1

ZPRS 170

A(179)= 0.904750-3

ZPRS 171

A(180)=-0.444910-2

ZPRS 172

A(181)=-0.37470-3

ZPRS 173

A(182)=-0.102400-2

ZPRS 174

A(183)= 0.512530-4

ZPRS 175

A(184)=-0.127220-3

ZPRS 176

A(185)=-0.101410-5

ZPRS 177

ZPRS 178

ZPRS 179

NOT REPRODUCIBLE

NOT REPRODUCIBLE

A(186)= 0.112210-4	ZPRS 180
A(187)= -0.395070-6	ZPRS 181
A(188)= -0.526920-6	ZPRS 182
A(189)= 0.374570-7	ZPRS 183
A(190)= 0.127710-7	ZPRS 184
A(191)= -0.125790-8	ZPRS 185
A(192)= -0.110650-9	ZPRS 186
A(193)= 0.140140-10	ZPRS 187

C COEFFICIENTS USED IN CALCULATING THE STRONG INTERACTION
C DRAG COEFFICIENT FOR CONE HALF ANGLE LESS THAN 15 DEGREES.

USED IN SUBROUTINE DRAGCO.	ZPRS 189
A(200)= -3.90021000	ZPRS 189
A(201)= 0.539100200	ZPRS 190
A(202)= 0.17677000	ZPRS 191
A(203)= 0.11997650-01	ZPRS 192
A(204)= 0.551859500	ZPRS 193
A(205)= 0.515930100	ZPRS 194
A(206)= 3.6094110-04	ZPRS 195
A(207)= -0.68253270-01	ZPRS 196

C COEFFICIENTS USED IN DETERMINING THE BLUNT CONE FORCEPCOY
C PRESSURE DRAG COEFFICIENT FOR CONE HALF ANGLE 4 TO 10 DEGREES
C INCLUSIVE. USED IN SUBROUTINE DRAGCO.

A(211)= 2.859586240-1	ZPRS 199
A(212)= -1.341971130-1	ZPRS 200
A(213)= -3.234465650-1	ZPRS 201
A(214)= -6.615724260-1	ZPRS 202
A(215)= -1.1690290100	ZPRS 203
A(216)= 1.6906154100	ZPRS 204
A(217)= 1.800336100	ZPRS 205
A(218)= -2.6053467300	ZPRS 206
A(219)= -2.5667407400	ZPRS 207
A(220)= -3.5693417700	ZPRS 208
A(221)= 6.044779220-1	ZPRS 209
A(222)= -3.0279349200	ZPRS 210
A(223)= -1.9005512700	ZPRS 211
A(224)= -3.3763703200	ZPRS 212
A(225)= -7.40644519061	ZPRS 213
A(226)= -1.61023831061	ZPRS 214
A(227)= 2.32733899061	ZPRS 215
A(228)= 1.05707834062	ZPRS 216
A(229)= 1.60740777061	ZPRS 217
A(230)= 7.2199503000	ZPRS 218
A(231)= -2.93557270061	ZPRS 219
A(232)= -6.47577211061	ZPRS 220
A(233)= -8.06730866061	ZPRS 221
A(234)= 7.41492409062	ZPRS 222
A(235)= 9.57017970061	ZPRS 223
A(236)= 2.26649809061	ZPRS 224
A(237)= -1.13698627063	ZPRS 225
A(238)= -2.45544587061	ZPRS 226
A(239)= -2.97182588061	ZPRS 227
A(240)= 1.1042267062	ZPRS 228
A(241)= 7.81429820061	ZPRS 229
A(242)= 2.8903903062	ZPRS 230
A(243)= -1.8107909063	ZPRS 231
A(244)= -1.89524636062	ZPRS 232
A(245)= -2.56443251062	ZPRS 233
A(246)= 2.9143899063	ZPRS 234

NOT REPRODUCIBLE

COEFFICIENTS USED IN DETERMINING THE BLUNT CONE FORM PRESSURE DRAG COEFFICIENT FOR CONE HALF-ANGLE GREATER THAN 10 AND LESS THAN OR EQUAL TO 20 DEGREES. USED IN SUBROUTINE DR		ZPRS 240
A(247) = 1.2343642400		ZPRS 241
A(248) = -9.608206510-2		ZPRS 242
A(249) = 4.941718560-1		ZPRS 243
A(250) = -4.11772110-1		ZPRS 244
A(251) = 4.8772495600		ZPRS 245
A(252) = -1.63991921061		ZPRS 246
A(253) = -1.961475610-1		ZPRS 247
A(254) = -4.4532661900		ZPRS 248
A(255) = 2.73652736061		ZPRS 249
A(256) = -3.46601734061		ZPRS 250
A(257) = 1.17239966061		ZPRS 251
A(258) = -1.67160696061		ZPRS 252
A(259) = 1.36242932061		ZPRS 253
A(260) = -2.95101207062		ZPRS 254
A(261) = 8.22067681062		ZPRS 255
A(262) = 2.13249139061		ZPRS 256
A(263) = 3.90719615062		ZPRS 257
A(264) = -1.60066305063		ZPRS 258
A(265) = 3.62270199062		ZPRS 259
A(266) = -2.4091524062		ZPRS 260
A(267) = 3.67997692062		ZPRS 261
A(268) = -2.30076632062		ZPRS 262
A(269) = 4.19917135063		ZPRS 263
A(270) = -1.24624267064		ZPRS 264
A(271) = -3.11638604061		ZPRS 265
A(272) = 5.87092896063		ZPRS 266
A(273) = -2.31096696064		ZPRS 267
A(274) = -1.32594998063		ZPRS 268
A(275) = 1.31928307063		ZPRS 269
A(276) = -2.10638114063		ZPRS 270
A(277) = 1.21038396063		ZPRS 271
A(278) = -2.09572913064		ZPRS 272
A(279) = 6.10107009064		ZPRS 273
A(280) = -5.64878622062		ZPRS 274
A(281) = 2.99870767064		ZPRS 275
A(282) = -1.11902873065		ZPRS 276
		ZPRS 277
		ZPRS 278
		ZPRS 279
		ZPRS 280
		ZPRS 281
COEFFICIENTS USED IN DETERMINING THE SHAPE FACTOR FOR CALCULATING LAMINAR-INDUCED DRAG COEFFICIENT. USED IN SUB- ROUTINE DRAGCN.		ZPRS 282
A(283) = 2.2271721100		ZPRS 283
A(284) = 1.332292200-1		ZPRS 284
A(285) = 3.099593000-1		ZPRS 285
A(286) = 1.892494320-1		ZPRS 286
A(287) = -1.8975237900		ZPRS 287
A(288) = 1.924158240-1		ZPRS 288
A(289) = 7.054577870-1		ZPRS 289
A(290) = -1.132074020-1		ZPRS 290
		ZPRS 291
		ZPRS 292
COEFFICIENTS USED IN CALCULATING THE TRANSITION ALTITUDE. USED IN SUBROUTINE-PRFLIN.		ZPRS 293
A(300) = 7.4922926604		ZPRS 294
A(301) = 9.9402970209		ZPRS 295
A(302) = 3.0974220602		ZPRS 296
A(303) = 5.9957280603		ZPRS 297
A(304) = 9.1030666601		ZPRS 298
		ZPRS 299

NOT REPRODUCIBLE

A(305) = -0.89478800	ZPRS 300
A(304) = 0.265414000	ZPRS 301
A(307) = 1.207805000	ZPRS 302
A(309) = 2.142371000	ZPRS 303
A(306) = 1.004362000	ZPRS 304
A(310) = 4.071300000	ZPRS 305
A(311) = 2.13041200	ZPRS 306

COEFFICIENTS USED IN DETERMINING THE BLUNT CONE CORRECTED PRESSURE DRAG COEFFICIENT FOR CONE HALF ANGLE GREATER THAN 20 AND LESS THAN OR EQUAL TO 40 DEGREES. USED IN SUPRCUTINE DRAGCO.

A(312) = 3.897512000	ZPRS 307
A(313) = -5.9188305100	ZPRS 308
A(314) = 2.7330701001	ZPRS 309
A(315) = -3.740113300	ZPRS 310
A(316) = -3.705545001	ZPRS 311
A(317) = -1.2134095002	ZPRS 312
A(318) = 1.05320173001	ZPRS 313
A(319) = -5.0525534001	ZPRS 314
A(320) = 1.50358725002	ZPRS 315
A(321) = -2.00301090052	ZPRS 316
A(322) = 4.37880682002	ZPRS 317
A(323) = -2.04755317003	ZPRS 318
A(324) = -3.37001615002	ZPRS 319
A(325) = -3.00007525003	ZPRS 320
A(326) = -0.70024580003	ZPRS 321
A(327) = -0.55067071002	ZPRS 322
A(328) = 4.93534830003	ZPRS 323
A(329) = -1.27345990004	ZPRS 324
A(330) = 3.92300070003	ZPRS 325
A(331) = -1.02140510004	ZPRS 326
A(332) = 4.95992110004	ZPRS 327
A(333) = -0.60765670003	ZPRS 328
A(334) = 2.72077200006	ZPRS 329
A(335) = -2.50134187005	ZPRS 330
A(336) = 2.76036121006	ZPRS 331
A(337) = -1.31876600005	ZPRS 332
A(338) = 3.40078040005	ZPRS 333
A(339) = -2.55921000006	ZPRS 334
A(340) = 2.88990500006	ZPRS 335
A(341) = -3.90810140005	ZPRS 336
A(342) = 0.81036940006	ZPRS 337
A(343) = -4.4281500005	ZPRS 338
A(344) = 2.07090520006	ZPRS 339
A(345) = -2.40591870005	ZPRS 340
A(346) = 1.13005330006	ZPRS 341
A(347) = -2.93180370006	ZPRS 342

NOT REPRODUCIBLE

COEFFICIENTS USED IN DETERMINING THE RATIO COP WITH ANGLE OF ATTACK EFFECTS/ COP FOR ANGLE OF ATTACK OF ZERO, WHERE THE ABSOLUTE VALUE OF THE ANGLE OF ATTACK IS GREATER THAN 4 AND LESS THAN OR EQUAL TO 40 DEGREES. IF THE ABSOLUTE VALUE OF ALPHA IS GREATER THAN 40 DEGREES THEN ALPHA IS SET EQUAL TO 40. USED IN SUPRCUTINE DRAGCO.

A(370) = -0.074061000-1	ZPRS 343
A(371) = 3.470636300-1	ZPRS 344
A(372) = 1.2475266300	ZPRS 345
A(373) = -6.740073170-1	ZPRS 346
A(374) = 6.147053000	ZPRS 347

NOT REPRODUCIBLE

A(353)	= -4.24159222D0	ZPRS 360
A(354)	= 1.0010310C	ZPRS 361
A(355)	= 7.20614788D-1	ZPRS 362
A(356)	= -3.21918340D-1	ZPRS 363
A(357)	= 1.60792640D0	ZPRS 364
A(358)	= -1.37411127D-1	ZPRS 365
A(359)	= -4.22070312D-1	ZPRS 366
A(360)	= 1.52746409C	ZPRS 367
A(361)	= 9.94206949C	ZPRS 368
A(362)	= -1.50244052D61	ZPRS 369
A(363)	= 5.06278106C	ZPRS 370
A(364)	= -1.94005005D61	ZPRS 371
A(365)	= 1.00926434D61	ZPRS 372
A(366)	= 1.10434749D61	ZPRS 373
A(367)	= -6.27928099C	ZPRS 374
A(368)	= 1.76307351C	ZPRS 375
A(369)	= -1.17943220C	ZPRS 376
A(370)	= -1.89185220C	ZPRS 377
A(371)	= 4.64702270D	ZPRS 378
A(372)	= 3.93474807C	ZPRS 379
A(373)	= -2.41764070C	ZPRS 380
A(374)	= 2.49771050C	ZPRS 381
A(375)	= -9.31610990D	ZPRS 382
A(376)	= 1.144103227C	ZPRS 383
A(377)	= 1.41876899D61	ZPRS 384
A(378)	= -3.36511599D61	ZPRS 385
A(379)	= 1.27079372D61	ZPRS 386
A(380)	= 4.51823102C	ZPRS 387
A(381)	= -3.44675942D61	ZPRS 388
A(382)	= 4.24088505D61	ZPRS 389
A(383)	= -1.34577750C	ZPRS 390

COEFFICIENTS USED IN CALCULATING THE STRONG INTERACTION DRAG COEFFICIENT FOR CONF HALF ANGLE GREATER THAN OR EQUAL TO 15 DEGREES, USED IN SUBROUTINE DRAGCO.

A(394)	= -1.11714455D1	ZPRS 392
A(395)	= 6.03330410D0	ZPRS 393
A(396)	= 1.25409017D0	ZPRS 394
A(397)	= -4.94133544D-1	ZPRS 395
A(398)	= -4.87044288D-2	ZPRS 396
A(399)	= 2.75931122D-2	ZPRS 397
A(400)	= 6.48529951D-4	ZPRS 398
A(401)	= -3.73062274D-4	ZPRS 399
A(402)	= 4.97741854D0	ZPRS 400
A(403)	= 2.40489449D0	ZPRS 401
A(404)	= 5.85917872D-1	ZPRS 402
A(405)	= 3.54374721D-1	ZPRS 403
A(406)	= 2.14400680D-2	ZPRS 404
A(407)	= -1.39491829D-2	ZPRS 405
A(408)	= -2.93983230D-4	ZPRS 406
A(409)	= 1.82042047D-4	ZPRS 407

COEFFICIENTS USED TO DETERMINE COEFFICIENTS A(118), A(119), A(120) USED IN SUBROUTINE DRAGCO TO CALCULATE TURBULENT SKIN FRICTION DRAG COEFFICIENT.

A(401)	= 0.70347602	ZPRS 411
A(402)	= 0.761990602	ZPRS 412
A(403)	= 0.729000601	ZPRS 413
A(404)	= 0.120450601	ZPRS 414
A(405)	= 0.102400	ZPRS 415

A(405)=0.511200-07	ZPRS 420
A(406)=-0.063770-04	ZPRS 421
A(407)=0.57456000	ZPRS 422
A(408)=-0.39260000	ZPRS 423
A(409)=0.11991000	ZPRS 424
A(410)=-0.19641000	ZPRS 425
A(411)=0.17040000	ZPRS 426
A(412)=-0.039400-01	ZPRS 427
A(413)=0.150120-02	ZPRS 428
A(414)=-0.20570000	ZPRS 429
A(415)=0.12001000	ZPRS 430
A(416)=-0.37451000	ZPRS 431
A(417)=0.61556000	ZPRS 432
A(418)=-0.56271000	ZPRS 433
A(419)=0.26611000	ZPRS 434
A(420)=-0.504520-02	ZPRS 435

COEFFICIENTS USED IN DETERMINING THE RATIO COP WITH ANGLE OF ATTACK EFFECTS/ COP FOR ANGLE OF ATTACK OF ZERO, WHERE THE ABSOLUTE VALUE OF ALPHA IS LESS THAN 4 DEGREES AND GREATER THAN ZERO, USED IN SUBCUTINE DRAG.

A(421)= 2.79597100-1	ZPRS 436
A(422)= 4.85647000-3	ZPRS 437
A(423)= -2.70951530-4	ZPRS 438
A(424)= 4.32957030-6	ZPRS 439
A(425)= 6.52190740-1	ZPRS 440
A(426)= -2.01116700-1	ZPRS 441
A(427)= 1.40935400-2	ZPRS 442
A(428)= -4.21256040-6	ZPRS 443
A(429)= -3.64419040-2	ZPRS 444
A(430)= 3.40956460-2	ZPRS 445
A(431)= -3.47499910-3	ZPRS 446
A(432)= 9.80119330-4	ZPRS 447
A(433)= 1.49927440-1	ZPRS 448
A(434)= -3.41042710-2	ZPRS 449
A(435)= 1.93234610-3	ZPRS 450
A(436)= -2.73140020-4	ZPRS 451
A(437)= -6.7486494100	ZPRS 452
A(438)= 2.0042417000	ZPRS 453
A(439)= -1.67252260-1	ZPRS 454
A(440)= 4.14679300-1	ZPRS 455
A(441)= 1.0731207000	ZPRS 456
A(442)= -4.21300010-1	ZPRS 457
A(443)= 3.97073450-2	ZPRS 458
A(444)= -1.00502270-1	ZPRS 459
A(445)= -2.64405050-1	ZPRS 460
A(446)= 4.70237030-2	ZPRS 461
A(447)= -2.44114930-3	ZPRS 462
A(448)= 3.56147310-4	ZPRS 463
A(449)= 1.57097197061	ZPRS 464
A(450)= -4.962948300	ZPRS 465
A(451)= 4.141428300-1	ZPRS 466
A(452)= -1.03761900-2	ZPRS 467
A(453)= -3.4790108000	ZPRS 468
A(454)= 1.1450007000	ZPRS 469
A(455)= -0.077100400-2	ZPRS 470
A(456)= 2.54002000-3	ZPRS 471

COEFFICIENTS USED IN DETERMINING THE BLUNTNES CORRECTION TO THE SKIN FRICTION DRAG COEFFICIENT IN SUBCUTINE DRAG.

NOT REPRODUCIBLE

FOR A BLUNTNESS RATIO GREATER THAN OR EQUAL TO 0.2.		ZPRS 480
A(457)	= 9.096609810-C1	ZPRS 481
A(458)	= 1.339938840-C0	ZPRS 482
A(459)	= -3.392472710-C0	ZPRS 483
A(460)	= 3.502531890-C0	ZPRS 484
A(461)	= -1.386618700-C1	ZPRS 485
A(462)	= 5.588019230-C0	ZPRS 486
A(463)	= -6.511434800-C0	ZPRS 487
A(464)	= 2.477340290-C0	ZPRS 488
A(465)	= -3.643502700-C1	ZPRS 489
A(466)	= 1.699732300-C2	ZPRS 490
A(467)	= -7.771402270-C1	ZPRS 491
A(468)	= 5.845058840-C1	ZPRS 492
A(469)	= -2.113237540-C1	ZPRS 493
A(470)	= 2.634821130-C0	ZPRS 494
A(471)	= -1.112296100-C1	ZPRS 495
FOR A BLUNTNESS RATIO LESS THAN 0.2.		ZPRS 496
A(472)	= 2.140180770-C0	ZPRS 497
A(473)	= -1.129645760-C0	ZPRS 498
A(474)	= 3.808566520-C1	ZPRS 499
A(475)	= -4.068843800-C2	ZPRS 500
A(476)	= 2.212652750-C3	ZPRS 501
A(477)	= -1.071073530-C1	ZPRS 502
A(478)	= 1.619308430-C1	ZPRS 503
A(479)	= -4.604561490-C0	ZPRS 504
A(480)	= 5.233621380-C1	ZPRS 505
A(481)	= -2.053205500-C2	ZPRS 506
A(482)	= 2.985043370-C1	ZPRS 507
A(483)	= -2.311356280-C1	ZPRS 508
A(484)	= 6.197134860-C0	ZPRS 509
A(485)	= -4.754137960-C1	ZPRS 510
A(486)	= 2.582076600-C2	ZPRS 511
COEFFICIENTS USED IN CALCULATING BASE DRAG COEFFICIENT IN		ZPRS 512
SUBROUTINE DRAGCO.		ZPRS 513
FOR FREE STREAM MACH NUMBER LESS THAN 7.		ZPRS 514
A(487)	= 3.075621510-C1	ZPRS 515
A(488)	= -4.112725760	ZPRS 516
A(489)	= -2.073955690-C1	ZPRS 517
A(490)	= -2.319474250-C1	ZPRS 518
A(491)	= 2.545262250-C1	ZPRS 519
A(492)	= -3.439138940-C2	ZPRS 520
A(493)	= 1.932662420-C3	ZPRS 521
A(494)	= -3.139044760-C3	ZPRS 522
A(495)	= -3.780488240-C2	ZPRS 523
A(496)	= 4.922843500-C3	ZPRS 524
A(497)	= -2.074094340-C1	ZPRS 525
A(498)	= 5.00025830-C4	ZPRS 526
FOR FREE STREAM MACH NUMBER GREATER THAN OR EQUAL TO 7.		ZPRS 527
A(499)	= 8.237055200-C1	ZPRS 528
A(500)	= -2.294006380-C1	ZPRS 529
A(501)	= 2.247472240-C2	ZPRS 530
A(502)	= -7.114447640-C2	ZPRS 531
A(503)	= 5.87333960-C1	ZPRS 532
A(504)	= -1.440142400-C3	ZPRS 533
A(505)	= 1.596231130-C4	ZPRS 534
A(506)	= -4.990470230-C4	ZPRS 535
A(507)	= -4.471205200-C2	ZPRS 536
A(508)	= 1.328957140-C4	ZPRS 537
A(509)	= 1.300434900-C5	ZPRS 538
		ZPRS 539

1(510) = 4.09475475065

LETURN

END

ZPRS 540

ZPRS 541

ZPRS 542

CZREADX	ZREAD	1
SUBROUTINE ZREADX	ZREAD	2
IMPLICIT REAL*8(A-M,O-Z)	ZREAD	3
	ZREAD	4
	ZREAD	5
SUBROUTINE ZREADX PROVIDES THE INFORMATION ABOUT THE SIZE	ZREAD	6
AND DIMENSION OF EACH INPUT VARIABLE IT REFERS TO. THIS	ZREAD	7
INFORMATION IS REQUIRED BY THE INPUT SUBROUTINE READIN, A	ZREAD	8
LIBRARY ROUTINE.	ZREAD	9
	ZREAD	10
	ZREAD	11
COMMON /CICCUR/ ICOCUR(320)	ZREAD	12
COMMON /CPCUR/ PCUR(11700)	ZREAD	13
COMMON /CPCSEC/ R2FRQ,R2,H3,HTWEN,R24,DX,ZNUS,CNE,OSB,X2PC,X3B,	ZREAD	14
LIND2,NSTWL	ZREAD	15
COMMON /NALTFG/PGSY,NALT	ZREAD	16
COMMON /NIMPIT/ AKW,PHOW,CRHCW,DELWH, TABL,CHCHEM,RHCSL,	ZREAD	17
1 RTQ,ACON,ACON, R21,R22,R23, CNUMR(169)	ZREAD	18
COMMON /IDNDS/ ID1(50), ID2(50)	ZREAD	19
COMMON /IXCON/ XCON(200), ICON(200)	ZREAD	20
COMMON /SENSE/ISEN1,ISEN2	ZREAD	21
COMMON /MIN/ALOW(20),UP(20), AMULT(20),CALOW(20),CTP(20)	ZREAD	22
COMMON /IORT/ IPRCC,IN,NCOS,IPNT,IEK,LIPIT,IRAND	ZREAD	23
COMMON /IORT/ERR,PRAND, FAC, DELTA	ZREAD	24
COMMON /IORT/DELX(20)	ZREAD	25
COMMON /TALSIZ/ FRNTLTP,IOI,ERNRTB(10),ERNUTB(8),	ZREAD	26
1 ENTARL(25,9),DSTARL(9),HSTARL(25),	ZREAD	27
2 ETARL(12,11),THTTL(11),EMGTBL(12),	ZREAD	28
3 D(11,5,4),XYZTBL(11,3)	ZREAD	29
COMMON /BLKQ/HH(1400),X(40),GXX(40),S(40),XP(40),TX(40),	ZREAD	30
IS(40),GS,GSP,GTP,GSS,GTT,RSO,FZZ,FP,FR,FC,E,PZZ,TD,RS,SL,ZZZ,	ZREAD	31
1922,AKT,PL	ZREAD	32
2DELKX,FAK,GXX(40,10)	ZREAD	33
WARNING - HLKD COMMON IS BEING SHARED BY DAVDON AND ROSBRK-GRAM	ZREAD	34
ORTIME/FPS	ZREAD	35
COMMON /MINSK/N,K, ALR(20),UB(20),XX(20),FXK	ZREAD	36
COMMON /MULT/ SMULT(25)	ZREAD	37
COMMON /IGNLY/ IGD(120), IGDH(20)	ZREAD	38
COMMON /OML/OVFC(20),MPF,LRF	ZREAD	39
COMMON /CMAKE/ PFT47,FRO1,FRO2,FRO3,PHI1,PHI2,PHI3,SIGNL1,SIGL2,	ZREAD	40
ISIGL3,TAU1,TAU2,TAU3,WKALT,WSTALT,IND,IWAKE,IWPRAT	ZREAD	41
COMMON OCCUR, NCCUR	ZREAD	42
	ZREAD	43
	ZREAD	44
DIMENSION OCCUR(4000), NCCUR(30)	ZREAD	45
DIMENSION AAC(27),ACDF(140),BCR(40),BCD(40),BCV(40)	ZREAD	46
DIMENSION BCWL1(40),BCWL2(40),BCWL3(40),BCWL(40),BCWR2(40)	ZREAD	47
DIMENSION TCWL1(40),TCWL2(40),TCWL3(40),TCWR1(40),TCWR2(40)	ZREAD	48
DIMENSION WR2P(160),WR3P(160),ZPLCT(160)	ZREAD	49
DIMENSION BETAP(10),PHI1(10),PHI2(10),PHI3(10),WKALT(10)	ZREAD	50
DIMENSION SWR3(40),TCR(40),TCO(40),TCV(40)	ZREAD	51
DIMENSION NCOMDV(50),NPV(160),SR(40),SD(40),SRS(9),SV(40)	ZREAD	52
DIMENSION SWL1(40),SWL2(40),SWL3(40),SWR1(40),SWR2(40)	ZREAD	53
DIMENSION BCWR3(40),BFT2PL(160),DWH(50),DVL(50),HI(40),IOP(40)	ZREAD	54
DIMENSION TCWR3(40),TPLCT(160),VGGPLT(140),VPLT(160)	ZREAD	55
DIMENSION WCR1(40),WCR2(160),WCR3(160),WRIP(160)	ZREAD	56
	ZREAD	57
	ZREAD	58
EQUIVALENCE (OCCUR(3043),CORA)	ZREAD	59

EQUIVALENCE (PCUR(11571), AA)	ZREAD 60
EQUIVALENCE (PCUR(105801), ACCP)	ZREAD 61
EQUIVALENCE (PCUR(05181), RCB)	ZREAD 62
EQUIVALENCE (PCUR(05101), RCD)	ZREAD 63
EQUIVALENCE (PCUR(06501), RCWL1)	ZREAD 64
EQUIVALENCE (PCUR(06521), RCWL2)	ZREAD 65
EQUIVALENCE (PCUR(06661), RCWL3)	ZREAD 66
EQUIVALENCE (PCUR(06261), RCWR1)	ZREAD 67
EQUIVALENCE (PCUR(06341), RCWR2)	ZREAD 68
EQUIVALENCE (PCUR(06421), RCWR3)	ZREAD 69
EQUIVALENCE (PCUR(06021), RCV)	ZREAD 70
EQUIVALENCE (PCUR(00641), RETAPL)	ZREAD 71
EQUIVALENCE (PCUR(11657), DVL)	ZREAD 72
EQUIVALENCE (PCUR(11607), DVH)	ZREAD 73
EQUIVALENCE (PCUR(05841), H)	ZREAD 74
EQUIVALENCE (PCUR(11757), PED)	ZREAD 75
EQUIVALENCE (PCUR(04921), SB)	ZREAD 76
EQUIVALENCE (PCUR(04781), SC)	ZREAD 77
EQUIVALENCE (PCUR(11558), SPS)	ZREAD 78
EQUIVALENCE (PCUR(00741), SV)	ZREAD 79
EQUIVALENCE (PCUR(04991), SWL1)	ZREAD 80
EQUIVALENCE (PCUR(07021), SWL2)	ZREAD 81
EQUIVALENCE (PCUR(07041), SWL3)	ZREAD 82
EQUIVALENCE (PCUR(06841), SWR1)	ZREAD 83
EQUIVALENCE (PCUR(06901), SWR2)	ZREAD 84
EQUIVALENCE (PCUR(06941), SWR3)	ZREAD 85
EQUIVALENCE (PCUR(04221), TCO)	ZREAD 86
EQUIVALENCE (PCUR(04141), TCO)	ZREAD 87
EQUIVALENCE (PCUR(04541), TCWL1)	ZREAD 88
EQUIVALENCE (PCUR(04621), TCWL2)	ZREAD 89
EQUIVALENCE (PCUR(04701), TCWL3)	ZREAD 90
EQUIVALENCE (PCUR(04301), TCWR1)	ZREAD 91
EQUIVALENCE (PCUR(04381), TCWR2)	ZREAD 92
EQUIVALENCE (PCUR(04461), TCWR3)	ZREAD 93
EQUIVALENCE (PCUR(04061), TCW)	ZREAD 94
EQUIVALENCE (PCUR(00001), TPLCT)	ZREAD 95
EQUIVALENCE (PCUR(00401), VOGPLT)	ZREAD 96
EQUIVALENCE (PCUR(00321), VPLCT)	ZREAD 97
EQUIVALENCE (PCUR(01291), WL1P)	ZREAD 98
EQUIVALENCE (PCUR(01441), WL2P)	ZREAD 99
EQUIVALENCE (PCUR(01401), WL3P)	ZREAD 100
EQUIVALENCE (PCUR(00901), WR1P)	ZREAD 101
EQUIVALENCE (PCUR(00951), WR2P)	ZREAD 102
EQUIVALENCE (PCUR(01121), WR3P)	ZREAD 103
EQUIVALENCE (PCUR(00141), ZPLCT)	ZREAD 104
EQUIVALENCE (TDCUR(00314), ZORL)	ZREAD 105
EQUIVALENCE (TDCUR(00001), ZOP)	ZREAD 106
EQUIVALENCE (TDCUR(00301), ZREF)	ZREAD 107
EQUIVALENCE (TDCUR(00302), ZPLET)	ZREAD 108
EQUIVALENCE (TDCUR(00303), ZMODE)	ZREAD 109
EQUIVALENCE (TDCUR(00001), ZNCMDV)	ZREAD 110
EQUIVALENCE (TDCUR(00304), ZNP)	ZREAD 111
EQUIVALENCE (TDCUR(00305), ZNFCOY)	ZREAD 112
EQUIVALENCE (TDCUR(00306), ZNVCH)	ZREAD 113
EQUIVALENCE (TDCUR(00007), ZNP)	ZREAD 114
EQUIVALENCE (TDCUR(00141), ZNPV)	ZREAD 115
	ZREAD 116
	ZREAD 117
	ZREAD 118
	ZREAD 119

		ZREAC120
		ZREAC121
CALL SETUP (PMAA	,P,AA,27)	ZREAC122
CALL SETUP (PACCF	,P,ACCF,140)	ZREAC123
CALL SETUP (PACCN	,P,ACCN)	ZREAC124
CALL SETUP (PAKW	,P,AKW)	ZREAC125
CALL SETUP (PAMLM	,P,ALCM,20)	ZREAC126
CALL SETUP (PAMULT	,P,AMULT,20)	ZREAC127
		ZREAC128
CALL SETUP (PBCR	,P,BCR,40)	ZREAC129
CALL SETUP (PBCD	,P,BCD,40)	ZREAC130
CALL SETUP (PBCCN	,P,BCCN)	ZREAC131
CALL SETUP (PBCV	,P,BCV,40)	ZREAC132
CALL SETUP (PBCWL1	,P,BCWL1,40)	ZREAC133
CALL SETUP (PBCWL2	,P,BCWL2,40)	ZREAC134
CALL SETUP (PBCWL3	,P,BCWL3,40)	ZREAC135
CALL SETUP (PBCWR1	,P,BCWR1,40)	ZREAC136
CALL SETUP (PBCWR2	,P,BCWR2,40)	ZREAC137
CALL SETUP (PBCWR3	,P,BCWR3,40)	ZREAC138
CALL SETUP (PBRFTAPL	,P,BRFTAPL,160)	ZREAC139
CALL SETUP (PBRFTAZ	,P,BRFTAZ,10)	ZREAC140
CALL SETUP (PBRFTFN	,P,BRFTFN)	ZREAC141
CALL SETUP (PBRFTFO	,P,BRFTFO)	ZREAC142
CALL SETUP (PBR2	,P,BR2)	ZREAC143
CALL SETUP (PBR3	,P,BR3)	ZREAC144
CALL SETUP (PBR21	,P,BR21)	ZREAC145
CALL SETUP (PBR22	,P,BR22)	ZREAC146
CALL SETUP (PBR23	,P,BR23)	ZREAC147
CALL SETUP (PBR24	,P,BR24)	ZREAC148
		ZREAC149
CALL SETUP (PBCALW	,P,BCALW,20)	ZREAC150
CALL SETUP (PCCCN	,P,CCCN)	ZREAC151
CALL SETUP (PCCNF	,P,CCNF)	ZREAC152
CALL SETUP (PCCNLR	,P,CCNLR,160)	ZREAC153
CALL SETUP (PCCPHW	,P,CCPHW)	ZREAC154
CALL SETUP (PCCPT	,P,CCPT,20)	ZREAC155
		ZREAC156
CALL SETUP (PDELTA	,P,DELTA)	ZREAC157
CALL SETUP (PDELX	,P,DELX)	ZREAC158
CALL SETUP (PDELX	,P,DELX,20)	ZREAC159
CALL SETUP (PDMCHM	,P,DMCHM)	ZREAC160
CALL SETUP (PDSR	,P,DSR)	ZREAC161
CALL SETUP (PENTARL	,P,ENTARL,11,4)	ZREAC162
CALL SETUP (PDMV	,P,DMV,50)	ZREAC163
CALL SETUP (PDMVL	,P,DMVL,50)	ZREAC164
CALL SETUP (PDMX	,P,DMX)	ZREAC165
		ZREAC166
CALL SETUP (PEMCTL	,P,EMCTL,12)	ZREAC167
CALL SETUP (PEMNTARL	,P,EMNTARL,25,9)	ZREAC168
CALL SETUP (PEMPTB	,P,EMPTB,10)	ZREAC169
CALL SETUP (PEMNTLR	,P,EMNTLR,10)	ZREAC170
CALL SETUP (PEMNTLR	,P,EMNTLR,R)	ZREAC171
CALL SETUP (PEMPP	,P,EMPP)	ZREAC172
CALL SETUP (PEMPTBL	,P,EMPTBL,12,11)	ZREAC173
		ZREAC174
CALL SETUP (PEMFC	,P,EMFC)	ZREAC175
CALL SETUP (PEMFSM	,P,EMFSM)	ZREAC176
CALL SETUP (PEMFO1	,P,EMFO1)	ZREAC177
CALL SETUP (PEMFO2	,P,EMFO2)	ZREAC178
CALL SETUP (PEMFO3	,P,EMFO3)	ZREAC179

CALL SETUP(RHM	,R,H,40)	ZREAD180
CALL SETUP(RHHH	,R,HH,1400)	ZREAD181
CALL SETUP (RINSTABL	,R,H,INSTABL,25)	ZREAD182
CALL SETUP(RHICOM	,4,ICOM,200)	ZREAD183
CALL SETUP(RHITRL	,4,ITRL)	ZREAD184
CALL SETUP(RHIDNO	,4,ITD1,50)	ZREAD185
CALL SETUP(RHIDC	,4,ITD2,50)	ZREAD186
CALL SETUP(RHIFX	,4,IFX)	ZREAD187
CALL SETUP(RHIGPH	,4,IGPH,20)	ZREAD188
CALL SETUP(RHIGDL	,4,IGDL,20)	ZREAD189
CALL SETUP(RHIN	,4,IN)	ZREAD190
CALL SETUP (RIND	,4,IND)	ZREAD191
CALL SETUP(RHIND2	,4,IND2)	ZREAD192
CALL SETUP(RHIOP	,4,IOP,20)	ZREAD193
CALL SETUP(RHIPT	,4,IPT)	ZREAD194
CALL SETUP(RHIPROC	,4,IPROC)	ZREAD195
CALL SETUP(RHIRAND	,4,IRAND)	ZREAD196
CALL SETUP(RHIFRF	,4,IRF)	ZREAD197
CALL SETUP(RHISEM1	,4,ISEM1)	ZREAD198
CALL SETUP(RHISEM2	,4,ISEM2)	ZREAD199
CALL SETUP(RHIWAKE	,4,IWAKE)	ZREAD200
CALL SETUP(RHIWPRNT	,4,IWPRNT)	ZREAD201
CALL SETUP(RHK	,4,	ZREAD202
CALL SETUP(RHLIMIT	,4,LIMIT)	ZREAD203
CALL SETUP(RHLPLD	,4,LPLD)	ZREAD204
CALL SETUP(RHLRED	,4,LRED)	ZREAD205
CALL SETUP(RHMODE	,4,MODE)	ZREAD206
CALL SETUP(RHNAIT	,4,NAIT)	ZREAD207
CALL SETUP(RHNOCKOV	,4,NOCKOV,20)	ZREAD208
CALL SETUP(RHNOCONS	,4,NOCONS)	ZREAD209
CALL SETUP(RHNOCP	,4,NOCP)	ZREAD210
CALL SETUP(RHNOECON	,4,NOECON)	ZREAD211
CALL SETUP(RHNOVCH	,4,NOVCH)	ZREAD212
CALL SETUP(RHNP	,4,NP)	ZREAD213
CALL SETUP(RHNPV	,4,NPV,1AC)	ZREAD214
CALL SETUP(RHNSTWL	,4,NSTWL)	ZREAD215
CALL SETUP(RHNOVCT	,R,NOVCT,20)	ZREAD216
CALL SETUP(RHPFC	,R,PF)	ZREAD217
CALL SETUP(RHPHT1	,R,PH1,10)	ZREAD218
CALL SETUP(RHPHT2	,R,PH2,10)	ZREAD219
CALL SETUP(RHPHT3	,R,PH3,10)	ZREAD220
CALL SETUP(RHPPAND	,R,PPAND)	ZREAD221
CALL SETUP (RPHSL	,R,RHSL)	ZREAD222
CALL SETUP (RPHW	,R,RHW)	ZREAD223
CALL SETUP (RIRSTABL	,R,RIRSTABL,20)	ZREAD224
CALL SETUP (RRTC	,R,RTD)	ZREAD225
CALL SETUP(RHSP	,R,SP,40)	ZREAD226
CALL SETUP(RHSC	,R,SC,40)	ZREAD227
CALL SETUP(RHSIGNL1	,R,SIGNL1)	ZREAD228
CALL SETUP(RHSIGNL2	,R,SIGNL2)	ZREAD229

CALL SETUP(OHSIGNL1 ,P,SIGNL1)
CALL SETUP(OHSIGNL2 ,P,SIGNL2)

ZREAC239
ZREAC239

CALL SETUP(OHSIGNL3 ,P,SIGNL3)
CALL SETUP(OHSMULT ,P,SMULT,25)
CALL SETUP(OHSRS ,P,SRS,9)
CALL SETUP(OHSV ,P,SV,40)
CALL SETUP(OHSWL1 ,P,SWL1,40)
CALL SETUP(OHSWL2 ,P,SWL2,40)
CALL SETUP(OHSWL3 ,P,SWL3,40)
CALL SETUP(OHSWP1 ,P,SWP1,40)
CALL SETUP(OHSWP2 ,P,SWP2,40)
CALL SETUP(OHSWP3 ,P,SWP3,40)

ZREAC240
ZREAC241
ZREAC242
ZREAC243
ZREAC244
ZREAC245
ZREAC246
ZREAC247
ZREAC248
ZREAC249
ZREAC250

CALL SETUP(HTABL ,P,HTABL)
CALL SETUP(HTAU1 ,P,HTAU1)
CALL SETUP(HTAU2 ,P,HTAU2)
CALL SETUP(HTAU3 ,P,HTAU3)
CALL SETUP(HTCP ,P,HTCP,40)
CALL SETUP(HTCC ,P,HTCC,40)
CALL SETUP(HTCV ,P,HTCV,40)
CALL SETUP(HTCWL1 ,P,HTCWL1,40)
CALL SETUP(HTCWL2 ,P,HTCWL2,40)
CALL SETUP(HTCWL3 ,P,HTCWL3,40)
CALL SETUP(HTCWR1 ,P,HTCWR1,40)
CALL SETUP(HTCWR2 ,P,HTCWR2,40)
CALL SETUP(HTCWR3 ,P,HTCWR3,40)
CALL SETUP(HTHTTL ,P,HTHTTL,11)
CALL SETUP(HTPLOT ,P,HTPLOT,160)

ZREAC251
ZREAC252
ZREAC253
ZREAC254
ZREAC255
ZREAC256
ZREAC257
ZREAC258
ZREAC259
ZREAC260
ZREAC261
ZREAC262
ZREAC263
ZREAC264
ZREAC265

CALL SETUP(HUID ,P,HUID,20)

ZREAC266
ZREAC267

CALL SETUP(HVPLT ,P,HVPLT,160)
CALL SETUP(HVOPPLT ,P,HVOPPLT,160)

ZREAC268
ZREAC269
ZREAC270
ZREAC271

CALL SETUP(HWKAL1 ,P,HWKAL1,10)
CALL SETUP(HWLI1P ,P,HWLI1P,160)
CALL SETUP(HWLI2P ,P,HWLI2P,160)
CALL SETUP(HWLI3P ,P,HWLI3P,160)
CALL SETUP(HWRF ,P,HWRF)
CALL SETUP(HWR1P ,P,HWR1P,160)
CALL SETUP(HWR2P ,P,HWR2P,160)
CALL SETUP(HWR3P ,P,HWR3P,160)
CALL SETUP(HWSTALT ,P,HWSTALT)

ZREAC272
ZREAC273
ZREAC274
ZREAC275
ZREAC276
ZREAC277
ZREAC278
ZREAC279
ZREAC280
ZREAC281

CALL SETUP(HXCOM ,P,HXCOM,200)
CALL SETUP(HXOTABL ,P,HXOTABL,11)
CALL SETUP(HX2PRD ,P,HX2PRD)
CALL SETUP(HX3B ,P,HX3B)

ZREAC282
ZREAC283
ZREAC284
ZREAC285
ZREAC286

CALL SETUP(HXOTABL ,P,HXOTABL(1,2), 11)

ZREAC287
ZREAC288

CALL SETUP(HXOTABL ,P,HXOTABL(1,3), 11)

ZREAC289
ZREAC290

CALL SETUP(HZPLT ,P,HZPLT,160)

ZREAC291
ZREAC292

RETURN

ZREAC293
ZREAC294

NOT REPRODUCIBLE

CACOS	IMPLICIT REAL*8 (A-H,O-Z)	ACOS	1
CJACOS	ACOS ACOSR ACOSD	ACOS	2
	REAL FUNCTION ACOSR*(W)	ACOS	3
	ENTRY ACOS(W)	ACOS	4
C	THE ANGLE RETURNED WILL BE BETWEEN 0 AND PIE	ACOS	5
	D = 1.000	ACOS	6
2	WS = 1.000 - W * W	ACOS	7
	ACOSR = ATANR (DSQRT(WS),W) / D	ACOS	8
3	RETURN	ACOS	9
	ENTRY ACOSDIW:	ACOS	10
	F = .17453292519943290-1	ACOS	11
	CD TO 2	ACOS	12
	END	ACOS	13
		ACOS	14

CASIN	IMPLICIT REAL*8 (A-H,O-Z)	ASIN	1
CJASIN	ASIN ASINR ASIND	ASIN	2
	REAL FUNCTION ASINR*(W)	ASIN	3
	ENTRY ASIN(W)	ASIN	4
C	THE ANGLE RETURNED WILL BE BETWEEN 0 AND $\text{PI}/2$ OR $3*\text{PI}/2$ AND $2*\text{PI}$	ASIN	5
	D = 1.000	ASIN	6
2	WS = 1.000 - W * W	ASIN	7
	ASINR = ATANQR (W,DSQRT(WS)) / D	ASIN	8
3	RETURN	ASIN	9
	ENTRY ASIND(W)	ASIN	10
	C = .17453292519943290-1	ASIN	11
	GO TO 2	ASIN	12
	END	ASIN	13
		ASIN	14

CATAN	IMPLICIT REAL*8 (A-H,O-Z)	ATAN	1
CJTNQR	ATANQR ATANQD	ATAN	2
	REAL FUNCTION ATANQR*8(W,H)	ATAN	3
	ENTRY ATANQ(W,H)	ATAN	4
	D = 1.000	ATAN	5
110	X = CATAN2 (W,H)	ATAN	6
	IF (X) 1, 2, 2	ATAN	7
1	X = XC6.283185307179586	ATAN	8
2	ATANQR = X / D	ATAN	9
4	RETURN	ATAN	10
	ENTRY ATANQD(W,H)	ATAN	11
	C = .17453292519943250-1	ATAN	12
	GO TO 110	ATAN	13
	END	ATAN	14
		ATAN	15

CENDJOB
SURROUTINE ENDJOB
C
C ENDJOB IS A DUMMY SLBROUTINE
C
RETURN
END

ENDJOB 1
ENDJOB 2
ENDJOB 3
ENDJOB 4
ENDJOB 5
ENDJOB 6
ENDJOB 7

CEZPLT	SUBROUTINE EZPLT	EZPLT	1
C		FZPLT	2
C	EZPLT IS A DUMMY SUBROUTINE	FZPLT	3
C		FZPLT	4
	RETURN	EZPLT	5
	END	EZPLT	6
		FZPLT	7

CFERROR		FERRO	1
	SUBROUTINE FERROR	FERRO	2
C		FERRO	3
C	FERROR IS A DUMMY SUBROUTINE	FERRO	4
C		FERRO	5
	RETURN	FERRO	6
	END	FERRO	7

CFRAMEV

SUBROUTINE FRAMEV

C
C
C

FRAMEV IS A DUMMY SUBROUTINE

RETURN
END

FRAME 1
FRAME 2
FRAME 3
FRAME 4
FRAME 5
FRAME 6
FRAME 7

CICFRMV

SUBROUTINE IOFRMV

C
C
C

IOFRMV IS A DUMMY SUBROUTINE

RETURN
END

IOFRM 1
IOFRM 2
IOFRM 3
IOFRM 4
IOFRM 5
IOFRM 6
IOFRM 7

CM180M

SUBROUTINE M180M

C
C
C

M180M IS A DUMMY SUBROUTINE

RETURN
END

M180 1
M180 2
M180 3
M180 4
M180 5
M180 6
M180 7

CPLTNC
SUBROUTINE PLTND
C
C PLTND IS A DUMMY SUBROUTINE
C
RETURN
END

PLTND 1
PLTND 2
PLTND 3
PLTND 4
PLTND 5
PLTND 6
PLTND 7

READ	TITLE	VARIABLE FIELD INPUT ROUTINE	EAD
	S FOR 05/360-		1
	PRINT 0N,DATA,GEN		2
	MACR0		3
+NAME	C0L72 +H0W,+T0		4
+NAME	LA RC,1)0,RC*	INCREMENT COLUMN COUNTER	5
	CR RC,RCMAX	SEE IF COLUMN 72 HAS BEEN REACHED	6
	B+H0W +T0	BRANCH ACCORDINGLY	7
	MEND		8
	MACR0		9
+NAME	C0MP +WITH,+H0W,+T0		10
+NAME	CLI 0IRC*,+WITH	EXAMINE THE CHARACTER ON THE CARD	11
	B+H0W +T0	BRANCH ACCORDINGLY	12
	MEND		13
	MACR0		14
+NAME	CHECK +REG,+H0W,+T0		15
+NAME	LTR +REG,+REG	TEST REGISTER	16
	B+H0W +T0	BRANCH ACCORDINGLY	17
	MEND		18
	MACR0		19
+NAME	CNV +L0C		20
+NAME	UNPK +L0C.12*,DB.6)2*	UNPACK FOR PRINTING	21
	0I +L0C.1,X-F0-	DELETE SIGN	22
	MEND		23
	MACR0		24
+NAME	C0NV +REG,+L0C		25
+NAME	SLL +REG,4	HALF-BYTE SHIFT TO MAKE ROOM FOR SIGN	26
	ST +REG,DB.4	STORE IN RIGHT HALF OF DOUBLE WORD	27
	0I DB.7,X-0C-	INSERT PLUS SIGN	28
	CNV +L0C		29
	MEND		30
LA000000	START 0		31
	ENTRY READIN,WHERE,SETUP,HEADING		32
	EXTRN EXIT,FDXPI=,JUC0M=		33
	EJECT		34
*	SUBROUTINE READIN(INC0L1* OR READIN(INC0L1,** WHEN INC0L1=8888		35
*	A. L. ATKINSON, R430, AUGUST 1966. BAL VERSION - JUNE 1967		36
*	LATEST MODIFICATION - 9 FEBRUARY 1968		37
	SPACE 2		38
READIN	BC 15,12)0,15*	BRANCH AROUND ID	39
	DC XL2-0700-		40
	DC CL6-READIN-		41
	STM 14,12,12)13*	SAVE REGISTERS	42
	L RW,0)0,1*	PICK UP ADDRESS OF INC0L1	43
	LR 12,15	PICK UP PROGRAM BASE REGISTER	44
	USING READIN,12	INFORM ASSEMBLER OF PROGRAM BASE REGISTER	45
	L RD,ADATA	PICK UP BASE REGISTER FOR PROGRAM DATA	46
	USING DATA,RD	INFORM ASSEMBLER OF THAT BASE REGISTER	47
	LR 4,13	PICK UP ADDRESS OF OLD SAVE AREA	48
	LA 13,SAVE	PICK UP ADDRESS OF NEW SAVE AREA	49
	ST 13,8)0,4*	SAVE ADDRESS OF NEW SAVE AREA IN OLD ONE	50
	ST 4,SAVE,4	SAVE ADDRESS OF OLD SAVE AREA IN NEW ONE	51
	ST RW,ADINC0L1	SAVE ADDRESS OF INC0L1	52
START0LD	N0P STARTNEW		53
	LH RT,NUMTAB		54
	CHECK RT,NP,ERR0R1	SEE IF THE INPUT TABLE WAS CONSTRUCTED	55
	L RW,0)0,RW*	PICK UP INC0L1	56
	C RW,C8808	WAS IT INITIALIZED TO 8888	57
	BNE SKIP	IF NOT, BRANCH AROUND	58
	MVI IEND,1,X-00-	SET NON-STANDARD RETURN SWITCH	59
			60

SKIP	MVI	STARTØLD.1,X-FØ-	EAD	61-
STARTNEW	SR	RA,RA	EAD	62
	ST	RA,INCØL1 ZERO ØUT INCØL1	EAD	63
	LA	RCMAX,CARD,71 PICK UP ADDRESS ØF CARD CØLUMN 72	EAD	64
TITLE	NØP	NEXTCARD	EAD	65
	SDR	Ø,Ø	EAD	66
	STD	Ø,DØ CLEAR DØUBLE WØRD	EAD	67
	TIME	DEC PICK UP TIME AND DATE	EAD	68
	ST	1,DATE	EAD	69
	SR	1,1	EAD	70
	SRDL	Ø,24 ØBTAIN HØURS	EAD	71
	CØNV	Ø,HRS CØNVERT FØR PRINTING	EAD	72
	CVB	Ø,DØ CØNVERT TØ BINARY	EAD	73
	C	Ø,=F-1Ø-	EAD	74
	UNL	ØKK IF LESS THAN TEN,	EAD	75
	MVI	HRS,X-4Ø- INSERT BLANK ØN LEFT	EAD	76
ØKK	SR	Ø,Ø	EAD	77
	SLDL	Ø,B ØBTAIN MINUTES	EAD	78
	CØNV	Ø,MNS CØNVERT FØR PRINTING	EAD	79
	SR	Ø,Ø	EAD	80
	SLDL	Ø,B ØBTAIN SECØNDS	EAD	81
	CØNV	Ø,SCS CØNVERT FØR PRINTING	EAD	82
	SR	Ø,Ø	EAD	83
	SLDL	Ø,B ØBTAIN HUNDREDTHS ØF A SECØND	EAD	84
	CØNV	Ø,HSC	EAD	85
	SDR	Ø,Ø	EAD	86
	STD	Ø,DØ	EAD	87
	L	5,DATE	EAD	88
	LR	1,5	EAD	89
	STH	5,DØ.6 ØBTAIN DAY ØF THE YEAR	EAD	90
	CVB	5,DØ CØNVERT TØ BINARY INTEGER	EAD	91
	SRL	1,16 ØBTAIN YEAR	EAD	92
	CØNV	1,YR CØNVERT FØR PRINTING	EAD	93
	CVB	2,DØ CØNVERT TØ A BINARY INTEGER	EAD	94
	SR	6,6 MONTH CØUNTER	EAD	95
	LR	3,6	EAD	96
	SRDL	2,2 DIVIDE YEAR BY FØUR	EAD	97
	LTR	3,2 IS THIS A LEAP YEAR	EAD	98
	LA	2,4 INDEXING FØR SEARCH FØR MONTH	EAD	99
	LA	3,44	EAD	100
	BNZ	NØLEAP	EAD	101
	C	5,=F-6Ø- CHECK FØR FEBRUARY 29TH	EAD	102
	BL	NØLEAP	EAD	103
	ØH	LEAP	EAD	104
FØD29	LA	6,2	EAD	105
	LA	5,29	EAD	106
	B	DØNE	EAD	107
LEAP	LA	6,8 MONTH IS MARCH, AT LEAST	EAD	108
	BCTR	5,Ø REDUCE DATE TØ ACCØUNT FØR LEAP YEAR	EAD	109
NØLEAP	C	5,JAN16* CØMPARE WITH LAST DAY ØF THE MONTH	EAD	110
	BNH	MONTH FOUND THE MONTH	EAD	111
	ØXLE	6,2,NØLEAP LOOP AGAIN	EAD	112
MONTH	S	5,INCØL116* CØNVERT TØ DAY ØF MONTH	EAD	113
	SRL	6,2 ØBTAIN NUMBER	EAD	114
	LA	6,11Ø,6* ØF MONTH	EAD	115
DØNE	CVD	5,DØ CØNVERT DAY ØF MONTH TØ DECIMAL	EAD	116
	CNV	DAY CØNVERT FØR PRINTING	EAD	117
	CVD	6,DØ CØNVERT MONTH TØ DECIMAL	EAD	118
	CNV	MTH CØNVERT FØR PRINTING	EAD	119

	C	6,F-10-		EAD	120
	UNL	OK	IF LESS THAN TEN,	EAD	121
	MVI	MTH,X-40-	INSERT BLANK ON LEFT	EAD	122
OK	L	15,VIBC0M	WRITE HEADING FOR A NEW PAGE OF DATA	EAD	123
	CNOP	0,4		EAD	124
	BAL	14,4)0,15*		EAD	125
	DC	F-6-		EAD	126
	DC	A)F1*		EAD	127
	BAL	14,16)0,15*		EAD	128
NEXTCARD	L	RW,INC0L1	PICK UP INC0L1	EAD	129
	CHECK	RW,P,REST0RE	SEE IF CARD WAS A TRANSFER CARD	EAD	130
READCARD	L	15,VIBC0M		EAD	131
	CNOP	0,4		EAD	132
	BAL	14,0)0,15*	READ A CARD	EAD	133
	DC	F-5-		EAD	134
	DC	A)F2*		EAD	135
	BAL	14,12)0,15*		EAD	136
	DC	A)CARD*		EAD	137
	DC	XL4-0860000A-		EAD	138
	BAL	14,16)0,15*		EAD	139
PRINT	NOP	START		EAD	140
	BAL	14,4)0,15*	WRITE THE CARD	EAD	141
	DC	F-6-		EAD	142
	DC	A)F3*		EAD	143
	BAL	14,12)0,15*		EAD	144
	DC	A)CARD*		EAD	145
	DC	XL4-0860000A-		EAD	146
	BAL	14,16)0,15*		EAD	147
START	LA	RC,CARD	PICK UP ADDRESS OF CARD COLUMN 1	EAD	148
	SR	R5,R5	ZER0 OUT INCREMENTING FLAG	EAD	149
NFLAG	NOP	ACARD2		EAD	150
	C0MP	BLANK,E,ETC	EXAMINE FIRST CHARACTER ON THE CARD	EAD	151
	C0MP	E,E,IEND		EAD	152
	C0MP	H,E,HCARD		EAD	153
	C0MP	Z,E,HEXCARD		EAD	154
	C0MP	X,E,HEXCARD		EAD	155
	C0MP	A,E,ACARD		EAD	156
	C0MP	D0LLAR,E,READCARD		EAD	157
	C0MP	STAR,E,READCARD		EAD	158
	C0MP	N,E,NNNN		EAD	159
	C0MP	F,E,FFFF		EAD	160
	C0MP	P,E,PPPP		EAD	161
	C0MP	ZER0,NH,ETC		EAD	162
	NI	0)RC*,X-0F-	OBTAIN DIGIT FROM TRANSFER CARD	EAD	163
	SR	RW,RW		EAD	164
	IC	RW,0)0,RC*		EAD	165
	ST	RW,INC0L1	STORE IT IN INC0L1	EAD	166
ETC	LA	RC,1)0,RC*	INCREMENT COLUMN COUNTER	EAD	167
NEWFIELD	CR	RC,RCMAX	LOOK FOR A NEW FIELD	EAD	168
	BH	NEXTCARD		EAD	169
S10	C0MP	BLANK,E,NEXT	EXAMINE NEXT CHARACTER ON THE CARD	EAD	170
	C0MP	C0MMA,E,NEXT		EAD	171
	C0MP	EQ029,E,NEXT		EAD	172
	C0MP	EQ026,NE,CHAR1		EAD	173
NEXT	C0L72	NH,S10		EAD	174
S9	B	NEXTCARD		EAD	175
NNNN	CLC	0)7,RC*,N0PRINT		EAD	176
	BE	N0PRINTS		EAD	177
	CLC	0)6,RC*,N0FLIP		EAD	178

	BNE	ETC		EAD	179
	MVI	F1.3,X-F0-	SUPPRESS PAGE FLIP FOR EACH CASE	EAD	180
	B	READCARD		EAD	181
NOPRINTS	MVI	PRINT.1,X-F0-	SUPPRESS PRINTING OF DATA CARDS	EAD	182
	MVI	TITLE.1,X-F0-		EAD	183
	B	READCARD		EAD	184
FFFF	CLC	014,RC*,FLIP		EAD	185
	BNE	ETC		EAD	186
	MVI	F1.3,X-F1-	FLIP PAGE FOR EACH CASE	EAD	187
	B	READCARD		EAD	188
PPPP	CLC	015,RC*,PRINTS		EAD	189
	BNE	ETC		EAD	190
	MVI	PRINT.1,X-00-	PRINT DATA CARDS	EAD	191
	MVI	TITLE.1,X-00-		EAD	192
	B	READCARD		EAD	193
ADATA	DC	AIDATA*		EAD	194
CHAR1	MVI	L0G.1,X-00-	EXAMINE FIRST CHARACTER IN NEW FIELD	EAD	195
	RS,RS		ZERØ OUT DECIMAL POINT LOCATOR	EAD	196
	LR	RB,RS	ZERØ OUT EXPONENT LOCATOR	EAD	197
	COMP	ZERØ,NL,S503	CHECK FOR NUMERIC	EAD	198
	COMP	A,NL,LETTER	CHECK FOR ALPHABETIC	EAD	199
	COMP	DOLLAR,E,LETTER	DOLLAR SIGN IS ALPHABETIC	EAD	200
	COMP	MINUS,E,S510		EAD	201
	COMP	SLASH,E,ALPHA		EAD	202
	COMP	LPO26,E,SUBSCRPT		EAD	203
	COMP	LPO29,E,SUBSCRPT		EAD	204
	COMP	DECIMAL,E,S508		EAD	205
	COMP	PLUS026,E,S509		EAD	206
	COMP	PLUS029,E,S509		EAD	207
	B	ERRØR		EAD	208
	EJECT			EAD	209
*		NUMERIC FIELD		EAD	210
		SPACE 2		EAD	211
S508	MVI	SIGN.1,X-C0-	DECIMAL POINT, SET SIGN PLUS	EAD	212
	MVI	L0G.1,X-F0-	INITIALIZE LOGICAL FLAG	EAD	213
S5080	CHECK	RS,NZ,ERRØR	CHECK FOR MORE THAN ONE DECIMAL POINT	EAD	214
	LR	RS,RC	SET DECIMAL POINT LOCATOR	EAD	215
	B	S5090		EAD	216
S510	MVI	SIGN.1,X-D0-	SET SIGN MINUS	EAD	217
	B	S5090		EAD	218
L0G	NØP	L0GICAL	BRANCH ONLY IF DECIMAL POINT PRECEDES	EAD	219
	B	ERRØR		EAD	220
S509	MVI	SIGN.1,X-C0-	SET SIGN PLUS	EAD	221
S5090	CØL72	H,ERRØR		EAD	222
	COMP	ZERØ,NL,NUM		EAD	223
	COMP	DECIMAL,E,S5080		EAD	224
	COMP	F,NL,L0G		EAD	225
	B	ERRØR		EAD	226
S503	MVI	SIGN.1,X-C0-	NUMERIC, SET SIGN PLUS	EAD	227
NUM	LR	RT,RC	PROCESS NUMERIC FIELD	EAD	228
	LA	RW,910,RT*		EAD	229
	ST	RW,D8		EAD	230
NEXTC	CØL72	H,S501	EXAMINE NEXT CHARACTER IN NUMERIC FIELD	EAD	231
	COMP	ZERØ,NL,S513		EAD	232
	COMP	D,NL,S504		EAD	233
	COMP	DECIMAL,E,S502		EAD	234
	COMP	MINUS,E,S521		EAD	235
	COMP	PLUS026,E,S521		EAD	236
	COMP	PLUS029,E,S521		EAD	237

	COMP	BLANK,E,S501		EAD	238
	COMP	LP026,E,LEFTP		EAD	239
	COMP	LP029,E,LEFTP		EAD	240
	COMP	RP026,E,RIGHTP		EAD	241
	COMP	RP029,E,RIGHTP		EAD	242
	COMP	COMMA,NE,ERROR		EAD	243
S501	CHECK	RS,NZ,FL0AT	CHECK FOR FLOATING POINT	EAD	244
INTGER	CHECK	R5,Z,INTGER1	CHECK FOR INCREMENTING	EAD	245
	BP	ERR0R	UNBALANCED PARENTHESES	EAD	246
	SR	R5,R5	ZER0 OUT INCREMENTING FLAG	EAD	247
	B	INCREM		EAD	248
INTGER1	BAL	14,LA000C	CONVERT BCD TO BINARY INTEGER	EAD	249
	CLI	LENGTH,1,3	CHECK LENGTH OF VARIABLE	EAD	250
	BH	INC4		EAD	251
	MVI	ST0RE1,X-40-	CONVERT ST0RE INSTRUCTION TO -STH-	EAD	252
ST0RE1	ST	R7,010,RA*	ST0RE INTEGER VALUE	EAD	253
	AR	RA,RI	INCREMENT ST0RAGE ADDRESS	EAD	254
DASHI	B	NEWFIELD		EAD	255
DASHI1	N0P	DASHI2		EAD	256
	LH	RS,NMAX		EAD	257
	SH	RS,NMIN		EAD	258
	A	RS,0NE		EAD	259
	MVI	DASHI1,1,X-F0-		EAD	260
DASHI2	BCT	RS,ST0RE1		EAD	261
	MVI	DASHI1,1,X-00-		EAD	262
	B	DASHD0NE		EAD	263
LEFTP1	S	RC,0NE		EAD	264
LEFTP	CHECK	RS,NZ,FL0ATL	CHECK FOR FLOATING POINT	EAD	265
	CHECK	R5,Z,FIRSTL	IS THIS THE FIRST LEFT PARENTHESIS	EAD	266
	BP	ERR0R	UNBALANCED PARENTHESES	EAD	267
	LA	R5,1	SET INCREMENTING FLAG	EAD	268
INCREM	BAL	14,LA000C	CONVERT BCD TO BINARY INTEGER	EAD	269
	LR	14,RW	PICK UP NST0P	EAD	270
	SR	14,R7	SUBTRACT NSTART	EAD	271
	SRDA	14,32	RIGHT ADJUST FOR DIVISION	EAD	272
	DR	14,R11	DIVIDE BY NINCRM	EAD	273
	CHECK	R5,P,MULT	CHECK FOR MULTIPLE INCREMENTING	EAD	274
	LA	15,110,15*	STANDARD INCREMENTING, S0 ST0RE NST0P ALS0	EAD	275
MULT	CLI	LENGTH,1,3	CHECK LENGTH OF VARIABLE	EAD	276
	BH	INC4		EAD	277
	MVI	LPINC,X-40-	CONVERT ST0RE INSTRUCTION TO -STH-	EAD	278
LPINC	ST	R7,010,RA*	ST0RE INTEGER VALUE	EAD	279
	AR	R7,R11	INCREMENT INTEGER VALUE	EAD	280
	AR	RA,RI	INCREMENT ST0RAGE ADDRESS	EAD	281
	BCT	15,LPINC		EAD	282
	CHECK	R5,P,QQQ	CHECK FOR MULTIPLE INCREMENTING	EAD	283
	C0L72	H,NEXTCARD		EAD	284
	B	NEWFIELD		EAD	285
INC4	MVI	LPINC,X-50-	CONVERT ST0RE INSTRUCTION TO -ST-	EAD	286
	B	LPINC		EAD	287
FIRSTL	LA	R5,1	SET INCREMENTING FLAG	EAD	288
	BAL	14,LA000C	CONVERT BCD TO BINARY INTEGER	EAD	289
QQQ	LR	R7,RW	PICK UP NSTART	EAD	290
	C0L72	H,ERR0R		EAD	291
	B	NEWFIELD		EAD	292
RIGHTP1	S	RC,0NE		EAD	293
RIGHTP	CHECK	RS,NZ,FL0ATR	CHECK FOR FLOATING POINT	EAD	294
	CHECK	R5,NP,ERR0R	ERR0R IF N0 LEFT PARENTHESIS HAS PRECEDED	EAD	295
	LCP	R5,R5	CHANGE SIGN OF INCREMENTING FLAG	EAD	296

	BAL	14,LA000C	CONVERT BCD TO BINARY INTEGER	EAD 297
	LR	R11,RW	PICK UP NINCRM	EAD 298
	C0L72	H,ERROR		EAD 299
	B	NEWFIELD		EAD 300
S519	CHECK	RS,NZ,NEXTC	BRANCH IF DECIMAL POINT HAS BEEN SEEN	EAD 301
	C	RC,D8	IF MORE THAN NINE DIGITS TO LEFT OF	EAD 302
	BH	GR9	DECIMAL POINT, BRANCH OUT	EAD 303
	B	NEXTC		EAD 304
S504	COMP	E,H,ERROR	EXPONENT FIELD	EAD 305
	CHECK	R8,NZ,ERROR	ERROR IF MORE THAN ONE EXPONENT FIELD	EAD 306
	CLI	1)RC*,BLANK	CHECK FOR BLANK AFTER D OR E	EAD 307
	BNE	S5041		EAD 308
	MVI	1)RC*,PLUS026	INSERT PLUS SIGN OVER BLANK	EAD 309
S5041	LR	R8,RC	PICK UP LOCATION OF EXPONENT	EAD 310
	CHECK	RS,NZ,NEXTC	SEE IF DECIMAL POINT HAS BEEN FOUND	EAD 311
	LR	RS,RC	IF NOT, ASSUME IT IS TO LEFT OF EXPONENT	EAD 312
	B	NEXTC	EXAMINE NEXT CHARACTER	EAD 313
INT4	MVI	STOREI,X-50-	CONVERT STORE INSTRUCTION TO -ST-	EAD 314
	B	STOREI		EAD 315
S521	CHECK	R8,NZ,NEXTC	BRANCH IF E OR D HAS BEEN SEEN	EAD 316
	B	S5041		EAD 317
S502	CHECK	RS,NZ,ERROR	ERROR IF MORE THAN ONE DECIMAL POINT	EAD 318
	LR	RS,RC	SET DECIMAL POINT LOCATOR	EAD 319
	B	NEXTC		EAD 320
GR9	C0L72	H,INTG	MORE THAN NINE DIGITS TO LEFT OF DECIMAL	EAD 321
	COMP	BLANK,E,INTG		EAD 322
	COMP	LPO26,E,LEFTP1		EAD 323
	COMP	LPO29,E,LEFTP1		EAD 324
	COMP	RPO26,E,RIGHTP1		EAD 325
	COMP	RPO29,E,RIGHTP1		EAD 326
	COMP	COMMA,NE,ERROR		EAD 327
INTG	S	RC,ONE		EAD 328
	B	INTGER		EAD 329
	EJECT			EAD 330
*		THE PROCESSING OF FLOATING POINT DATA		EAD 331
	SPACE	2		EAD 332
*		ADDRESS OF FIRST NUMERICAL CHARACTER IN RT		EAD 333
*		ADDRESS OF NEXT BYTE AFTER LAST CHARACTER IN RC		EAD 334
*		ADDRESS OF DECIMAL POINT IN RS		EAD 335
*		ADDRESS OF EXPONENT IN R8		EAD 336
*		IF NO EXPONENT, R8 IS SET TO ZERO		EAD 337
*		FLOATING POINT ANSWER IN REGISTER 2		EAD 338
FL0ATL	CHECK	R5,2,FL0ATLP	IS THIS THE FIRST LEFT PARENTHESIS	EAD 339
	BP	ERROR	UNBALANCED PARENTHESES	EAD 340
MULTIRC	LA	R5,1	SET INCREMENTING FLAG	EAD 341
	B	FL0ATN-4		EAD 342
FL0ATLP	MVI	STOREF,1,X-F0-		EAD 343
	LA	R5,1	SET INCREMENTING FLAG	EAD 344
	B	FL0ATN		EAD 345
FL0ATR	CHECK	R5,NP,ERROR	ERROR IF NO LEFT PARENTHESIS HAS PRECEDED	EAD 346
	LCR	R5,R5	CHANGE SIGN OF INCREMENTING FLAG	EAD 347
	MVI	STOREF,5,X-F0-		EAD 348
	B	FL0ATN		EAD 349
FL0AT	CHECK	R5,2,FL0ATN	CHECK FOR INCREMENTING	EAD 350
	BP	ERROR	UNBALANCED PARENTHESES	EAD 351
	SR	R5,R5	ZERO OUT INCREMENTING FLAG	EAD 352
	MVI	STOREF,9,X-F0-		EAD 353
FL0ATN	ST	RC,C0LN0	SAVE CARD COLUMN COUNTER	EAD 354
	SDR	D,0		EAD 355

	CHECK	R8,NZ,EXPON	CHECK FOR EXPONENT IN FIELD	EAD	356
G00N	CR	RS,R1		EAD	357
	BL	LT0NE		EAD	358
	ST	RC,C0L		EAD	359
	LR	RC,RS		EAD	360
	BAL	14,LA000C	CONVERT TO BINARY INTEGER	EAD	361
	BAL	14,LA000F	CONVERT TO FLOATING POINT	EAD	362
	L	RC,C0L		EAD	363
LT0NE	LR	RW,RC		EAD	364
	LA	RT,110,RS*		EAD	365
	LDR	2,0		EAD	366
	SR	RW,RT		EAD	367
	BNP	EXPF		EAD	368
	C	RW,NINE		EAD	369
	BH	M0RE		EAD	370
FIRST9	LR	RS,RW		EAD	371
	BAL	14,LA000C	CONVERT TO BINARY INTEGER	EAD	372
	BAL	14,LA000F	CONVERT TO FLOATING POINT	EAD	373
	SLA	RS,3		EAD	374
	MD	0,TENT)RS*		EAD	375
	ADR	2,0		EAD	376
MR1	N0P	M0RE1		EAD	377
MR2	N0P	M0RE2		EAD	378
EXPF	N0P	EXPN		EAD	379
	B	ST0REF		EAD	380
EXPON	LR	RC,R8		EAD	381
	IC	RW,0)0,R8*		EAD	382
	STC	RW,NUMWD		EAD	383
	MVI	EXPF,1,X-F0-		EAD	384
	B	G00N		EAD	385
M0RE	MVI	MR1,1,X-F0-		EAD	386
	ST	RC,C0L		EAD	387
	LA	RC,9)0,RT*		EAD	388
	IC	RW,0)0,RC*		EAD	389
	STC	RW,NUMWD,1		EAD	390
	L	RW,NINE		EAD	391
	B	FIRST9		EAD	392
M0RE1	MVI	MR1,1,X-00-		EAD	393
	IC	RW,NUMWD,1		EAD	394
	STC	RW,0)0,RC*		EAD	395
	LR	RT,RC		EAD	396
	L	RC,C0L		EAD	397
	MVI	MR2,1,X-F0-		EAD	398
	LR	RW,RC		EAD	399
	SR	RW,RT		EAD	400
	C	RW,NINE		EAD	401
	BNH	*,12		EAD	402
	L	RW,NINE		EAD	403
	LA	RC,9)0,RT*		EAD	404
	A	RW,NINE		EAD	405
	B	FIRST9		EAD	406
M0RE2	MVI	MR2,1,X-00-		EAD	407
	L	RC,C0L		EAD	408
	B	EXPF		EAD	409
REAL4	MVI	ST0REFL,X-70-	CONVERT STORE INSTRUCTION TO -STE-	EAD	410
	B	ST0REFL		EAD	411
NEXTE	C0L72	H,ERR0R		EAD	412
	B	NXTE		EAD	413
P0S	MVI	SIGN,1,X-C0-		EAD	414

NEXTN	C0L72	H,ERROR	EAD	415
	B	NEXP	EAD	416
NEG	MVI	SIGN.1,X-D0-	EAD	417
	B	NEXTN	EAD	418
EXPN	LR	RT,R8	EAD	419
	MVI	SIGN.1,X-C0- ASSUME EXPONENT SIGN IS POSITIVE	EAD	420
	MVI	EXPF.1,X-00-	EAD	421
	LR	RC,RT	EAD	422
	IC	RW,NUMWD	EAD	423
	STC	RW,010,RC*	EAD	424
	COMP	D,E,NEXTE	EAD	425
	COMP	E,E,NEXTE	EAD	426
NXTE	COMP	ZER0,NL,NEXP	EAD	427
	COMP	MINUS,E,NEG	EAD	428
	COMP	PLUS026,E,POS	EAD	429
	COMP	PLUS029,E,POS	EAD	430
	B	ERROR	EAD	431
NEXP	LR	RT,RC	EAD	432
	L	RC,C0LN0 RESTORE CARD COLUMN COUNTER	EAD	433
	BAL	14,LA000C CONVERT EXPONENT TO A BINARY INTEGER	EAD	434
	ST	RW,DB	EAD	435
	ST	RA,C0L	EAD	436
	LA	1,ARGLIST	EAD	437
	STD	2,D0UBLE	EAD	438
	L	15,VFDXPI	EAD	439
	BALR	14,15	EAD	440
	LD	2,D0UBLE	EAD	441
	L	RA,C0L	EAD	442
	MDR	2,0	EAD	443
ST0REF	N0P	FLTRP	EAD	444
	N0P	FLTRP	EAD	445
	N0P	FINCRM	EAD	446
	CLI	LENGTH.1,6 CHECK LENGTH OF VARIABLE	EAD	447
	DL	REAL4	EAD	448
	MVI	ST0REFL,X-60- CONVERT STORE INSTRUCTION TO -STD-	EAD	449
ST0REFL	STD	2,010,RA* STORE FLOATING POINT VALUE	EAD	450
	AR	RA,R1 INCREMENT STORAGE ADDRESS	EAD	451
DASHF	B	NEWFIELD	EAD	452
DASHF1	N0P	DASHF2	EAD	453
	LH	RS,NMAX	EAD	454
	SH	RS,NMIN	EAD	455
	A	RS,0NE	EAD	456
	MVI	DASHF1.1,X-F0-	EAD	457
DASHF2	BCT	RS,ST0REFL	EAD	458
	MVI	DASHF1.1,X-00-	EAD	459
	B	DASHD0NE	EAD	460
FLTRP	MVI	ST0REF.1,X-00-	EAD	461
	LDR	6,2 PICK UP FSTART	EAD	462
	C0L72	H,ERROR	EAD	463
	B	NEWFIELD	EAD	464
FLTRP	MVI	ST0REF.5,X-00-	EAD	465
	LDR	4,2 PICK UP FINCRM	EAD	466
	C0L72	H,ERROR	EAD	467
	B	NEWFIELD	EAD	468
FINCRM	MVI	ST0REF.9,X-00-	EAD	469
	LDR	0,2 PICK UP FSTOP	EAD	470
	SDR	0,6 SUBTRACT FSTART	EAD	471
	DDR	0,4 DIVIDE BY FINCRM	EAD	472
	AD	0,TENT.96 ADJUST FOR ROUND-OFF	EAD	473

	AW	0,CFLOAT	CONVERT TO AN INTEGER	EAD	474
	STD	0,D8		EAD	475
	L	15,D8.4		EAD	476
	CHECK	R5,P,MULTF	CHECK FOR MULTIPLE INCREMENTING	EAD	477
	LA	15,110,15*	STANDARD INCREMENTING, SO STORE FSTOP ALSO	EAD	478
MULTF	CLI	LENGTH,1,6	CHECK LENGTH OF VARIABLE	EAD	479
	BL	RINC4		EAD	480
	MVI	LPINCF,X-60-	CONVERT STORE INSTRUCTION TO -STD-	EAD	481
LPINCF	STD	6,010,RA*	STORE FLOATING POINT VALUE	EAD	482
	ADR	6,4	INCREMENT FLOATING POINT VALUE	EAD	483
	AR	RA,RI	INCREMENT STORAGE ADDRESS	EAD	484
	BCT	15,LPINCF		EAD	485
	CHECK	R5,P,FLTLP,4	CHECK FOR MULTIPLE INCREMENTING	EAD	486
	C0L72	H,NEXTCARD		EAD	487
	B	NEWFIELD		EAD	488
RINC4	MVI	LPINCF,X-70-	CONVERT STORE INSTRUCTION TO -STE-	EAD	489
	B	LPINCF		EAD	490
	EJECT			EAD	491
*		THE PROCESSING OF A-TYPE CARDS		EAD	492
	SPACE	2		EAD	493
ACARD	MVI	NFLAG,1,X-F0-	SET A-CARD FLAGS	EAD	494
	MVI	NFLAG1,1,X-F0-		EAD	495
	MVI	NFLAG2,1,X-F0-		EAD	496
	MVI	NFLAG3,1,X-F0-		EAD	497
	LA	R8,1		EAD	498
S321	BXLE	RC,R8,*8		EAD	499
	B	ERRØR		EAD	500
	CØMP	BLANK,E,S321		EAD	501
	CØMP	A,NL,LETTER		EAD	502
	CØMP	DØLLAR,E,LETTER		EAD	503
	B	ERRØR		EAD	504
ACARDI	LA	R8,1	LOOK FOR FIELD LENGTH	EAD	505
	SR	RW,RW		EAD	506
S322	BXLE	RC,R8,*8		EAD	507
	B	ERRØR		EAD	508
	CØMP	BLANK,E,S322		EAD	509
	CØMP	ZERØ,NL,S323		EAD	510
	CØMP	LPO26,E,SUBSCRPT		EAD	511
	CØMP	LPO29,E,SUBSCRPT		EAD	512
	B	ERRØR		EAD	513
S323	LR	RT,RC	PROCESS FIELD LENGTH	EAD	514
S325	BXLE	RC,R8,*8		EAD	515
	B	S326		EAD	516
	CØMP	BLANK,E,S326		EAD	517
	CØMP	ZERØ,NL,S325		EAD	518
	B	ERRØR		EAD	519
S326	MVI	SIGN,1,X-C0-	INSERT PLUS SIGN	EAD	520
	BAL	14,LA000C	CONVERT BCD TO BINARY INTEGER	EAD	521
	STH	RW,NUMWD	SAVE AS NUMBER OF WORDS	EAD	522
	MH	RW,LENGTH	MULTIPLY BY LENGTH OF VARIABLE	EAD	523
	C	RW,C72	COMPARE RESULT WITH 72	EAD	524
	BH	ERRØR	ERROR IF GREATER THAN 72	EAD	525
S327	BXLE	RC,R8,*8		EAD	526
	B	S329		EAD	527
	CØMP	BLANK,E,S327		EAD	528
	CØMP	ZERØ,NL,S323		EAD	529
	B	ERRØR		EAD	530
S329	CHECK	RW,Z,ERRØR		EAD	531
	B	READCARD		EAD	532

ACARD2	LH	RW,LENGTH	STORE THE ALPHAMERIC DATA	EAD 533
	LR	RT,RW		EAD 534
	S	RT,ØNE		EAD 535
	STC	RT,MVA.1		EAD 536
	LH	RT,NUMWD		EAD 537
MVA	MVC	Ø1Ø,RA*,Ø1RC*		EAD 538
	AR	RC,RW		EAD 539
	AR	RA,RI		EAD 540
	BCT	RT,MVA		EAD 541
	MVI	NFLAG.1,X-ØØ-	RESET A-CARD FLAGS	EAD 542
	MVI	NFLAG1.1,X-ØØ-		EAD 543
	MVI	NFLAG2.1,X-ØØ-		EAD 544
	MVI	NFLAG3.1,X-ØØ-		EAD 545
	B	READCARD		EAD 546
		SPACE 5		EAD 547
*		THE PROCESSING OF H-TYPE CARDS		EAD 548
		SPACE 2		EAD 549
HCARD	MVC	HEADING.1179*,1)RC*	MØVE TØ ØUTPUT FØRMAT FØR HEDING	EAD 550
	B	READCARD		EAD 551
		EJECT		EAD 552
*		THE PROCESSING OF VARIABLE NAME FIELDS		EAD 553
		SPACE 2		EAD 554
LETTER	MVI	LPAREN.1,X-ØØ-		EAD 555
	LA	RØ,1		EAD 556
	MVI	EITHER.1,X-FØ-		EAD 557
	LR	RW,RC		EAD 558
	A	RW,C7		EAD 559
	MVI	S42Ø.1,X-ØØ-		EAD 560
	LR	RT,RC		EAD 561
MØREL	BXLE	RC,RØ,*Ø.Ø	CHECK FØR CØLUMN 72	EAD 562
	B	S42Ø		EAD 563
	CR	RC,RW	SEE IF EIGHT CHARACTERS HAVE BEEN SEEN	EAD 564
	BH	S4Ø5		EAD 565
PASS1	COMP	BLANK,E,S42Ø	LØØK FØR A FIELD TERMINATØR	EAD 566
	COMP	EQØ26,E,S42Ø		EAD 567
	COMP	EQØ29,E,S42Ø		EAD 568
	COMP	LPO26,E,S419		EAD 569
	COMP	LPO29,E,S419		EAD 570
EITHER	B	MØREL		EAD 571
EXTRA	BXLE	RC,RØ,PASS1	CHECK FØR CØLUMN 72	EAD 572
	B	S42Ø		EAD 573
S4Ø5	MVI	EITHER.1,X-ØØ-	EIGHT CHARACTERS HAVE BEEN SEEN	EAD 574
	MVI	S42Ø.1,X-FØ-		EAD 575
	MVI	MVC1.1,7		EAD 576
	B	PASS1		EAD 577
S419	MVI	LPAREN.1,X-FØ-	LEFT PARENTHESIS TERMINATES NAME	EAD 578
S42Ø	NØP	S42ØØ		EAD 579
	LR	RW,RC		EAD 580
	SR	RW,RT		EAD 581
	S	RW,ØNE		EAD 582
	STC	RW,MVC1.1		EAD 583
S42ØØ	MVC	ØØ18*,BLANKS		EAD 584
MVC1	MVC	ØØ1Ø*,Ø1RT*		EAD 585
LAØØØL	LH	RT,NUMTAB	LØØK FØR VARIABLE NAME IN TABLE	EAD 586
	L	RW,STABLE		EAD 587
LP2	CLC	Ø18,RW*,ØØ		EAD 588
	BE	FØUND		EAD 589
	LA	RW,2Ø1Ø,RW*		EAD 590
	BCT	RT,LP2		EAD 591

	B	ERROR2		EAD 592
FOUND	L	RT,8)0,RW*	NAME HAS BEEN FOUND	EAD 593
	ST	RT,NADD		EAD 594
	LH	RT,12)0,RW*		EAD 595
	STH	RT,LENGTH		EAD 596
	LH	RT,14)0,RW*		EAD 597
	STH	RT,NUMSCR		EAD 598
	MVC	NS)12*,16)RW*		EAD 599
	SR	RI,RI		EAD 600
	CHECK	RT,Z,NOSUBS	SEE IF VARIABLE IS AN ARRAY	EAD 601
	LH	RI,LENGTH		EAD 602
	SLA	RT,1		EAD 603
	ST	RT,NUMSCR2		EAD 604
NOSUBS	L	RA,NADD	PICK UP STORAGE ADDRESS	EAD 605
LPAREN	NOP	S423	BRANCH ONLY IF LEFT PARENTHESIS WAS SEEN	EAD 606
NFLAG1	NOP	ACARD1		EAD 607
	B	NEWFIELD		EAD 608
S423	CHECK	RT,Z,ERROR	ERROR IF VARIABLE IS NOT AN ARRAY	EAD 609
	EJECT			EAD 610
*		THE PROCESSING OF SUBSCRIPT FIELDS		EAD 611
		SPACE 2		EAD 612
SUBSCRIPT	MVI	DASHDNE.1,X-FO-		EAD 613
	B	DASHDNE		EAD 614
SUBSC	MVI	DONE1.1,X-DO-		EAD 615
	MVI	SIGN.1,X-CO-		EAD 616
	MVI	DNE.1,X-FO-		EAD 617
	SR	RS,RS		EAD 618
SUBL00P	C0L72	H,ERROR		EAD 619
	COMP	BLANK,E,SUBL00P		EAD 620
	COMP	ZER0,L,ERROR		EAD 621
	LR	RT,RC		EAD 622
NEXTNS	C0L72	H,ERROR		EAD 623
	COMP	ZER0,NL,NEXTNS		EAD 624
	COMP	COMMA,E,ENDSUB		EAD 625
	COMP	RPO26,E,ESUBS		EAD 626
	COMP	RPO29,E,ESUBS		EAD 627
	COMP	BLANK,E,SKIPS		EAD 628
	COMP	MINUS,NE,ERROR		EAD 629
S1048	NOP	ERROR		EAD 630
	MVI	S1048.1,X-FO-		EAD 631
	MVI	ND1.1,X-FO-		EAD 632
	B	ENDSUB		EAD 633
ESURS	MVI	DNE.1,X-00-		EAD 634
ENDSUB	NOP	S1046		EAD 635
	LA	RS,2)0,RS*		EAD 636
	C	RS,NUMSCR2		EAD 637
	BH	ERROR		EAD 638
	BAL	14,LA000C		EAD 639
	STH	RW,SUB-2)RS*		EAD 640
ND1	NOP	S1049		EAD 641
BLNK	NOP	LCOM		EAD 642
DNE	NOP	SUBL00P		EAD 643
PAR	SRA	RS,1		EAD 644
	LR	RW,RS		EAD 645
ND2	NOP	S1047		EAD 646
OUTSUB	C0L72	H,SUBD0NE		EAD 647
	COMP	BLANK,E,SUBD0NE		EAD 648
	COMP	EQ026,E,SUBD0NE		EAD 649
	COMP	EQ029,E,SUBD0NE		EAD 650

	CLI	OIRC*,ZERO	EAD 651
DONE1	BNH	ERROR	EAD 652
	NI	OIRC*,X-OF-	EAD 653
	SR	RW,RW	EAD 654
	IC	RW,OIO,RC*	EAD 655
	CH	RW,NUMSCR	EAD 656
	BH	ERROR	EAD 657
ND3	MVI	DONE1.1,X-F0-	EAD 658
	B	OUTSUB	EAD 659
SKIPS	MVI	BLNK.1,X-F0-	EAD 660
	B	ENDSUB	EAD 661
LCOM	COL72	H,ERROR	EAD 662
	MVI	BLNK.1,X-00-	EAD 663
	COMP	BLANK,E,LCOM	EAD 664
	COMP	COMMA,E,SUBLOOP	EAD 665
	COMP	RPO26,E,PAR	EAD 666
	COMP	RPO29,E,PAR	EAD 667
	COMP	MINUS,E,S1045	EAD 668
	B	ERROR	EAD 669
SUBDNE	LH	RT,SUB	EAD 670
	S	RT,ONE	EAD 671
	LH	RI,LENGTH	EAD 672
	S	RS,ONE	EAD 673
	BZ	ONESUB	EAD 674
	STH	RW,NUMWD	EAD 675
	SR	RW,RW	EAD 676
SL00P	LH	R8,SUB.2IRW*	EAD 677
	S	R8,ONE	EAD 678
	MH	R8,NSIRW*	EAD 679
	AR	RT,R8	EAD 680
	LA	RW,2IO,RW*	EAD 681
	BCT	RS,SL00P	EAD 682
	LH	RW,NUMWD	EAD 683
ONESUB	L	RA,NADD	EAD 684
	MH	RT,LENGTH	EAD 685
	AR	RA,RT	EAD 686
	S	RW,ONE	EAD 687
	BZ	NFLAG2	EAD 688
	SLA	RW,1	EAD 689
	S	RW,TWO	EAD 690
	MH	RI,NSIRW*	EAD 691
NFLAG2	NOP	ACARD1	EAD 692
	COL72	H,NEXTCARD	EAD 693
	B	NEWFIELD	EAD 694
S1045	EX	J,S1048	EAD 695
	MVI	S1048.1,X-F0-	EAD 696
S1049	STH	RW,NMIN	EAD 697
	LR	RW,RS	EAD 698
	SRA	RW,1	EAD 699
	STH	RW,NUMWD	EAD 700
	MVI	ND1.1,X-00-	EAD 701
	MVI	ENDSUB.1,X-F0-	EAD 702
	B	SUBLOOP	EAD 703
S1046	MVI	ENDSUB.1,X-00-	EAD 704
NFLAG3	NOP	ERROR	EAD 705
	MVI	ND2.1,X-F0-	EAD 706
	BAL	14,LA000C	EAD 707
	STH	RW,NMAX	EAD 708
	B	BLNK	EAD 709

S1047	MVI	S104B.1,X-00-	EAD	710
	MVI	ND2.1,X-00-	EAD	711
	LH	RW,NUMWD	EAD	712
	MVI	DASHDNE.1,X-00-	EAD	713
	MVI	DASHA.1,X-00- SET DASH SWITCHES	EAD	714
	MVI	DASHL.1,X-00-	EAD	715
	MVI	DASHH.1,X-00-	EAD	716
	MVI	DASHF.1,X-00-	EAD	717
	MVI	DASHI.1,X-00-	EAD	718
	B	ND3	EAD	719
DASHDNE	MVI	DASHA.1,X-F0- RESET DASH SWITCHES	EAD	720
	MVI	DASHL.1,X-F0-	EAD	721
	MVI	DASHH.1,X-F0-	EAD	722
	MVI	DASHF.1,X-F0-	EAD	723
	MVI	DASHI.1,X-F0-	EAD	724
DASHDNE	NOP	SUBSC	EAD	725
	B	NEWFIELD	EAD	726
	EJECT		EAD	727
*		THE PROCESSING OF HEXADECIMAL DATA	EAD	728
		SPACE 2	EAD	729
HEXCARD	C0L72	H,READCARD	EAD	730
	COMP	BLANK,E,HEXCARD	EAD	731
	LR	RT,RC	EAD	732
STAY	COMP	ZER0,NL,NUMB	EAD	733
	COMP	F,H,ERR0R	EAD	734
	COMP	A,L,ERR0R	EAD	735
	IC	RW,0)0,RC*	EAD	736
	AH	RW,CONV CONVERT -C1- - -C6- TO -FA- - -FF-	EAD	737
	STC	RW,0)0,RC*	EAD	738
NUMB	C0L72	H,STOREH	EAD	739
	COMP	BLANK,NE,STAY	EAD	740
STOREH	LR	RW,RC	EAD	741
	SR	RW,RT	EAD	742
	LH	RB,LENGTH	EAD	743
	SLA	RB,1	EAD	744
	CR	RW,RB	EAD	745
	BH	ERR0R	EAD	746
	SDR	0,0	EAD	747
	STD	0,DB	EAD	748
	STD	0,DOUBLE	EAD	749
	S	RW,0NE	EAD	750
	STC	RW,LENG.1	EAD	751
	NI	PK.1,X-F0-	EAD	752
LENG	0I	PK.1,X-00-	EAD	753
PK	PACK	D8-1)9*,0)0,RT* CONVERT BCD TO HEXADECIMAL	EAD	754
	LM	RT,RW,DB	EAD	755
	SRDL	RT,4 SHIFT OUT SIGN	EAD	756
	STM	RT,RW,DB	EAD	757
	IC	RW,DB-1 INSERT FIRST HALF-BYTE	EAD	758
	SLL	RW,4	EAD	759
	STC	RW,LM0ST.1	EAD	760
LM0ST	0I	D8,X-00-	EAD	761
	L	RS,DB.4	EAD	762
	LD	4,DB	EAD	763
	CLI	LENGTH.1,6 CHECK LENGTH OF VARIABLE	EAD	764
	UL	TIMES4	EAD	765
	MVI	STOREH,X-60- CONVERT STORE INSTRUCTION TO -STD-	EAD	766
STOREH	STD	4,0)0,RA* STORE HEXADECIMAL DATA	EAD	767
	AR	RA,RI INCREMENT STORAGE ADDRESS	EAD	768

DASHH	B	NEXTH		EAD 769
DASHH1	NOP	DASHH2		EAD 770
	LH	RW,NMAX		EAD 771
	SH	RW,NMIN		EAD 772
	A	RW,ONE		EAD 773
	MVI	DASHH1.1,X-F0-		EAD 774
DASHH2	BCT	RW,STORHEX		EAD 775
	MVI	DASHH1.1,X-00-		EAD 776
	MVI	DASHA.1,X-F0-	RESET DASH SWITCHES	EAD 777
	MVI	DASHL.1,X-F0-		EAD 778
	MVI	DASHH.1,X-F0-		EAD 779
	MVI	DASHF.1,X-F0-		EAD 780
	MVI	DASHI.1,X-F0-		EAD 781
NEXTH	CR	RC,RCMAX		EAD 782
	BH	READCARD		EAD 783
	B	HEXCARD		EAD 784
TIMES4	MVI	STORHEX,X-50-	CONVERT STORE INSTRUCTION TO -STE-	EAD 785
	B	STORHEX		EAD 786
	EJECT			EAD 787
*		THE PROCESSING OF ALPHAMERIC DATA		EAD 788
	SPACE	2		EAD 789
ALPHA	LH	RS,LENGTH	PICK UP LENGTH OF VARIABLE	EAD 790
	S	RS,ONE		EAD 791
	STC	RS,MVS.1		EAD 792
	STC	RS,MVS1.1		EAD 793
	C0L72	H,ERROR		EAD 794
	COMP	BLANK,E,NOTHING		EAD 795
	LR	RT,RC		EAD 796
	LR	RW,RC		EAD 797
	AR	RW,RS		EAD 798
MOREA	C0L72	H,STOREA		EAD 799
	COMP	BLANK,E,STOREA		EAD 800
	CR	RC,RW		EAD 801
	BH	ERROR		EAD 802
	B	MOREA		EAD 803
STOREA	LR	RW,RC		EAD 804
	SR	RW,RT		EAD 805
	S	RW,ONE		EAD 806
	STC	RW,MVS1.1		EAD 807
MVS	MVC	010,RA*,BLANKS		EAD 808
MVS1	MVC	010,RA*,0IRT*	STORE ALPHAMERIC DATA	EAD 809
NEXTA	AR	RA,RI	INCREMENT STORAGE ADDRESS	EAD 810
DASHA	B	NEWFIELD		EAD 811
DASHA1	NOP	DASHA2		EAD 812
	LH	RS,NMAX		EAD 813
	SH	RS,NMIN		EAD 814
	A	RS,ONE		EAD 815
	MVI	DASHA1.1,X-F0-		EAD 816
DASHA2	BCT	RS,MVS		EAD 817
	MVI	DASHA1.1,X-00-		EAD 818
	B	DASHDONE		EAD 819
NOTHING	LA	RT,BLANKS		EAD 820
	B	MVS1		EAD 821
	EJECT			EAD 822
*		THE PROCESSING OF LOGICAL DATA		EAD 823
	SPACE	2		EAD 824
LOGICAL	COMP	T,E,TRUE	CHECK FOR -TRUE-	EAD 825
	COMP	F,E,LOGIC-2	CHECK FOR -FALSE-	EAD 826
	COMP	Y,E,TRUE	CHECK FOR -YES-	EAD 827

	COMP	N,NE,ERROR	ERROR IF NOT NO	EAD	828
	SR	RW,RW		EAD	829
LØGIC	CØL72	H,STØREL		EAD	830
	COMP	BLANK,E,STØREL	LOOK FOR A FIELD TERMINATOR	EAD	831
	COMP	COMMA,E,STØREL		EAD	832
	B	LØGIC		EAD	833
TRUE	LA	RW,1		EAD	834
	B	LØGIC		EAD	835
TIMES4L	MVI	STØRLØG,X-50-	CONVERT STØRE INSTRUCTION TO -STE-	EAD	836
	B	STØRLØG		EAD	837
STØREL	CLI	LENGTH,1,3	CHECK LENGTH OF VARIABLE	EAD	838
	BH	TIMES4L		EAD	839
	MVI	STØRLØG,X-42-	CONVERT STØRE INSTRUCTION TO -STC-	EAD	840
STØRLØG	ST	RW,Ø1Ø,RA*	STØRE LØGICAL VALUE	EAD	841
	AR	RA,R1	INCREMENT STØRAGE ADDRESS	EAD	842
DASHL	B	NEWFIELD		EAD	843
DASHL1	NØP	DASHL2		EAD	844
	LH	RS,NMAX		EAD	845
	SH	RS,NMIN		EAD	846
	A	RS,ØNE		EAD	847
	MVI	DASHL1,1,X-FØ-		EAD	848
DASHL2	BCT	RS,STØRLØG		EAD	849
	MVI	DASHL1,1,X-ØØ-		EAD	850
	B	DASHØNE		EAD	851
	EJECT			EAD	852
*		THIS ROUTINE CONVERTS BCD TO A SIGNED BINARY INTEGER		EAD	853
		SPACE 2		EAD	854
*		ADDRESS OF FIRST NUMERICAL CHARACTER IN RT		EAD	855
*		ADDRESS OF NEXT BYTE AFTER LAST CHARACTER IN RC		EAD	856
*		BINARY INTEGER ANSWER IN RW		EAD	857
*		TO INSERT MINUS SIGN, USE MVI SIGN,1,X-DØ-		EAD	858
*		TO INSERT PLUS SIGN, USE MVI SIGN,1,X-CØ-		EAD	859
LAØØC	LR	RW,RC		EAD	860
	SR	RW,RT		EAD	861
	S	RW,ØNE		EAD	862
	STC	RW,LEN,1	INSERT FIELD LENGTH INTO PACK INSTRUCTION	EAD	863
	NI	PACK,1,X-FØ-		EAD	864
LEN	ØI	PACK,1,X-ØØ-		EAD	865
	LR	RW,RC	GET ADDRESS OF RIGHTMOST BYTE	EAD	866
	S	RW,ØNE		EAD	867
	NI	Ø1RW*,X-ØF-	CLEAR ZØNE FOR SIGN	EAD	868
SIGN	ØI	Ø1RW*,X-CØ-	INSERT SIGN	EAD	869
	SDR	Ø,Ø		EAD	870
	STD	Ø,D8		EAD	871
PACK	PACK	D81Ø*,Ø1Ø,RT*	PACK BCD	EAD	872
	CVB	RW,D8	CONVERT TO BINARY	EAD	873
	ØR	14	RETURN	EAD	874
	EJECT			EAD	875
*		THIS ROUTINE CONVERTS A SIGNED BINARY INTEGER TO DOUBLE		EAD	876
*		PRECISION FLOATING POINT, USING THE SAME PROCEDURE AS FORTRAN		EAD	877
		SPACE 2		EAD	878
*		BINARY INTEGER IN RW		EAD	879
*		FLOATING POINT ANSWER IN REGISTER 0		EAD	880
LACØØF	LD	Ø,CFLØAT		EAD	881
	STD	Ø,D8		EAD	882
	LTR	RW,RW		EAD	883
	ØM	*,16		EAD	884
	ST	RW,D8,4		EAD	885
	AD	Ø,D8		EAD	886

	B	*.14		EAD	887
	LPR	RW,RW		EAD	888
	ST	RW,D8.4		EAD	889
	SD	0,D8		EAD	890
	BCR	15.14		EAD	891
	EJECT			EAD	892
*	ERROR PROCESSING			EAD	893
	SPACE	2		EAD	894
ERROR1	LA	RW,FE1	ILLEGAL ENTRY TO SETUP OR READIN	EAD	895
	ST	RW,FERROR		EAD	896
	B	ERR		EAD	897
ERROR2	MVC	NAME18*,D8	NAME OF VARIABLE NOT FOUND IN TABLE	EAD	898
	LA	RW,FE2		EAD	899
	ST	RW,FERROR		EAD	900
	B	ERR		EAD	901
ERROR	LA	RW,FE	ERROR IN FORMAT ON LAST CARD	EAD	902
	ST	RW,FERROR		EAD	903
ERR	L	15,VIBCOM	WRITE ERROR MESSAGE	EAD	904
	CNOP	0.4		EAD	905
	BAL	14.4)0,15*		EAD	906
	DC	F-6-		EAD	907
FERROR	DC	A)FE*		EAD	908
	BAL	14.16)0,15*		EAD	909
LEAVE	L	15,VEXIT	CALL EXIT	EAD	910
	BALR	14.15		EAD	911
	EJECT			EAD	912
*	END-OF-DATA PROCESSING			EAD	913
	SPACE	2		EAD	914
IEND	B	LEAVE	END-OF-JOB CARD	EAD	915
	LA	15.4	SET UP NON-STANDARD RETURN	EAD	916
	B	RESTORE.2		EAD	917
RESTORE	SR	15.15	NORMAL RETURN TO CALLING PROGRAM	EAD	918
	L	13,SAVE.4	RESTORE OLD SAVE AREA ADDRESS	EAD	919
	L	RW,ADINC0L1		EAD	920
	L	RT,INC0L1		EAD	921
	ST	RT,0)0,RW*		EAD	922
	LM	2,12,28)13*	RESTORE REGISTERS	EAD	923
	L	14.12)0,13*	PICK UP RETURN ADDRESS	EAD	924
	MVI	12)13*,X-FF-	INSERT RETURN FLAG	EAD	925
	BR	14	RETURN	EAD	926
	DROP	RD	DON'T LET ASSEMBLER USE RD AS A BASE	EAD	927
	EJECT			EAD	928
*	SUBROUTINE HEDING)TAPE*			EAD	929
*	THIS ROUTINE EJECTS A PAGE AND WRITES THE H-CARD			EAD	930
	SPACE	2		EAD	931
	USING	*.12	INFORM ASSEMBLER OF BASE REGISTER	EAD	932
HEDING	STM	12.14,12)13*	SAVE REGISTERS	EAD	933
	LR	12.15	PICK UP BASE REGISTER	EAD	934
	ST	13,SAVE.4	SAVE ADDRESS OF OLD SAVE AREA	EAD	935
	L	13.0)0,1*	PICK UP ADDRESS OF TAPE NUMBER	EAD	936
	L	13.0)0,13*	PICK UP TAPE NUMBER	EAD	937
	ST	13,TAPEN0	INSERT IT IN ARG LIST FOR IBCOM	EAD	938
	LA	13,SAVE	PICK UP ADDRESS OF NEW SAVE AREA	EAD	939
	L	15,VIBCOM		EAD	940
	CNOP	0.4		EAD	941
	BAL	14.4)0,15*	FLIP PAGE AND WRITE HEADING	EAD	942
TAPEN0	DC	F-6-		EAD	943
	DC	A)FH*		EAD	944
	BAL	14.16)0,15*		EAD	945

```

L      13,SAVE,4      PICK UP ADDRESS OF OLD SAVE AREA      EAD 946
LM     12,14,12)13*  RESTORE REGISTERS                      EAD 947
MVI   12)13*,X=FF=  INSERT RETURN FLAG                    EAD 948
BR    14              RETURN                                EAD 949
EJECT
*     SUBROUTINE WHERE)TABLE*                               EAD 950
*     THIS ROUTINE PICKS UP THE LOCATION OF -TABLE- FOR -SETUP- EAD 951
*     -WHERE- HAS BEEN MODIFIED 5-2-68                     EAD 952
*     TO ACTIVATE MIBCOM. )W. SOHN*                        EAD 954
*
WHERE  EXTRN MIBCOM                                         EAD 955
      BC  15,12)15*                                         EAD 956
      DC  X=7=                                              EAD 957
      DC  CL7-WHERE =                                       EAD 958
*     SAVE PROCEDURE                                       EAD 959
      STM 14,12,12)13*                                     EAD 960
      BALR 10,0                                             EAD 961
      USING *,10                                           EAD 962
      LR  2,13                                             EAD 963
      LA  13,SAVEREG                                       EAD 964
      ST  13,8)2*                                          EAD 965
      ST  2,4)13*                                          EAD 966
      BC  15,FIRST                                         EAD 967
SAVEREG DS 18F                                             EAD 968
*     END SAVE PROCEDURE                                    EAD 969
FIRST  NOP NOTFIRST                                       EAD 970
      L   RT,0)1)*                                         EAD 971
      ST RT,TABLE                                         EAD 972
      ST RT,STABLE                                        EAD 973
      MVI FIRST,1,X=FO=                                    EAD 974
      LA  1,ARGULST                                        EAD 975
      L   15,ADCON                                        EAD 976
      BALR 14,15                                          EAD 977
*     RETURN PROCEDURE                                     EAD 978
NOTFIRST L 13,SAVEREG,4                                    EAD 979
      LM  2,12,28)13*                                     EAD 980
      L   14,12)13*                                       EAD 981
      MVI 12)13*,X=FF=                                    EAD 982
      BCR 15,14                                           EAD 983
*     END RETURN PROCEDURE                                EAD 984
ADCON  DC A)MIBCOM*                                       EAD 985
*     ARGUMENT LIST                                       EAD 986
ARGULST DC X=80=                                         EAD 987
      DC  AL3)**                                          EAD 988
*     END SUBROUTINE WHERE                                EAD 989
EJECT
*     SUBROUTINE SETUP)NAME,LENGTH,VARIABLE,DIMENSIONS,....* EAD 991
*     THIS ROUTINE CONSTRUCTS THE TABLE OF INPUT VARIABLES EAD 992
*     WORD          CONTENTS                               EAD 993
*     *****                                           EAD 994
*     *                                                    * EAD 995
*     *                                                    * EAD 996
*     1 *                                                    * EAD 997
*     *          EBCDIC NAME OF VARIABLE                    * EAD 998
*     2 *                                                    * EAD 999
*     *          *****                                   * EAD 1000
*     *          *****                                   * EAD 1001
*     *          *****                                   * EAD 1002
*     3 *          CORE ADDRESS OF VARIABLE                 * EAD 1003
*     *          *****                                   * EAD 1004

```

*	*****	EAD 1005
*	*	EAD 1006
4	* LENGTH SPECIFICATION * NUMBER OF SUBSCRIPTS *	EAD 1007
*	*	EAD 1008
*	*****	EAD 1009
*	*	EAD 1010
5	* N1=NS)1* * N1*N2=NS)2*	EAD 1011
*	*	EAD 1012
*	*****	EAD 1013
*	*	EAD 1014
6	* N1*N2*N3=NS)3* * N1*N2*N3*N4=NS)4*	EAD 1015
*	*	EAD 1016
*	*****	EAD 1017
*	*	EAD 1018
7	* N1*N2*N3*N4*N5=NS)5* * N1*N2*N3*N4*N5*N6=NS)6*	EAD 1019
*	*	EAD 1020
*	*****	EAD 1021
	SPACE 2	EAD 1022
	USING *,15	EAD 1023
SETUP	STM R6,R12,44)13* SAVE REGISTERS	EAD 1024
CHECK	NOP G0 BRANCH AROUND IF NOT THE FIRST TIME THROUGH	EAD 1025
	L R11,STABLE	EAD 1026
	CHECK R11,2,ERRS SEE IF -WHERE- HAS BEEN ENTERED	EAD 1027
	MVI CHECK,1,X-F0-	EAD 1028
G0	L R9,TABLE PICK UP LOCATION FOR NEXT TABLE ENTRY	EAD 1029
	L R11,0)0,1* PICK UP ADDRESS OF VARIABLE NAME	EAD 1030
	MVC 0)8,R9*,0)R11* MOVE NAME INTO TABLE	EAD 1031
	L R11,4)0,1*	EAD 1032
	L R11,0)0,R11* PICK UP LENGTH OF VARIABLE	EAD 1033
	STH R11,12)0,R9* INSERT IT INTO TABLE	EAD 1034
	L R11,8)0,1* PICK UP ADDRESS OF VARIABLE	EAD 1035
	ST R11,8)0,R9* INSERT IT INTO TABLE	EAD 1036
	SR R12,R12	EAD 1037
	CLI 8)1*,X-80- LEAVE IF IT WAS THE LAST ARGUMENT	EAD 1038
	BE ALLDONE	EAD 1039
	LR R7,1	EAD 1040
	L R11,0NE	EAD 1041
	LA R6,14	EAD 1042
LOOP	LA R12,1)0,R12*	EAD 1043
	CLI 12)R7*,X-80- LEAVE IF THIS IS THE LAST ARGUMENT	EAD 1044
	BE ALLDONE	EAD 1045
	C R12,C7 LEAVE IF SEVEN SUBSCRIPTS	EAD 1046
	BE ALLDONE	EAD 1047
	L R8,12)0,R7* PICK UP DIMENSION	EAD 1048
	MH R11,2)0,R8* COMPUTE DIMENSION FACTOR	EAD 1049
	LA R7,4)0,R7*	EAD 1050
	LA R6,2)0,R6*	EAD 1051
	STH R11,0)R6,R9* INSERT DIMENSION FACTOR INTO TABLE	EAD 1052
	B LOOP	EAD 1054
ALLDONE	STH R12,14)0,R9* INSERT NUMBER OF SUBSCRIPTS INTO TABLE	EAD 1055
	LA R9,28)0,R9* OBTAIN LOCATION OF NEXT TABLE ENTRY	EAD 1056
	ST R9,TABLE	EAD 1057
	LH R11,NUMTAB INCREMENT NUMBER OF TABLE ENTRIES	EAD 1058
	LA R11,1)0,R11*	EAD 1059
	STH R11,NUMTAB	EAD 1060
	LM R6,R12,44)13* RESTORE REGISTERS	EAD 1061
	BR 14 RETURN	EAD 1062
ERRS	L 12,AREAD ERROR IF WHERE NOT ENTERED BEFORE SETUP	EAD 1063
	L RD,AREAD,4	EAD 1064

	USING READIN,12		
	B	ERROR1	EAD 1065
	DRØP	12	EAD 1066
	DS	ØF	EAD 1067
AREAD	DC	AIREADIN*	EAD 1068
	DC	AIDATA*	EAD 1069
	EJECT		EAD 1070
*	REGISTER ASSIGNMENT		EAD 1071
	SPACE	2	EAD 1072
RI	EQU	Ø	EAD 1073
RA	EQU	1	EAD 1074
RT	EQU	2	EAD 1075
RW	EQU	3	EAD 1076
RS	EQU	4	EAD 1077
R5	EQU	5	EAD 1078
RC	EQU	6	EAD 1079
R6	EQU	6	EAD 1080
R7	EQU	7	EAD 1081
R8	EQU	8	EAD 1082
RCMAX	EQU	9	EAD 1083
R9	EQU	9	EAD 1084
RD	EQU	10	EAD 1085
R11	EQU	11	EAD 1086
R12	EQU	12	EAD 1087
	EJECT		EAD 1088
*	BCD AND EBCDIC CHARACTERS		EAD 1089
	SPACE	2	EAD 1090
BLANK	EQU	C- -	EAD 1091
A	EQU	C-A-	EAD 1092
D	EQU	C-D-	EAD 1093
E	EQU	C-E-	EAD 1094
F	EQU	C-F-	EAD 1095
H	EQU	C-H-	EAD 1096
N	EQU	C-N-	EAD 1097
P	EQU	C-P-	EAD 1098
T	EQU	C-T-	EAD 1099
X	EQU	C-X-	EAD 1100
Y	EQU	C-Y-	EAD 1101
Z	EQU	C-Z-	EAD 1102
DØLLAR	EQU	C-\$-	EAD 1103
STAR	EQU	C-*	EAD 1104
PLUSØ26	EQU	C-++-	EAD 1105
PLUSØ29	EQU	C-.-	EAD 1106
MINUS	EQU	C--	EAD 1107
DECIMAL	EQU	C-.-	EAD 1108
CØMMA	EQU	C-.-	EAD 1109
ZERØ	EQU	C=Ø-	EAD 1110
EQØ26	EQU	C=Ø-	EAD 1111
EQØ29	EQU	C=Ø-	EAD 1112
LPØ26	EQU	C= -	EAD 1113
LPØ29	EQU	C= -	EAD 1114
RPØ26	EQU	C= -	EAD 1115
RPØ29	EQU	C= -	EAD 1116
SLASH	EQU	C=/-	EAD 1117
	EJECT		EAD 1118
*	PRØGRAM DATA		EAD 1119
	SPACE	2	EAD 1120
DATA	DS	ØØ	EAD 1121
CARD	DS	1ØØ	EAD 1122
		CARD CØLUMNS 1 - 8Ø	EAD 1123

CFLDPT	DC	XL8=4E00000000000000- FOR INTEGER TO FLOATING POINT	EAD 1124
TEN	DC	D=10-	EAD 1125
TENT	DC	D=1,1.E-1,1.E-2,1.E-3,1.E-4,1.E-5,1.E-6,1.E-7-	EAD 1126
	DC	D=1.E-8,1.E-9,1.E-10,1.E-11,1.E-12,1.E-13,1.E-14-	EAD 1127
	DC	D=1.E-15,1.E-16,1.E-17,1.E-18,1.E-19,1.E-20-	EAD 1128
DOUBLE	DS	D THESE TWO TEMPORARIES MUST	EAD 1129
DB	DS	D REMAIN CONTIGUOUS TO EACH OTHER	EAD 1130
ONE	DC	F=1-	EAD 1131
TWO	DC	F=2-	EAD 1132
C7	DC	F=7-	EAD 1133
NINE	DC	F=9-	EAD 1134
C72	DC	F=72-	EAD 1135
C888	DC	F=8888-	EAD 1136
ARGLIST	DC	A)TEN*	EAD 1137
	DC	A)D8*	EAD 1138
VFDXPI	DC	A)FDXPI**	EAD 1139
VEXIT	DC	A)EXIT*	EAD 1140
VIUCOM	DC	A)IUCOM**	EAD 1141
INCOL1	DC	F=0-	EAD 1142
JAN	DC	F=31,59,90,120,151,181,212,243,273,304,334,365-	EAD 1143
DATE	DS	F	EAD 1144
SAVE	DS	18F	EAD 1145
ADINCOL1	DS	F	EAD 1146
STABLE	DC	F=0-	EAD 1147
TABLE	DC	F=0-	EAD 1148
NADD	DC	F=0-	EAD 1149
C3L	DC	F=0-	EAD 1150
COLN0	DC	F=0-	EAD 1151
NUMSCR2	DC	F=0-	EAD 1152
	LTORG		EAD 1153
CONV	DC	H=57-	EAD 1154
NUMTAB	DC	H=0-	EAD 1155
NUMWD	DC	H=0-	EAD 1156
NS	DC	6H=0-	EAD 1157
LENGTH	DC	H=0-	EAD 1158
NUMSCR	DC	H=0-	EAD 1159
SUB	DS	7H	EAD 1160
NMIN	DC	H=0-	EAD 1161
NMAX	DC	H=0-	EAD 1162
BLANKS	DC	CL8-	EAD 1163
NOFLIP	DC	CL6=NOFLIP-	EAD 1164
FLIP	DC	CL4=FLIP-	EAD 1165
NOPRINT	DC	CL7=NOPRINT-	EAD 1166
PRINTS	DC	CL5=PRINT-	EAD 1167
	EJECT		EAD 1168
*	FORMAT STATEMENTS		EAD 1169
	SPACE 2		EAD 1170
FH	DC	XL3=021A50-	EAD 1171
HEADING	DC	XL1=F1-	EAD 1172
	DC	9CL8-	EAD 1173
	DC	CL7-	EAD 1174
	DC	XL1=22-	EAD 1175
F1	DC	XL4=021A11F1-	EAD 1176
	DC	C=INPUT CARDS READ-	EAD 1177
	DC	XL2=1832- 50 BLANKS	EAD 1178
	DC	XL2=1A10- FORMAT FOR TIME OF DAY FIELD	EAD 1179
MRS	DS	CL2 HOURS	EAD 1180
	DC	CL1-	EAD 1181
MNS	DS	CL2 MINUTES	EAD 1182

SCS	DC	CL1--		EAD 1183
	DS	CL2	SECONDS	EAD 1184
HSC	DC	CL1--		EAD 1185
	DS	CL2	HUNDREDTHS OF A SECOND	EAD 1186
	DC	CL5-		EAD 1187
MTH	DC	XL2-1A0B-	FORMAT FOR DATE FIELD	EAD 1188
	DS	CL2	MONTH	EAD 1189
DAY	DC	CL1-/-		EAD 1190
	DS	CL2	DAY	EAD 1191
YR	DC	CL1-/-		EAD 1192
	DS	CL2	YEAR	EAD 1193
F2	DC	XL1-22-	END OF FORMAT	EAD 1194
	DC	XL6-02060A140822-		EAD 1195
F3	DC	XL3-021A06-		EAD 1196
	DC	CL6- DATA*-		EAD 1197
	DC	XL6-060A14081A05-		EAD 1198
	DC	CL5-*DATA-		EAD 1199
FE2	DC	XL1-22-		EAD 1200
	DC	XL3-021A1E-		EAD 1201
NAME	DC	CL13-0 NO SYMBOL **		EAD 1202
	DC	CL8-		EAD 1203
	DC	CL9== LISTED.-		EAD 1204
FE1	DC	XL1-22-		EAD 1205
	DC	XL3-021A44-		EAD 1206
	DC	CL32-0 WHERE NOT ENTERED BEFORE SETUP-		EAD 1207
FE	DC	CL36- OR SETUP NOT ENTERED BEFORE READIN.-		EAD 1208
	DC	XL1-22-		EAD 1209
	DC	XL3-021A1F-		EAD 1210
	DC	CL31-0 ERROR IN FORMAT ON LAST CARD.-		EAD 1211
	DC	XL1-22-		EAD 1212
	END			EAD 1213

IMPROVED DATA								PRESE 1
ERNRTR	0.	2.26D-6	1.7D-5	4.26D-5	1.11D-4	.00033		PRESE 2
.001068	.01171	.0804	1000000.					PRESE 3
ERNRTR	0.	10.	14.	18.	22.	26.	30.	1000000.
ERNRTR								PRESE 4
(1.01)1	100.	1.15D6	4.0D9	2.1D11	7.0D11	1.3D12	2.0D12	2.0D12
(1.02)1	100.	1.15D6	4.0D9	2.1D11	7.0D11	1.3D12	2.0D12	2.0D12
(1.03)1	100.	2.0D6	2.0D10	7.2D11	3.1D12	7.2D12	1.0D13	1.0D13
(1.04)1	100.	4.3D7	1.0D11	3.5D12	1.5D13	4.0D13	7.0D13	7.0D13
(1.05)1	100.	6.7D8	7.7D11	2.0E13	8.8D13	2.1D14	3.5D14	3.5D14
(1.06)1	100.	1.2D10	2.2D12	1.7D14	7.4D14	1.2D15	2.0D15	2.0D15
(1.07)1	100.	6.0D11	1.1D14	1.9D15	7.0D15	1.9D16	2.9D16	2.8D16
(1.08)1	100.	2.3D12	8.1D14	1.4D16	9.4D16	1.9D17	2.2D17	2.2D17
(1.09)1	100.	1.4D13	5.0D15	2.1D16	3.6D17	1.0D18	2.0D18	2.0D18
(1.10)1	100.	1.4D13	5.0D15	2.1D16	3.6D17	1.0D18	2.0D18	2.0D18
RSTABL	0	1.E-6	1.F-5	1.E-4	1.E-3	1.E-2	1	100.
RSTABL	0.	30.120.1450.	470.	1.E10				PRESE 16
ERNRTR								PRESE 17
(1.9)1	0.0	0.72E9	0.25E14	0.20E15	0.79E16	0.35E17	0.13E18	PRESE 19
0.34E18	0.62E18	0.90E18	0.14E19	0.17E19	0.21E19	0.25E19	0.36E19	PRESE 20
0.38E19	0.47E19	0.56E19	0.63E19	0.70E19	0.86E19	0.20E19		PRESE 21
0.12E20	0.14E20	0.14E20						PRESE 22
(1.6)1	0.							PRESE 23
0.72E7	0.25E12	0.20E13	0.78E14	0.35E15	0.13E16	0.34E16		PRESE 24
0.62E16	0.90E16	0.14E17	0.17E17	0.21E17	0.25E17	0.36E17	0.38E17	PRESE 25
0.47E17	0.56E17	0.63E17	0.70E17	0.86E17	0.20E17	0.12E18		PRESE 26
0.14E18	0.14E18							PRESE 27
(1.7)1	0.							PRESE 28
0.45E7	0.48E11	0.91E12	0.56E13	0.29E14	0.14E15	0.34E15		PRESE 29
0.48E15	0.91E15	0.17E16	0.14E16	0.17E16	0.21E16	0.24E16	0.28E16	PRESE 30
0.29E16	0.42E16	0.46E16	0.57E16	0.72E16	0.83E16	0.94E16		PRESE 31
0.14E17	0.14E17							PRESE 32
(1.6)1	0.							PRESE 33
0.19E7	0.43E10	0.60E11	0.37E12	0.27E13	0.13E14	0.30E14		PRESE 34
0.43E14	0.68E14	0.86E14	0.10E15	0.13E15	0.15E15	0.17E15	0.20E15	PRESE 35
0.26E15	0.28E15	0.34E15	0.38E15	0.52E15	0.62E15	0.72E15	0.10E16	PRESE 36
0.10E16								PRESE 37
(1.5)1	0.							PRESE 38
0.58E6	0.15E8	0.28E10	0.23E11	0.26E12	0.13E13	0.24E13		PRESE 39
0.15E13	0.50E13	0.62E13	0.76E13	0.94E13	0.10E14	0.12E14		PRESE 40
0.14E14	0.16E14	0.19E14	0.24E14	0.28E14	0.33E14	0.45E14		PRESE 41
0.62E14	0.74E14	0.74E14						PRESE 42
(1.4)1	0.							PRESE 43
0.92E5	0.16E8	0.16E9	0.14E10	0.32E11	0.10E12			PRESE 44
0.16E12	0.27E12	0.36E12	0.44E12	0.53E12	0.60E12	0.74E12		PRESE 45
0.89E12	0.97E12	0.11E13	0.13E13	0.15E13	0.23E13	0.26E13	0.35E13	PRESE 46
0.26E13	0.45E13	0.45E13						PRESE 47
(1.3)1	0.							PRESE 48
0.52E4	0.94E6	0.60E7	0.12E9	0.31E10	0.82E10	0.14E11		PRESE 49
0.12E11	0.27E11	0.31E11	0.40E11	0.44E11	0.51E11	0.59E11	0.68E11	PRESE 50
0.80E11	0.95E11	0.12E12	0.16E12	0.20E12	0.25E12	0.28E12		PRESE 51
0.42E12	0.42E12							PRESE 52
(1.2)1	0.							PRESE 53
0.80E3	0.42E5	0.32E6	0.25E7	0.32E9	0.71E9	0.11E10		PRESE 54
0.15E10	0.18E10	0.23E10	0.27E10	0.31E10	0.42E10	0.52E10	0.60E10	PRESE 55
0.75E10	0.90E10	0.12E11	0.14E11	0.17E11	0.24E11	0.29E11	0.33E11	PRESE 56
0.33E11								PRESE 57
(1-30.1)1	0.							PRESE 58
EMCTBL	1.	3.(3.1)30.	1.E6					PRESE 59
THTTBL	0.	2.(4.1)30.	45.	1.E6				PRESE 60

NOT REPRODUCIBLE

(1.111)	1.0	3.0	6.0	9.0	12.0	15.0	18.0	21.0	24.0	27.0	PRESE 61
30.0	100.0										PRESE 62
(1.211)	1.13	3.11	6.30	9.64	13.1	16.8	20.6	24.6	28.8	33.4	PRESE 64
35.0	100.0										PRESE 65
(1.311)	1.29	3.33	7.00	11.2	16.2	22.0	29.0	37.7	48.6	63.0	PRESE 66
41.0	100.0										PRESE 67
(1.411)	1.43	3.58	7.84	13.4	21.0	33.4	49.2	80.0	100.0		PRESE 68
47.0	100.0	100.0									PRESE 69
(1.511)	1.57	3.85	8.89	16.7	29.8	58.0	100.0	100.0	100.0		PRESE 70
53.0	100.0	100.0									PRESE 71
(1.611)	1.70	4.15	10.2	21.8	51.0	100.0	100.0	100.0	100.0		PRESE 72
59.0	100.0	100.0									PRESE 73
(1.711)	1.84	4.49	12.0	31.6	100.0	100.0	100.0	100.0			PRESE 74
65.0	100.0	100.0	100.0								PRESE 75
(1.811)	2.00	4.88	14.6	53.0	100.0	100.0	100.0	100.0	100.0		PRESE 76
71.0	100.0	100.0									PRESE 77
(1.911)	2.13	5.32	18.4	100.0	100.0	100.0	100.0	100.0	100.0		PRESE 78
77.0	100.0	100.0									PRESE 79
(1.101)	2.76	7.79	100.0	100.0	100.0	100.0	100.0	100.0	100.0		PRESE 80
83.0	100.0	100.0									PRESE 81
(1.111)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0		PRESE 82
100.0	100.0	100.0	100.0								PRESE 83
XNTABL	.0000001	.01		.03	.1	1.E6					PRESE 84
YNTABL	0.	15.	20.(20.)160.	1.E6							PRESE 85
ZNTABL	0.	.01	.1	1.							PRESE 86
NTABL											PRESE 87
(1.5,111)	0.0	0.70E1	0.38E3	0.32E10	0.25E12	0.32E13	0.17E14				PRESE 88
1.4E14	0.24E15	0.44E15	0.44E15								PRESE 89
(1.4,111)	0.0	0.50E1	0.38E3	0.32E10	0.25E12	0.32E13	0.17E14				PRESE 90
1.3E14	0.24E15	0.44E15	0.44E15								PRESE 91
(1.1,111)	0.0	0.32E2	0.12E4	0.11E10	0.79E11	0.10E13	0.65E13				PRESE 92
1.2E14	0.37E14	0.16E15	0.16E15								PRESE 93
(1.2,111)	0.0	0.95E2	0.30E4	0.20E9	0.05E11	0.14E12	0.20E13				PRESE 94
1.1E14	0.22E14	0.38E14	0.38E14								PRESE 95
(1.1,111)	0.0	0.90E2	0.30E4	0.20E9	0.05E11	0.14E12	0.20E13				PRESE 96
1.7E14	0.32E14	0.38E14	0.38E14								PRESE 97
(1.5,111)	0.0	1.0E5	3.5E7	2.0E11	1.3E12	2.0E12	4.5E12	1.4E13			PRESE 98
1.4E13	3.0E13	4.0E13									PRESE 99
(1.4,111)	0.0	1.0E5	3.5E7	2.0E11	1.3E12	2.0E12	4.5E12	1.4E13			PRESE100
2.4E13	3.0E13	4.0E13									PRESE101
(1.0,211)	0.0	5.5E4	1.9E7	1.1E11	2.3E11	4.5E11	2.0E12	5.0E12			PRESE102
1.1E12	1.5E13	2.5E13									PRESE103
(1.0,211)	0.0	2.0E4	7.0E6	4.0E10	1.6E11	2.0E11	6.0E11	1.4E12			PRESE104
1.1E12	3.0E12	4.0E12									PRESE105
(1.1,211)	0.0	2.0E4	7.0E6	4.0E10	1.6E11	2.0E11	6.0E11	1.4E12			PRESE106
2.1E12	3.0E12	4.0E12									PRESE107
(1.5,111)	0.0	2.4E5	1.5E8	7.6E11	5.6E11	1.3E13	1.6E13	2.1E13			PRESE108
3.1E13	4.0E13	4.5E13									PRESE109
(1.4,111)	0.0	12.0E5	1.1E8	7.6E11	5.6E12	1.3E13	1.6E13	2.1E13			PRESE110
3.1E13	4.0E13	4.5E13									PRESE111
(1.0,311)	0.0	1.2E5	7.1E7	3.7E11	2.4E12	5.8E12	6.2E12	7.7E12			PRESE112
1.1E13	1.5E13	3.0E13									PRESE113
(1.2,211)	0.0	4.5E4	2.7E7	1.4E11	7.1E11	1.7E12	1.7E12	2.0E12			PRESE114
2.7E12	3.5E12	4.0E12									PRESE115
(1.1,311)	0.0	4.5E4	2.7E7	1.0E11	7.1E11	1.7E12	1.7E12	2.0E12			PRESE116
3.7E12	3.5E12	4.0E12									PRESE117
(1.1,111)	0.0	7.9E5	2.3E8	2.0E12	1.8E13	6.0E13	1.4E14	1.3E14			PRESE118
1.7E14	1.5E14	2.0E14									PRESE119

NOT REPRODUCIBLE

(1.4.4)1	0.0	7.9E5	2.3E8	2.0E12	1.8E13	6.0E13	1.4E14	1.3E14	PRESE120
1.2E14	1.5E14	2.0E14							PRESE121
(1.3.4)1	0.0	5.2E5	1.5E8	1.3E12	8.0E12	3.0E13	5.7E13	5.2E13	PRESE122
4.7E13	5.0E13	6.0E13							PRESE123
(1.2.4)1	0.0	1.7E5	5.0E7	4.3E11	2.5E12	1.1E13	1.5E13	1.3E13	PRESE124
1.2E13	1.3E13	2.0E13							PRESE125
(1.1.4)1	0.0	1.7E5	5.0E7	4.3E11	2.5E12	1.1E13	1.5E13	1.3E13	PRESE126
1.2E13	1.3E13	2.0E13							PRESE127
CNUMB(1)	0.02	2.0	1.0	0.0	0.5				PRESE128
CNUMB(59)	0.66	1.2 (64)	0.0	0.0	0.25	0.5			PRESE129
(83)	4.4E26	0.0	1.0	0.0	1.0	0.0	1.0		PRESE130
(84)	1.0	6.3246E14	0.0	1.0	(100)	0.0	(115)	1.0E11	PRESE131
(116)	1000.0	1.0	1.0	1.0E3	0.0	0.0	1.0		PRESE132
(123)	5.0E-12	-1.5	1.0	(130)	1.0	0.0	1.0	0.0 2.0 1.0E-10 0.0	PRESE133
(159)	0.66	120.0 (164)	0.04	86.0	(169)	1.0			PRESE134
ACCN	1.								PRESE135
AKN	50.								PRESE136
BCCN	1.								PRESE137
B21	1.								PRESE138
B22	.25								PRESE139
B23	0.0								PRESE140
CCCN	1.								PRESE141
CRHOW	.75								PRESE142
DELWH	.01	DHCHM	0.0						PRESE143
RHPSL	.08042								PRESE144
RHOW	115.								PRESE145
RT	8.475D5								PRESE146
TABL	15.0.								PRESE147
UZFRQ	5.8E-21	B2	4.0E-10	U3	2.0	BTWEN	1.0	B24 1.0E-26	PRESE148
DX	50.0	NSTML	100	ZNUS	2.0E11	CNE	0.0	DSD 0.0 X2BDD 0.0 X3B 0.0	PRESE149
INDZ	0	IMPRNT	0	IND	0	IWAKE	2	WKALT 500000.0 0.0 IDBL 4	PRESE150
MTAZ	22.0	22.0	PHI1	6.0	6.0	PHI2	6.0	6.0 PHI3 6.0 6.0	PRESE151
FRQ1	4.35E8	FRQ2	1.375E9	FRQ3	5.4E9	SIGNL1	1.0E-6		PRESE152
SIGNL2	1.0E-6	SIGNL3	1.0E-6	TAU1	1.0	TAU2	0.4	TAU3 0.4	PRESE153
ICOM(1)	180000.0	XCOM(7)	1.0	ICOM(4)	1	1	1	ACOE 1.0 PFD 0.03	PRESE154
(1-9)	40.0								PRESE155
(122-9)	0								PRESE156
(17)	126	135	134	XCOM(8)	1.0				PRESE157
(79)	1								PRESE158

PRINTING OF THE PRESET INPUT DECK

NOT REPRODUCIBLE

APPENDIX II - Overlay Information and Other Pertinent
Information for running the Optimum Decoy
Design Program

SET UP OF BINARIES FOR THE OVERLAY

Program 2542 has been run successfully on the OS65 at Avco using version 14 of the operating system.

All binaries whose names do not follow one of the OVERLAY cards should be first in order in the binary deck.

These binaries should then be followed by the following six overlay segments. The names of the binaries included in a particular segment follow the overlay card for that segment.

OVERLAY A1

READIT

SR2490

ZPRM

ZPRS

ZREADX

OVERLAY A1

VIXEN

OVERLAY A1

ADD

INFCOF

INTGRL

WRITEM

OVERLAY A2

DAVDØN

READY

OVERLAY A2

GIMAX

MIMAX

OVERLAY A2

GRAM

RØSBRK

OTHER INFORMATION

- 1.0 A Preset Deck is needed to run the program.
- 2.0 The subroutine LA000000 is written in Assembly Language.
- 3.0 The input quantity IOP(76) should be set to 1 at AVCO and to 0 at Aerospace.
- 4.0 The dummy subroutine PLT must be removed at Aerospace.
- 5.0 The dummy subroutines ENDJOB, EZPLOT, FRAMEV, IDFRMV, and PLTND must be removed at AVCO.