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	

SAMSO TR-69-140, APP. I

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

Prepared for

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

FOR OFFICIAL USE ONLY

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

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

PART II

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 DØ loop ending at statement 130 is entered to perform the calculations needed to evaluate the polynomial. The value of this polynmial 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. PØLCAL is called by SUBROUTINE MISC only.

B. PØLCAL 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 used to constrain the decoy within specified corridors and the effectiveness integrals which are used in the probability of discrimination calculations in SUBROUTINE EFFECT.

SUBRØUTINE F123(XXX)

1. Purpose

F123 can optionally

a. Read the data for a reentry vehicle trajectory from input or call SUBRØUTINE 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 SUBRØUTINE 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
NØTE	The numbers in the Common Location column refer to positions in the ØCCUR 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 inital 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_α, in degrees
THEZET	224	READIT	Thrust misalignment angle in degrees

NØTE Unless otherwise indicated, the entries in the Common Location Column refer to positions in the PCCUR array.

Name	Common Location	Source of Input	Description
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
V0GPLT, 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
*** NØTE*** 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 coefficent 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 ØCCUR 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_δ PSIZET
CTHZET	215	Cosine of the thrust offset angle θ_δ 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
GAMF0	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 lbm/ft^2-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
QDØT, 32	2676-2707	Distribution along the body of cold wall aerodynamic heating rates
Q0	110	Initial value of the angular rate Q in radians/sec.
RBDØT	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
RNDØT	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
SINGO	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_{α} , in radians
WDØT	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

NØTE Unless otherwise designated, numbers in Common Location column refer to positions in the PCCUR array.

Name	Common Location	Description
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	TTMCØM	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

NØTE 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
NØTE The following quantities are integers (cont'd)		
NDDC0Y	INFTST	Control code used in INTGRL
NGEØM	NØCCUR(15)	Input geometry option code
NLØW	--	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 SUBRØUTINE 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.

rvt Values	Differences	Characteristic Name
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 SUBRØUTINE SAVEDV to save the values of ØCCUR(133-145), ØCCUR(205-209) and ØCCUR(222) in DVAR(1-19) respectively. These values will be restored to the ØCCUR 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 ≠ 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 NLØW and NHIGH are set and statement 404 is executed next. At 404, NCØN is defined. This is the location in the ØCCUR array of the Ith parameter being perturbed. Next, the call to SAVEDV restores the values saved in DVAR to the ØCCUR array. Then going to 499, NLØW and NHIGH are set for the rvt and 499 follows.

At 499, the DØ loop ending at 444 is entered to obtain and process trajectory data. If KK equals two, ØCCUR(NCØN) is set equal to DVL(II) for the first comparison decoy. If KK equals three, ØCCUR(NCØN) is set equal to DVH(II) for the second comparison decoy. Next, if MØDE 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 coefficient quantities to zero. Then LØPT is tested. If LØPT equals zero go to 201. If LØPT is not equal to zero, the next nine statements set rotational quantities equal to zero. If LØPT = 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	Descrip
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)	Altitude
IØP, 90	ZREADX	IØCCUR (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	IØCCUR (313)	The nun altitudes
LPLØT	VIXEN	IØCCUR (302)	The nun altitudes
LX	F123		Flag sp array a integral
NCP	SR2490 or ZREADX	IØCCUR (304)	Number
NDECØY	F123	INFTST	Test pa
ØCCUR, 4000	Unlabeled Common	ØCCUR (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. <u>Input</u> (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		Standard
T, 200	SR2490 or ZREADX	PCCUR (5561 - 5760)	See Tex
TPLØT, 160	VIXEN	PCCUR (1-160)	Array o of altitu
Z, 160	VIXEN	PCCUR (161-320)	Array o data

****
NOTE

DC is a special input of alphanumeric data

A

ock	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	

B

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	
YY, 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, . . . , 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, IØP(KA + 2*KB) is tested. If IØP(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 DØ 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 DØ 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 DØ 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). IGØ is set = NX + NY and the computed GØ TØ is executed, sending control to the statement having the statement number = 100*IGØ. 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), RQI.

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, IØP(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, . . . , 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 RLØ is set = $[R(1)/SIG(1)]^2$ and SUM is set = 0. The values of R at the altitudes H(2), H(3), . . . 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

Name	Source	Description
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

Name	Description
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 INFCØF (DC, IZ, ZQ)

1. Purpose

SUBROUTINE INFCØF 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	IØCCUR(308)*	F123	second subscript of H
IZ, 19	-	INTGRL	locations, in ØCCUR array, of 19 special parameters
JJ	IØCCUR(310) *	F123	first subscript of SAVE
LPLØT	IØCCUR(302) *	VIXEN	number of output altitudes
NCØN	IØCCUR(312) *	F123	location in the ØCCUR array of design parameter being varied
NDECØY	IØCCUR(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 INFCØF 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 decy.

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 HØLD 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 11 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 NCØN 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. <u>Other Information</u>

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

Name	Source	Description
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

Name	Description
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, ..., (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

Name	Source	Description
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

Name	Symbol	Description
A, ND	A_i	coefficient of X in linear fit
B, ND	B_i	constant in linear fit

4. Numerical Procedure

$A_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$ and $B_i = y_i - A_i \, x_i$ 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, SLØPE, 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

Name	Description
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
SLØPE	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, ØCCUR)

1. Purpose

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

2. Input

*indicates integer quantity

Name	Source of Input	Description
DVAR, 50	F123	array to save selected ØCCUR values
K *	F123	test parameter
ØCCUR, 4000	F123	common array

3. Output

Name	Description
DVAR, 50	See Input
ØCCUR, 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 ØCCUR(133-145), ØCCUR(205-209) and ØCCUR(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 CHARACTERISTICS" 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

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

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

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

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

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

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

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

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 SUBROUTINE 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

Name	Common Location	Source of Input	Description
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.

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

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

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

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

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. <u>Other Information</u>

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
HØLD, 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

ock	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 HØLD 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 ØCCUR 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 NCØN. 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 locp 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 IØP(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, IØP(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 if 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

Name	Source	Description
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

Name	Description
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, ... 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, ... 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)

PROGRAMER	MACHINE	DATE	BCD NAME
M. DeBolt	IBM 360/75	19 June 1967	ACØS, ACØSR, ACØSD

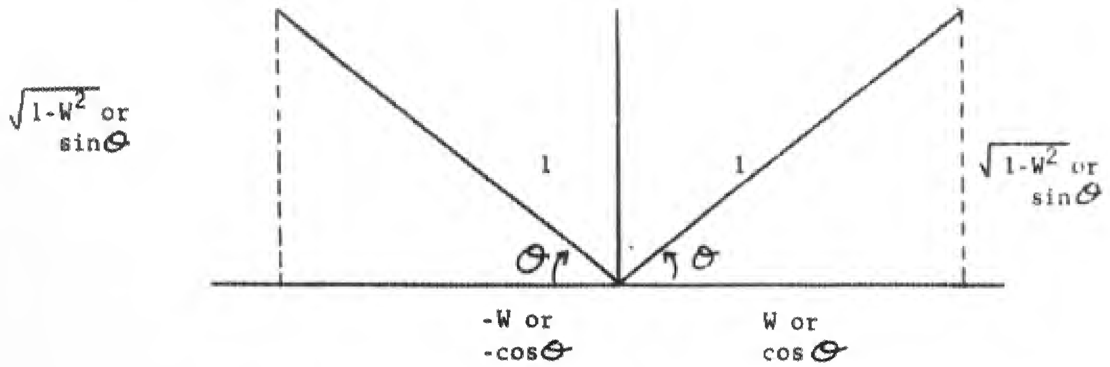
IDENTIFICATION: Arc Cosine Function

USAGE:

X = ACØS(W) X returned in radians
X = ACØSR(W) X returned in radians
X = ACØSD(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 ØPT = 2.

RAD 1-1101 10-63

8.2 Arc Sine Function - ASIN (ASINR, ASIND)

PROGRAMER	MACHINE	DATE	BCD NAME
M. DeBolt	IBM 360/75	26 June 1967	ASIN(W),ASINR(W),ASIND(W)

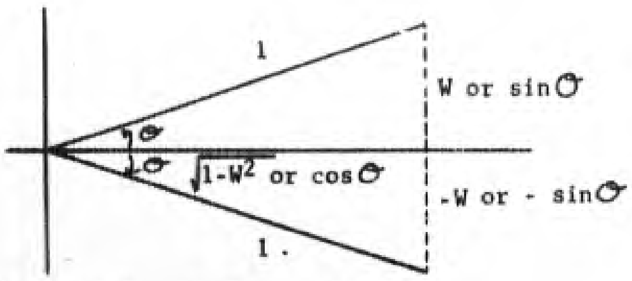
IDENTIFICATION: Arc Sine Function

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 ØPT = 2.

8.3 Arc Tangent Function - ATANQ (ATANQR, ATANQD)

	Avco Corporation — research and advanced development division **PROGRAMER'S HANDBOOK**		WRITE-UP NUMBER [illegible] PAGE 1 of [illegible]
PROGRAMER M. DeBolt	MACHINE IBM 360/75	DATE 26 June 1967	BCD NAME ATANQD(Y,X) ATANQ(Y,X), ATANQR([illegible]

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 (IHC265I).

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 ØPT = 2.

RAD 1-1101
10-63

8.4 FIXIBCØM (MIBCØM) has been dummied out of the program. Originally this routine was a modification of the IBM system routine IBCØM.

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)

Language: ØS/360 Fortran IV and BAL

IDENTIFICATION:

Title: Variable Field Input Subroutines

PURPOSE:

To perform the same input operations on OS/360 as BCDCØN/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 (8HNAME    , L, XY, N1, N2,....)

CALL READIN (INCØI1)

CALL KEDING (NTAPE)
```

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 BCDCØN 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 BCDCØN. It is

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

RAD 1-1101 10-63

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 LØGICAL*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 <u>must</u> 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 BCDCØN and HØLNAM, <u>does</u> distinguish between single-valued variables and one-dimensional arrays. This can be seen by the following:

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

          CALL SETUP (8HY      , 8, Y)
```

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 subroutines) 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 //GØ.FT05F001 DD *), lists the cards on the output tape (FT06F001), 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 <u>not</u> terminate a case. If the routine encounters an E (part of END-ØF-JØB) 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 <u>in alphanumeric fields</u>, if EBCDIC characters (such as parentheses, equals, apostrophe) are desired, the 029 keypunch should be used.

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/SYMBLS, 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 <u>sequentially</u> (note this is different from 7094/SYMBLS). 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/SYMBLS). However, storing on any one of the subscripts may be forced by putting the subscript number <u>immediately</u> after the right hand parenthesis of the subscript field.

Examples: DIMENSIØN X(4,6,3)

i) X 5.0 6.0 8.0
   X(1) 5.0 6.0 8.0
   X(1, 1, 1)1 5.0 6.0 8.0
   } are all equivalent to

   X(1, 1, 1) = 5.0

   X(2, 1, 1) = 6.0

Language: ØS/360 Fortran IV and BAL

X(3, 1, 1) = 8.0

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

X(1, 1, 1) = 5.0

X(1, 2, 1) = 6.0

X(1, 3, 1) = 8.0

iii) X(1, 1, 1) 5.0 6.0 8.0
X(1, 1, 1)3 5.0 6.0 8.0
or even X(1)3 5.0 6.0 8.0 } are equivalent to

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/SYMBLS one would have to write X(27,2) since 27 = 4 x 6 x 1 + 3 and must be computed from the dimensions. Now one only needs to write X(3, 2, 2)2

As in 7094/SYMBLS 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) 5.0 is equivalent to

X(3, 2, 2) = 5.0

X(3, 3, 2) = 5.0

X(3, 4, 2) = 5.0

Any <u>one</u> 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.

Language: ØS/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 ($2^{31} - 1$) 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

      XXEYY, 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.XXXXXXXXXXXXXXXEYY

   ± X.XXXXXXXXXXXXXXXE±YY

   ± X.XXXXXXXXXXXXXXX±YY

   ± 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.

c) Hexadecimal numbers are input in a manner analogous to the way octal numbers are input by way of 7094/SYMBLS. A Z (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:

i) 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/SYMBLS) 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, <u>starting in column one</u> 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.

RAD 1-1101 10-63

iii) 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

(LØGIC1 = .TRUE.)

```
LØGIC1    .T
LØGIC1    .T.
LØGIC1    .TRUE.
LØGIC1    .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/SYMBLS. 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:

$$X_1(X_2)X_3(X_4)X_5 \text{ , etc.}$$

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 <u>should</u> be no imbedded blanks in the expression, but they are permitted 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 INCØL1.

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/SYMBLS. All data cards are automatically printed. There is also no CØMMØN 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/SYMBLS.

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.

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#

   FDXPI#

   EXIT

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 be greater than or equal to seven times the num of calls to SETUP

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

In 7094/SYMBLS a dollar sign anywhere on a data card meant 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 CØMMØN

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

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-ØF-JØB 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 INCØL1 (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 INCØL1/8888/
CALL SETUP (etc.)
      etc
```

```
1    CALL READIN (INCØL1, $99) [or&99 if EBCDIC]
     [Calculations]
     GØ TØ 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.

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   PLØT . 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 $11BB_{16}$ bytes or 1135 single precision words.

8.7 The dummy subroutines EZPLØT, ENDJØB, 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

```
CMAIN
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
C
C
C       PURPOSE
C
C          MAIN DIRECTS THE CALCULATION OF THE OPTIMUM VALUES FOR THE
C       DESIGN PARAMETERS OF A DECOY.  THE TECHNIQUE USED IS TO
C       MINIMIZE THE DIFFERENCES BETWEEN CERTAIN CHARACTERISTICS OF A
C       REENTRY VEHICLE AND THE  SAME CHARACTERISTICS FOR A DECOY.
C       THE FOLLOWING NINE CHARACTERISTICS MAY BE USED - VELOCITY,
C       DECELERATION, BALLISTIC COEFFICIENT, WAKE LENGTH AT EACH OF
C       THREE RADAR FREQUENCIES AND WAKE CROSS SECTION AT EACH OF THESE
C       FREQUENCIES.   FOUR METHODS FOR FINDING THESE MINIMA ARE
C       (1)  A FIBONACCI ONE VARIABLE SEARCH  (2)  A FIBONACCI TWO
C       VARIABLE SEARCH  (3)  THE DAVIDON METHOD  (4)  THE ROSENBROCK
C       ROTATING COORDINATE METHOD)      A QUANTITY PD, THE PROBABILI-
C       TY THAT A DECOY WILL BE DISCRIMINATED AS BEING A DECOY, CAN
C       ALSO BE DETERMINED.
C
C
C
C
      REAL*4 TAB99(2800)
      COMMON /XXSAVE/ X1SAVE(40), X2SAVE(40), KOUNT
      COMMON/IOPT/IPROC,IN,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON/FOPT/ERR,PRAND, FAC, DELTA
      COMMON/MIN/ALOW(20),UP(20),AMULT(20),CALOW(20),CTP(20)
      COMMON OCCUR, NOCCUR
      COMMON/CCRN/D
      COMMON /IDNOS/ ID1(50), ID2(50)
      COMMON/OWL/OVECT(20),WRF,LRED
      COMMON/BLKJ/ HH(1600),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FD,E, P,TD,RS,SL, Z,Q,A,EL,
     2DELZZ,FAK,C(40,10)
      COMMON/SENSE/ISEN1,ISEN2
      COMMON /NALTFG/FGSM,NALT
      COMMON /CICCUR/ IOCCUR(320)
      COMMON /END/ ITERM
      COMMON/MINSK/N,K, ALB(20), UB(20),XX(20),F1
      COMMON/DOPT/DELX(20)
C
       DIMENSION  IOP(90)
       DIMENSION CUP(20)
       DIMENSION OCCUR(4000), NOCCUR(30)
C
       EQUIVALENCE (NOCCUR(14),NPRINT)
       EQUIVALENCE(OCCUR(4000),LP)
       EQUIVALENCE (IOCCUR(00001), IOP   )
       EQUIVALENCE (IOCCUR(00309), IMPLOT)
       EQUIVALENCE (IOCCUR(00301), IREF  )
       EQUIVALENCE (IOCCUR(00303), MODE  )
C
C
C ITERM IS DEFINED IN FEV
C IF ITERM = -1, FUNCTION = ZERO
C IF ITERM = 0, FUNCTION IS NOT ZERO
```

```
C IF ITERM = 1, FUNCTION IS UNDEFINED.
C
      CALL WHERE(TA899)
      OCCUR(42)= 3.141592653589793
      CALL SR2490
      CALL ZPRM
      MODE = 3
      P = 0.D0
      D = 0.D0
      DELTA=1.0D0
      FAC=1.0D0
      K=0
      ILL111 = 1
      IMPLOT=0
      LRED = 0
      DO 707 I = 1, 50
      ID1(I) = 0
  707 ID2(I) = 0
C
C
    2 CALL READIT(ILL111,IMPLOT)
      ILL111=2
      IF(MODE-2) 100,110,120
C
  100 IREFS = IREF
      IREF = 1
      IOP(74) = 0
      CALL F123(VAL)
      IREF = IREFS
      GO TO 2
  110 CALL F123(VAL)
      GO TO 2
C
C
  120 IF(IREF.EQ.2) GO TO 5
      CALL F123(VAL)
      GO TO 2
C
    5 IRED = -1
C     SAVE QUANTITIES WHICH CAN BE REDEFINED IN REDUCE
      CTPSAV = CTP(1)
      IY = ID2(1)
      YSAV = OCCUR(IY)
C
C
C            SETUP  INITIAL  STATE
      DO 10I=1,IN
      IZ= ID1(I)
      OCCUR(IZ)=OVECT(I)
      X(I)=OVECT(I)
   10 XX(I)=OVECT(I)
C
   20  GO TO (500,600,700,800,850),IPROC
C
C     SETUP FOR FIBONACCI SEARCH
  500 CONTINUE
      IZMIN= ID1(1)
      CALL MIMAX(ALOW(1), UP(1), 1, -1,ERR, LIMIT, XMIN,YMIN)
      IF(ITERM.EQ.1) GO TO 200
      X(1) = XMIN
```

```
      NPSAVE = NPRINT
      NPRINT = 1
      CALL FEV(IN,X,VAL)
      NPRINT = NPSAVE
      GO TO 900
C
C
  600 CONTINUE
      CALL FEV(IN,XX,VAL)
      GO TO 900
C
C                   SETUP  FOR CAVIDON
  700 CONTINUE
      CALL DAVDON
      IF(ITERM.EQ.1) GO TO 200
      GO TO 900
  800 CONTINUE
C
C SET UP FOR ROSBRK
      CALL ROSBRK
      IF(ITERM.EQ.1) GO TO 2C0
      NPSAVE = NPRINT
      NPRINT = 1
      CALL FEV(IN,X,VAL)
      NPRINT = NPSAVE
      GO TO 900
C
C
 850  DO 860 I = 1,IN
      ALB(I) = ALOW(I)
      UB(I) = UP(I)
 860  CONTINUE
      KOUNT = 0
      CALL  GIMAX(ALB(2),UB(2),2,-1,ERR,LIMIT,XX(2),VAL)
      IF(ITERM.EQ.1) GO TO 2C0
      DO 666 I = 1, 40
      IF(XX(2).NE.X2SAVE(I)) GO TO 666
      XX(1) = X1SAVE(I)
      GO TO 870
666   CONTINUE
      WRITE(6,867)
867   FORMAT(1H0 50HERROR IN MAIN.  XX(2) UNEQUAL X2SAVE(I) FOR ANY I )
870   NPSAVE = NPRINT
      NPRINT = 1
      CALL FEV(IN,XX,VAL)
      NPRINT = NPSAVE
      GO TO 900
C
C
C PREPARE TO TIGHTEN CONSTRAINTS.
 900  CALL      REDUCE(CTP,KRED,IRED,LRED,WRF)
      WRITE(6,1100)KRED, ITERM
 1100 FORMAT(1H0 'KRED = 'I6, ' ITERM = ' I6)
C
C
       IF(KRED.EQ.0) GO TO 20
C
C     RESTORE SAVED QUANTITIES
 200  CTP(1) = CTPSAV
      OCCUR(IY) = YSAV
```

```
C                                                                       MAIN 180
      GO TO 2                                                           MAIN 181
C                                                                       MAIN 182
      END                                                               MAIN 183
```

```
CADD
      IMPLICIT REAL*8(A-H,O-Z)
      FUNCTION ADD(L,R,S,A,B,C,D,E)
      DIMENSION E(9)
C
C
C         FUNCTION ADD EVALUATES THE INTEGRAL OF F(T) DT FROM R TO S
C        OVER A TRAPEZOIDAL REGION IN THE T,Y PLANE DEFINED AS THE AREA
C        ENCLOSED BY THE FOUR LINES - T=R, T=S, Y=A*T&B, Y=C*T&D.
C        F(T)= E(L)*T**2&E(L&1)*T&E(L&2)
C
C
      BD = D-B
      CA = C-A
      C1 = BD*E(L&2)
      C2 = (CA*E(L&2)&BD*E(L&1))/2.D0
      C3 = (CA*E(L&1)&BD*E(L))/3.D0
      C4 = CA*E(L)/4.D0
      ADD = C1*(S-R)&C2*(S*S-R*R)&C3*(S**3-R**3)&C4*(S**4-R**4)
      RETURN
      END
```

```
CADM4RK
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE ADM4RK(NZ,ZDEL, VALUE, DERN,UPBND,DNBND,FACTOR,FREQ,
     1 HLIMIT,LZ,ZXINDE,DELMIT)
      DIMENSION   ZBAR(70)
     2 DIF(70),  VALUE(70) , DERNN(70) ,DERN(70),
     3DERNM1(70) ,DERNM2(70), DERNM3(70),
     4VALUEN(70)
      DIMENSION  B0(70) , B1(70)
      DIMENSION    UPBND(70) , DNBND(70)
      DIMENSION  SAVD (70)  ,  SAVE (70)
C
C
C         SUBROUTINE ADM4RK PERFORMS THE INTEGRATION OF NZ FIRST ORDER
C         DIFFERENTIAL EQUATIONS OF THE FORM DY(I)/DX EQUALS THE ITH
C         FUNCTION OF (X,Y(1),...,Y(NZ)) FOR VALUES OF I FROM 1 TO NZ BY
C         A FOUR POINT PREDICTOR CORRECTOR METHOD WHICH WILL ALTER THE
C         INTEGRATION INTERVAL TO MAINTAIN A REQUIRED ACCURACY.
C
C
      N=NZ
      L=LZ
      DEL=ZDEL
      XINDEP=ZXINDE
      IF ( DELMIT ) 9002, 9001, 9002
 9001 DELMIT   =   DEL / 1000.0D0
 9002 CONTINUE
C
      IF (L)  1210, 201, 1211
 1210 L= IABS(L)
      GO  TO  23
 1211 GO TO (23,204,204,204,24),L
C
 204  DO 205  JJ = 1,N
      DERN (JJ)  =  SAVD (JJ)
      VALUE (JJ)  =  SAVE (JJ)
 205  CONTINUE
      XINDEP  =  SINDEP
      GO  TO  24
C
 23   XFREQ  =  XINDEP  &  FREQ
      LL = 2
      SAVDEL  =  DEL
      SINDEP  =  XINDEP
      ASSIGN  41  TO  M
      CALL   DEREQ  ( VALUE  ,  XINDEP  ,  DERN  ,  L  )
      IF(L.EQ.6)GO TO 220
      GO  TO  40
C
 24   GO  TO  M, ( 41 , 42 )
 41   DO  62  KK = 1,N
      DERNM3 (KK) = DERN (KK)
 62   CONTINUE
C
      DO  43  NN = 1,3
      W1  =  DEL / 2.0D0
      DO  44   I = 1,N
      B0(I)  =   0.0D0
 44   CONTINUE
```

```
C
      DO  50    J  = 1,4
      GO  TO  ( 45 , 46 , 48 , 49 ) , J
   45 DIV1  =  6.000
   47 DIV2  =  2.000
      H     =  W1
      GO  TO  60
   46 DIV1  =  3.000
      GO  TO  47
   48 DIV1  =  3.000
      DIV2  =  1.000
      H     =  DEL
      GO  TO  60
   49 DIV1  =  6.000
C
   60 DO  51    I  =  1, N
      B1(I)  =  DERN(I)  *  DEL
      B0(I)  =  B0(I)  &  B1(I) / DIV1
      GO   TO  ( 52  , 52  , 52  , 53 ) , J
   52 VALUEN(I)  =  VALUE(I) &  B1(I) / DIV2
      TINDEP     =  XINDEP    &  H
      GO  TO  51
   53 VALUEN(I)  =  VALUE(I)  &   B0(I)
   51 CONTINUE
   54 CALL  DEREQ  ( VALUEN  , TINDEP  , DERN , L  )
      IF(L.EQ.6)GO TO 220
   50 CONTINUE
C
      DO  61   II  =  1,N
      GO  TO  ( 63, 64, 65 )  ,  NN
   63 DERNM2(II) =  DERN(II)
      GO  TO  65
   64 DERNM1(II)  =  DERN(II)
   65 VALUE (II) =  VALUEN(II)
   61 CONTINUE
      XINDEP  =  TINDEP
C
   43 CONTINUE
      H1  =  DEL
      H2  =  DEL
      H3  =  DEL
      ASSIGN  42  TO  M
      GO  TO  9042
C
   42 CONTINUE
      XFREQ  =  XFREQ  &  FREQ
 9042 CONTINUE
C
      GO  TO  ( 1406, 1407, 1407, 1407, 1207 ) , L
 1407 IF (ABS( XINDEP - XFREQ ) - ABS( DEL ) ) 1403, 1408, 14060
14060 IF(ABS(XINDEP-HLIMIT)-ABS(DEL)) 1408,1408,1406
 1408 CONTINUE
      DO 1410 IJK  =  1,N
      ZBAR(IJK) = VALUE(IJK)
      DERNN(IJK)   =  DERN(IJK)
      DERN    (IJK)  =  DERNM1 (IJK)
      DERNM1  (IJK)  =  DERNM2 (IJK)
      DERNM2  (IJK)  =  DERNM3 (IJK)
 1410 CONTINUE
C
```

```
      LL  =  1
 11   TEMP = ABS( XINDEP - XFREQ )
      IF ( TEMP - ABS( DEL ) ) 16, 16, 15
 16   L  = 2
      SINDEP  =  XINDEP
      SAVDEL  =  DEL
      TEMP1 = ABS( DEL ) / DEL
      DEL  =  TEMP * TEMP1
 15   TEMP = ABS( XINDEP - HLIMIT )
      IF ( TEMP - ABS( DEL ) ) 118, 18, 17
 118  L  =  1
 18   L  =  L & 2
      SAVDEL  =  DEL
      TEMP1 = ABS( DEL ) / DEL
      DEL  =  TEMP * TEMP1
 17   GO TO ( 22 , 6 , 6 , 6 , 22)          , L
C
 1406 CONTINUE
      LL  =  1
      L   =  1
C
 1207 CONTINUE
 12   W1 =  H1 & H2
      W2 =  W1 * W1
      W3 =  W1 & H1
      W4 =  W1 & H3
      W5 =  W4 & W1
      W6 =  W4 & W3
      W7 =  W3 & H3
      W8 =  W1 * H1
      W9 =  W1 * H3
      W10=  H2 & H3
      W11=  H2 * H3
      W12=  H1 * H2
      W16=  W2 & W9
C
      GO  TO  ( 14, 202, 202, 202, 14 ) , L
 14   TEMP  =  XINDEP & DEL
      DEL  =  TEMP  -  XINDEP
      IF ( ABS( DEL ) - ABS( DELMIT ) ) 201, 202, 202
 201  CONTINUE
      L  =  6
      GO TO 4
C
 202  XINDEP  =  XINDEP  &  DEL
      W13=  DEL / 2.0D0
      W14=  W13 * W13 * DEL
      W15=  DEL * DEL / 3.0D0
      W17=  H1 & DEL
      W18=  W1 & DEL
      W19=  W1 * DEL
      W29  =  W3 * DEL
      B0(1)= DEL / W4 * ( W14 & W15 * W6 & W13 * (W2 & 2.0D0 * H1 * W4 &
     X W
     111) & H1* W16 ) / W8
      B1(1)= -DEL / W10* ( W14 & W15 * W5 & W13 * W16) /W12
      B2     =  DEL / W1 * ( W14 & W15 * W7 & W13 * H1 * W4)
     1/W11
      B3     = -DEL / W4 * ( W14 & W15 * W3 & W13 * W8) /W10
     1/H3
```

```
C
      DO 1  I = 1,N
      ZBAR(I) =VALUE(I)&B0(I)*DERN(I)&B1(I)*DERNM1(I)&B2 * DERNM2(I)&B3
     1 *DERNM3(I)
    1 CONTINUE
      CALL  DEREQ  ( ZBAR     , XINDEP , DERNN     , L )
      IF(L.EQ.6)GO TO 220
      A     = (W14 & W15 * W3 & W13 * W8) / W17 / W18
      AO    =-(W14 & W15 * (W3- DEL) & W13*(W8 - W29) -DEL *W8)
     1/W8
      A1   =DEL/W17*(W14& W15 *(W1 -DEL)- W13 * W19 )/W12
      A2    = DEL /W1 * (W14/3.000  & H1 * W15 /2.000) /H2 / W 18
      DO 13  I= 1,N
      VALUEN(I)=VALUE(I)&A    *DERNN(I)& AO    *DERN(I)&A1    *
     1DERNM1(I)&A2    *DERNM2(I)
      DIF(I) = ABS( VALUEN(I) - ZBAR(I) )
C
      IF  (  DIF(I)  -  UPBND(I) )  13 , 13 , 2
    2 XINDEP  =  XINDEP  - DEL
      DEL     =  DEL  -  FACTOR * DEL
      IF(L-5)7000,14,220
 7000 CONTINUE
      L   =  1
      LL  =  1
      GO TO  14
C
   13 CONTINUE
      IF(L-2)500,5,500
  500 CONTINUE
      H3  =  H2
      H2  =  H1
      H1  =  DEL
      GO TO  ( 11, 5, 5, 5, 22 ) , L
  200 SAVDEL  =  DEL
      GO TO 21
   22 DO 19   K  = 1 ,N
      IF  (  DIF(K)  -  DNBND(K) )  19 , 19 , 7010
   19 CONTINUE
      DEL  =  DEL & DEL * FACTOR
 7010  IF(L-5)6,200,220
C
 5    CONTINUE
      DO  1040  LLL  = 1,N
      VALUE (LLL) = ZBAR(LLL)
      DERN(LLL)  = DERNN(LLL)
 1040 CONTINUE
   21 LL  =  2
C
      GO TO (6,40,40,40,6),L
 6    DO 8  J  =  1,N
      DERNM3(J)  =  DERNM2(J)
      DERNM2(J)  =   DERNM1(J)
      DERNM1(J)  =  DERN   (J)
      DERN(J)   =  DERNN(J)
      VALUE(J)=ZBAR(J)
 8    CONTINUE
   40 CONTINUE
C
      GO TO (212,208,208,208,209),L
  212 CONTINUE
```

```
      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
  203 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)
      SUBROUTINE AERODY
C
      REAL*8 K1L,K2L,K1T,K2T,K3T,K4T,LA,LA1,LA2,LAMDA,LAPS,MB,MINF
C
      DIMENSION A(514),XLA(8),QDOT(4,8),PEPSB(8),OCCUR(4000),NUCCUR(30)
C
      EQUIVALENCE
     1(OCCUR(010),CAPL  ),(OCCUR(029),HSRT) ),(OCCUR(032),LAMDA ),
     2(OCCUR(033),LA    ),(OCCUR(035),MINF  ),(OCCUR(046),PINFPS),
     3(OCCUR(047),PS    ),(OCCUR(052),RN    ),(OCCUR(055),RHOINF),
     4(OCCUR(064),SINT  ),(OCCUR(069),THETAD),(OCCUR(070),TANT  ),
     5(OCCUR(076),THETA ),(OCCUR(082),V     ),(OCCUR(091),Z     ),
     6(OCCUR(092),ZTR   ),(OCCUR(138),LA1   ),(OCCUR(144),LA2   ),
     7(OCCUR(145),ZTURN ),(OCCUR(189),FACTR1),
     8(OCCUR(0301),     A(1)),(OCCUR(0815),    XLA(1)),
     9(OCCUR(2676),QDOT(1,1)),(OCCUR(2801),PEPSB(1))
C
      COMMON OCCUR,NUCCUR
C
C
C
C          SUBROUTINE AERODY CALCULATES HEATING RATES FOR APPROPRIATE
C       BODY STATIONS.   FOR A SHARP CONE, THESE STATIONS ARE THE
C       STAGNATION POINT AND THE MAXIMUM DIAMETER POINT.   FOR THE
C       BLUNT CONE, THE STAGNATION POINT, SONIC POINT IN TURBULENT
C       FLOW ONLY, TANGENT POINT, THE 20,40,60,75,90 PER CENT STATIONS
C       BASED ON THE INITIAL AXIAL LENGTH, AND THE MAXIMUM DIAMETER
C       POINT - IN ADDITION, SHARP CONE VALUES ARE CALCULATED AT THE
C       MAX. DIAMETER POINT FOR COMPARISON PURPOSES.
C          QDOT(1,1)=STAG. POINT
C          QDOT(2,J)=LAMINAR
C          QDOT(3,J)=TURBULENT
C          QDOT(4,1)=SONIC  POINT
C          J=1  TANGENT POINT  ON BLUNT CONE
C          J=2 THROUGH 7  CORRESPOND TO THE 20,40,60,75,90 PER CENT
C       STATIONS AND THE MAXIMUM DIAMETER POINT ON THE BLUNT CONE.
C          J=8  MAXIMUM DIAMETER POINT ON THE SHARP CONE OF ANGLE THETA.
C
C
C          CALCULATION OF STAGNATION POINT HEATING RATE
      IF(LAMDA.LT.1.0D-3)RS=1.0D0
      IF(LAMDA.GE.1.0D-3)RS=RN
      QDOTST=1.76D4*SQRT(RHOINF/(.002375D0*RS))*(V/2.604)**3.15D0
      QDOT(1,1)=QDOTST
C
C          SHARP CONE PE/PS AT MAX DIAMETER
      DO 20 J=1,7
   20 PEPSB(J)=0.D0
      XLA(8)=1.0D0
      PEPSB(8)=0.0331D0*EXP(0.0064D0*THETAD-0.33D0*(MINF-5.D0)**0.85D0)
      PEPSB(8)=PEPSB(8)&0.468D-3*THETAD**1.88032D0
C
C          BLUNT CONE STATION LOCATIONS
      IF(LAMDA.LT.1.0D-3)GO TO 1
      XLA(1)=RN*(1.D0-SINT)/LA
      IF(Z.GT.ZTURN)DLA=LA1/LA
      IF(Z.LE.ZTURN)DLA=LA2/LA
```

```
      XLA(2)=1.0D0-0.8D0*DLA
      XLA(3)=1.0D0-0.6D0*DLA
      XLA(4)=1.0D0-0.4D0*DLA
      XLA(5)=1.0D0-0.25D0*DLA
      XLA(6)=1.0D0-0.1D0*DLA
      XLA(7)=1.0D0
C
C           QUANTITY  TEST  IS A NON-DIMENSIONAL AXIAL DISTANCE USED IN
C         DETERMINING WHEN FLOW IS CONSIDERED CONICAL
      TEST=1.13D0*RN/(LA*TANT*TANT)
      DO 2 I=1,7
      IF(I.EQ.1)GO TO 16
      IF(XLA(I).LE.XLA(1))GO TO 21
   16 CONTINUE
      IF(XLA(I).GE.TEST)GO TO 3
C
      TEM=LA*TANT*TANT*1.0D4/RN
      XP=0.1740D0*THETAD
      YP=10.D0/MINF
      ZP=0.2D0*DLOG(XLA(I)*TEM)
      WP=0.D0
      IF(THETAD.LT.20.D0)GO TO 13
C           CALCULATION OF BLUNT CONE PRESSURE DISTRIBUTION FOR CONE HALF
C         ANGLE GREATER THAN OR EQUAL TO 20 DEGREES.
      DO 4 N=1,2
      NN=N-1
      ISUB1=41&NN
      TEMN=XP**NN
      DO 4 J=1,3
      JJ=J-1
      ISUB2=ISUB1&2*JJ
      TEMJ=YP**JJ
      DO 4 K=1,3
      KK=K-1
      ISUB=ISUB2&6*KK
    4 WP=WP&A(ISUB)*TEMN*TEMJ*ZP**KK
      GO TO 15
C           CALCULATION OF BLUNT CONE PRESSURE DISTRIBUTION FOR CONE HALF
C         ANGLE LESS THAN 20 DEGREES.
   13 DO 14 N=1,3
      NN=N-1
      ISUB1=135&NN
      TEMN=XP**NN
      DO 14 J=1,3
      JJ=J-1
      ISUB2=ISUB1&3*JJ
      TEMJ=YP**JJ
      DO 14 K=1,3
      KK=K-1
      ISUB=ISUB2&9*KK
   14 WP=WP&A(ISUB)*TEMN*TEMJ*ZP**KK
   15 PEPSB(I)=WP*TANT*TANT&PINFPS
      GO TO 2
   21 XLA(I)=XLA(1)
      PEPSB(I)=0.0D0
      GO TO 2
    3 PEPSB(I)=PEPSB(8)
    2 CONTINUE
C
C
```

```
C         LAMINAR HEATING RATE CALCULATIONS
    1 IF(Z.LT.ZTR)GO TO 5
      QBL=10.D0**(0.51420C*DLOG(HSRT0)**0.9736D0)
      K1L=0.9664D0&THETAD*(0.00528D0&0.288D-3*THETAD)
      K2L=1.CD0&LAMDA*(1.782D0-2.CC8D0*LAMDA)
      LAPS=LA/PS
      IF(LAMDA.GT.1.CD-3) GO TO 6
C
C         SHARP CONF
      QDOT(2,8)=QBL*SQRT(PEPSB(8)/(2116.D0*LAPS*XLA(8)))/(K1L*K2L)
      RETURN
C
C         BLUNT CONF
C         LAMINAR TANGENT POINT HEATING
    6 QDOT(2,1)=1.7320C*QDOT(1,1)*(0.007789D0&PEPSB(1)*(1.849D0&PEPSB(1)
     1*(-1.6832D0&0.841D0*PEPSB(1))))
C         LAMINAR HEATING ON CONICAL FRUSTUM
      DO 7 I=2,8
    7 QDOT(2,I)=QBL*SQRT(PEPSB(I)/(2116.D0*LAPS*XLA(I)))/(K1L*K2L)
      RETURN
C
C
C         TURBULENT HEATING RATE CALCULATIONS
    5 QBTC=10.D0**(0.745D0*DLOG(HSRT0)**0.3122D0)
      K1T=0.9D0&0.02D0*THETAD
      K2T=0.6D0
      K3T=0.65D0&LA*(0.0318D0-0.65D-3*LA)
      K4T=1.0D0
      IF(LAMDA.GT.1.0D-3)GO TO 8
C
C         SHARP CONE
      TM=PS/2116.D0
      QDOT(3,8)=QBTC*(PEPSB(8)*TM)**0.8D0/(K1T*K2T*K3T*K4T*XLA(8)**0.165
     XD0)
      RETURN
C
C         BLUNT CONF
C         TURBULENT SONIC POINT HEATING
    8 IF(Z.LE.115000.D0)MB=3.45D0
      IF(Z.GT.115000.D0)MB=2.254D0&Z*(2.246D-5&Z*(-146.9D-12&Z*367.1D-18
     X))
      QDOT(4,1)=3760.D0*(RHOINF**0.8D0)*V**MB
      QDOT(4,1)=QDOT(4,1)/((RS**0.2D0)*(FACTR1          )*(10.D0**(4.00*
     XMB)))
      TM=PS/2116.D0
      QDOT(3,8)=QBTC*(PEPSB(8)*TM)**0.8D0/(K1T*K2T*K3T*K4T*XLA(8)**0.165
     XD0)
C         TURBULENT TANGENT POINT HEATING
      FITANG=(PEPSB(1)**(4.1D0/6.D0))*(1.D0-(PEPSB(1))**(1.D0/6.D0))**0.
     X4D0
      QDOT(3,1)=1.346D4/((RS*(1.D0-THETA))**0.2D0)
      QDOT(3,1)=QDOT(3,1)*((RHOINF/2.375D-3)**0.8D0)*FITANG*(0.0001D0*V)
     X**MB
C         ASSIGNMENT OF APPROPRIATE DENOMINATOR FOR HEATING ON STATIONS
C      AFT OF TANGENT POINT.
      TEST=1.13D0*RN/(3.D0*LA*TANT*TANT)
      K2T=1.0D0
      IF(LAMDA.LE.0.2D0)K2T=0.6D0&2.0D0*LAMDA
      K3T=0.65D0& LA *(0.0318D0-0.69D-3* LA )
      K4T=1.0D0
```

```
      DENOM1=K1T*K2T*K3T*K4T
C     DENOM1 FOR XLA .LE. TEST
      K3T=0.69D0&LA*(0.0318D0-0.69D-3*LA)
      DENOM2=K1T*K2T*K3T
C     DENOM2 FOR XLA .LE. 3.*TEST
      K2T=0.6D0
      K4T=1.0D0
      TEM=LA-RN&RN/SINT
      K3T=0.69D0&TEM*(0.0318D0-0.69D-3*TEM)
      DENOM3=K1T*K2T*K3T*K4T
C     DENOM3 FOR XLA .GT. 3.*TEST
      DO 9 I=2,7
      IF(XLA(I).LE.TEST)GO TO 10
      IF(XLA(I).LE.(3.0D*TEST))GO TO 11
      TM=PS/2116.D0
C          TURBULENT HEATING ON CONICAL FRUSTUM.
      QDOT(3,I)=QBTC*(PEPSB(I)*TM)**0.8D0/(DENOM3          *XLA(I)**0.165
     XD0)
      GO TO 9
   10 DENOM=DENOM1
      GO TO 12
   11 TEM=XLA(I)-TEST
      K4T=0.901D0&TEM*(-0.867D0&0.966D0*TEM)
      DENOM=DENOM2*K4T
      TM=PS/2116.D0
   12 QDOT(3,I)=QBTC*(PEPSB(I)*TM)**0.8D0/(DENOM)
    9 CONTINUE
      RETURN
      END
```

```
CAIM
      SUBROUTINE AIM
         IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FC,E, P,TO,RS,SL, Z,Q,A,EL,
     2DELTA,FAC,C(40,10)
      COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC
      COMMON /END/ ITERM
C
C
C           SUBROUTINE AIM, PART OF THE DAVIDON MINIMIZATION METHOD,
C         ESTIMATES THE LOCATION OF THE RELATIVE MINIMUM WITHIN THE
C         INTERVAL SELECTED BY SUBROUTINE READY.  THIS LOCATION IS
C         COMPARED WITH THE VALUE THAT WOULD BE EXPECTED FROM A 'PERPEN-
C         DICULAR STEP'.
C
C
  300 Z=GS&GSP&3.0*(F-FP)/EL
  301 TO=GS/Z
      TI=GSP/Z
  302 Q = DSQRT(1.0D0-TO*TI)
      Q = DABS(Q*Z)
      A=(GSP&Q-Z)/(GSP-GS&2.0*Q)
  303 TO=(EL*(GSP&Z&2.0*Q)*A**2)/3.0
  304 FO=FP-TO
  305 CALL MATMP (N,N,H,GP,T)
  306 TP1=GSP/SL
  307 DO 308 I=1,N
  308 T(I)=-T(I)&TP1*S(I)
  309 M=1
  310 CALL MATMP (M,N,T,GP,GTP)
  311 IF (2.0*TO&GTP)317,312,312
  312 TP1=1.0-A
  313 DO 314 I=1,N
  314 T(I)=A*X(I)&TP1*XP(I)
  315 L=1
  316 RETURN
C
C
  317 IF (F&GTP/2.0)312,318,318
  318 DO 319 I=1,N
  319 T(I)=T(I)&XP(I)
  320 M1=2
C
  321 CALL FCN(N,GB,FB,T,M1)
      IF(ITERM.EQ.0) GO TO 322
C
      DO 10 I = 1,N
10    X(I) = T(I)
      GO TO 316
  322 IF (FB-FO)323,312,312
  323 WRITE (6,1)
  324 DO 325 I=1,N
  325 S(I)=T(I)-XP(I)
  326 M=1
  327 CALL MATMP (M,N,S,GB,GTT)
  328 GTT=GTT-GTP
  329 IF (GTT)335,330,330
C
```

```
  330 GSS=GTT
  331 SL=-GTP
  332 EL=1.0
  333 L=2
  334 GO TO 316
C
  335 L=3
  336 GO TO 316
    1 FORMAT (9HORICOCHET)
      END
```

```
CARFDT2
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE ARFDT2(H,ASOUND,RHO)
C
C           SUBROUTINE ARFDT2 CALLS IN SUBROUTINE COMP62 AND CONVERTS THE
C        NON-DIMENSIONAL DENSITY OBTAINED FROM IT INTO UNITS OF SLUGS
C        PER CUBIC FOOT.
C
      RHOO = 0.0023769D0
      CALL COMP62(H, RHO,ASOUND,T,P)
      RHO= RHO* RHOO
      RETURN
      END
```

```
CAR2DIM
      IMPLICIT REAL * 8 (A-H, O-Z )
            SUBROUTINE  AR2DIM(N, M, X, XTB, Y, YTB, Z , ZTB  )
C   AR2DIM  NOTE THIS HAS ZTB(J,I) . THE LIBRARY HAS Z(I,J)
      DIMENSION  XTB(N) ,YTB(M) ,  ZTB(M, N)
C
C
C           SUBROUTINE AR2DIM PERFORMS A LINEAR INTERPOLATION IN A TWO-
C        DIMENSIONAL TABLE FOR Z AS A FUNCTION OF X AND Y.
C
C
        NN = N-1
        MM =M-1
         DO 5 I = 1, NN
         I = I
       IF( XTB(I&1) -X ) 5, 6, 6
    6  P2  = (X- XTB(I) ) / (XTB(I&1) - XTB(I) )
          GO TO 7
    5  CONTINUE
C
    7   DO 9 J =1, MM
         J=J
       IF( YTB(J&1) - Y )  9, 10, 10
   10  P1 = (Y- YTB(J)) / ( YTB(J&1) -YTB(J) )
             GO TO 11
    9  CONTINUE
C
   11    Z = (1.D0 -P1 -P2 & P1 * P2) * ZTB(J,I) & ( P2 -P2 * P1) * ZTB(J
     1,I&1)  &( P1 - P1 * P2) * ZTB(J&1,I) &  P1 * P2 * ZTB(J&1,I&1)
C
   13   RETURN
      END
```

```
CAR3DIM
      IMPLICIT REAL * 8( A-H, O-Z )
      FUNCTION AR3DIM( D , N20, M20, N10 ,  ROW, H , EM  ,XYZTBL )
      DIMENSION D(N20, M20, N10 ) , XYZTBL( 11,3)
C
C
C           FUNCTION AR3DIM IS THE RESULT OF A TABLE LOOK-UP EMPLOYING
C         LINEAR INTERPOLATION TO DETERMINE THE VALUE OF A DEPENDENT
C         VARIABLE WHICH IS A FUNCTION OF THREE INDEPENDENT VARIABLES.
C
C
      DO 1  I=2, N10
         I=I
      IF( EM .LE.  XYZTBL( I,3)  )  GO TO 2
    1 CONTINUE
C
      I= N10
      WRITE(6,1000)  XYZTBL(I,3)
 1000   FORMAT(  '    THE VALUE OF M-RAT IS GREATER THAN THE M-RAT VALUE
     1OF THE LARGEST D-TABLE WHICH IS'1P1D18.6  /  )
C
    2 CONTINUE
      I1= I-1
      CALL AR2DIM( N20, M20, ROW, XYZTBL(1,1),  H, XYZTBL(1,2),
     1 ANS1   , D(1,1,I1)  )
      CALL AR2DIM( N20, M20, ROW, XYZTBL(1,1),  H, XYZTBL(1,2),
     1 ANS2   , D(1,1,I)  )
      AR3DIM = (ANS1 *(XYZTBL( I,3) - EM)  & ANS2 * (EM - XYZTBL(I1,3
     1) ) ) / ( XYZTBL(I,3) -XYZTBL(I1,3)  )
C
      RETURN
      END
```

```
CAVPLT  PRØGRAM 2542
CAVPLT WITH CØRE REMØVED  SEPT. 11, 1968
CAVPLT
      SUBRØUTINE AVPLT(XQ,IZ,X,Y,Z,KY,IC)
      IMPLICIT REAL*8(A-H,Ø-Z)
      CØMMØN OCCUR, NØCCUR
      CØMMØN /CPCCUR/ PCCUR(11770)
      CØMMØN /CICCUR/ IØCCUR(320)
      REAL*4 B,B1,B2,B3,   XQ
       DIMENSIØN  XQ(40),  IØP(90)
      DIMENSIØN B1(18),B2(10),B3(10),IC(1),      HØLD(3,50),IZ(1)
      DIMENSIØN NØCCUR(30),NPV(160),ØCCUR(4000),R(3),SAVE(9,160,3),
     1 W(160)
      DIMENSIØN X(1),Y(1),Z(1),ZPLØT(160)
       DIMENSIØN  T(200)
       EQUIVALENCE  (PCCUR(07101), HØLD  )
       EQUIVALENCE  (PCCUR(07251), SAVE  )
       EQUIVALENCE  (PCCUR(05561), T     )
       EQUIVALENCE  (PCCUR(00161), ZPLØT )
      EQUIVALENCE  (IØCCUR(00308), II     )
      EQUIVALENCE  (IØCCUR(00309), IMPLØT)
      EQUIVALENCE  (IØCCUR(00001), IØP    )
      EQUIVALENCE  (IØCCUR(00310), JJ     )
      EQUIVALENCE(IØCCUR(00313), KX      )
      EQUIVALENCE  (IØCCUR(00302), LPLØT)
      EQUIVALENCE  (IØCCUR(00312), NCØN   )
      EQUIVALENCE  (IØCCUR(00305), NDECØY)
      EQUIVALENCE  (IØCCUR(00307), NPA    )
      EQUIVALENCE  (IØCCUR(00141), NPV    )
      DATA B/4H    /
C
C
C         SUBRØUTINE AVPLT CAN PRØDUCE FØUR KINDS ØF PLØTS, DEPENDING
C      ØN THE VALUE ØF KY.  IF KY=1, INFLUENCE CØEFFICIENT PLØTS WILL
C      BE PREPARED AT A SPECIFIED NUMBER ØF ALTITUDES, NPA.  IF KY=2,
C      ØNE CØRRIDØR PLØT IS PRØDUCED.  IF KY=3, ØNE PLØT ØF Y VS. X
C      IS PRØDUCED.  IF KY=4, ØNE PLØT ØF LØG(Y) TØ THE BASE 10 VS.
C      X WILL BE PRØDUCED.  IF KY=5, THE PLØT FILE WILL BE CLØSED AND
C      THE CØMPUTER RUN WILL BE TERMINATED.
C
C
      REWIND 9
      IF(IMPLØT.NE.0) GØ TØ 10
      CALL IDFRMV(12HLØRAYNE K210,2H  ,8H2542F    )
      IMPLØT = 1
C
C
 10   GØ TØ (20,20,110,110,230), KY
 20   CØNTINUE
      WRITE(9,2000) IC(1), IC(2)
      DØ 33 I = 1, 19
      JP = I
      IF(IZ(I).EQ.NCØN) GØ TØ 40
 33   CØNTINUE
      JP = 0
      GØ TØ 45
 40   JS = JP + JP
      JR = JS -1
 45   IF(KY.EQ.2) GØ TØ 70
C
```

```
C
      W(1)= HØLD(1,II)
      W(2)= HØLD(2,II)
      W(3)= HØLD(3,II)
      DØ 66 I=1,NPA
      J = NPV(I)
      R(1) = SAVE(JJ,J,1)
      R(2) = SAVE(JJ,J,2)
      R(3) = SAVE(JJ,J,3)
      WRITE(9,2100)(ØCCUR(L),L=127,129),ZPLØT(J)
      IF(JP.EQ.0) GØ TØ 50
      WRITE(9,2200) XQ(JR), XQ(JS)
      GØ TØ 55
   50 WRITE(9,2300) NCØN
   55  ENDFILE 9
      REWIND 9
      READ(9, 3100) B2
      READ(9, 2900) B1
      READ(9, 3100) B3
      REWIND 9
      IF(IØP(76).EQ.1) GØ TØ 60
C
      XL = DMIN1(W(1),W(2),W(3))
      XU = DMAX1(W(1),W(2),W(3))
      YL = DMIN1(R(1),R(2),R(3))
      YU = DMAX1(R(1),R(2),R(3))
      XU = (XU -XL)/8.D0
      XL = XL -XU
      YU = (YU -YL)/8.D0
      YL = YL -YU
      CALL PLT(1,0,XL,XU,W,0,25,YL,YU,7,0,R,70,B1,40,B3,40,B2)
      CALL PLT(2,3,1)
      CALL PLT(5,12,PLTIM)
      WRITE(6,3000) PLTIM
      GØ TØ 66
C
   60 CALL EZPLØT(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)
C
   66 CØNTINUE
      RETURN
C
C
C
C
   70 XU = ZPLØT(1)
       XL = ZPLØT(LPLØT)
      CALL MAXMIN(KX,KX,X ,XMIN,XMAX)
      CALL MAXMIN(KX,KX,Y ,YMIN,YMAX)
      CALL MAXMIN(KX,KX,Z ,ZMIN,ZMAX)
      YL = DMIN1(XMIN,YMIN,ZMIN)
      YU = DMAX1(XMAX,YMAX,ZMAX)
      WRITE(9,2500)
      IF(NDECØY.GT.1) GØ TØ 80
      WRITE(9,2400)(ØCCUR(L),L=127,129)
      GØ TØ 95
   80 IF(JP.EQ.0) GØ TØ 90
      JT = IZ(JP)
      WRITE(9,2600)(ØCCUR(L),L=127,129),XQ(JR),XQ(JS),ØCCUR(JT)
```

```
      GO TØ 95
   90 WRITE(9,2700)(ØCCUR(L),L=127,129),NCØN,ØCCUR(NCØN)
   95  ENDFILE 9
      REWIND 9
      READ(9, 3100) B2
      READ(9, 3100) B3
      READ(9, 2900) B1
      REWIND 9
      IF(IØP(76).EQ.1) GØ TØ 100
C
        XU = (XU-XL)/8.0D0
      XL = XL -XU
      YU = (YU -YL)/8.D0
      YL = YL -YU
      NF = 123
      CALL PLT(1,0,XL,XU,T,0,NF,YL,YU,4,1,X,YL,YU,4,1,Y,YL,YU,4,1,Z,
     1 70,B1,8,B3,8,B2,8,B3,8,B2,8,B3,8,B2)
      CALL PLT(2,KX,1)
      CALL PLT(5,12,PLTIM)
      WRITE(6,3000) PLTIM
      RETURN
  100 CALL EZPLØT(9.,1,1,T,X,KX,1,2,42,70,B1,40,B3,40,B2,1,-2,XL,XU,-2,
     1YL,YU,1,0,1)
      CALL EZPLOT(9.,1,1,T,Z,KX ,1,2,42,1,B,1,B,1,B,2,-2,XL,XU,-2,YL,YU,
     1 1,0,1)
      CALL EZPLØT(9.,1,1,T     ,Y,KX    ,1,2,42,1,B,1,B,1,B,2,-2,XL,
     1 XU,-2,YL,YU,1,0,1)
      RETURN
C
C
C
  110 N = 1
      IF(IC(1).EQ.0.AND.KY.EQ.4) N = 2
      DØ 111 J = 2, LPLØT
      IF(Y(J-1).NE.Y(J))GØ TØ 120
  111 CØNTINUE
      RETURN
  120 DO 121 I = 1, LPLØT
      W(I) = Y(I)
  121 CØNTINUE
      IF(N.NE.2) GØ TØ 125
      IF(KY.NE.4) GØ TØ 125
      IF(IC(1).NE.0) GØ TØ 125
      DØ 122 I = 1, LPLØT
      W(I) = DLØG10(W(I))
  122 CØNTINUE
  125 CØNTINUE
      WRITE(9,2800)(ØCCUR(L),L=127,129)
       ENDFILE 9
      REWIND 9
      READ(9, 2900) B1
      REWIND 9
      IF(IØP(76).EQ.1) GØ TØ 130
C
      XU = (ZPLØT(1)-ZPLØT(LPLØT))/8.D0
      XL = ZPLØT(LPLØT) -XU
      CALL MAXMIN(LPLØT,LPLØT,W,YMIN,YMAX)
      YU = (YMAX-YMIN)/8.D0
      YL = YMIN -YU
```

```
      CALL PLT(1,0,XL,XU,X,0,25,YL,YU,4,0,W,70,B1,40,B3,40,B2)
      CALL PLT(2,LPLØT,1)
      CALL PLT(5,12,PLTIM)
      WRITE(6,3000) PLTIM
      RETURN
  130 CALL EZPLØT(9.,1,N,X,W,LPLØT,1,2,38,70,91,40,XQ,40,IZ,1,1,XL,XU,1,
     1YL,YU,1,0,1)
       RETURN
C
C
C
  230 IF(IØP(76).EQ.1) GØ TØ 240
      CALL PLT(6)
      CALL EXIT
      RETURN
C
C
C
  240 CALL FRAMEV
       CALL ENDJØB
       CALL PLTND
       RETURN
C
C
C
C
 2000 FØRMAT(1H 5HDELTA 2A4, 27X)
 2100 FØRMAT(1H 5HDATE F9.3, 6H CASE F9.3,6H MEMØ F9.3, 10H ALTITUDE
     1 F9.1, 9X)
 2200 FØRMAT(1H 2A4, 32X)
 2300 FØRMAT(1H I12, 28X)
 2400 FØRMAT(1H 5HDATE F9.3, 6H CASE F9.3, 6H MEMØ F9.3, 28X)
 2500 FØRMAT(1H 8HALTITUDE 32X)
 2600 FØRMAT(1H 5HDATE F9.3, 6H CASE F9.3, 6H MEMØ F9.3, 1X 2A4, 1X
     1 F10.2, 8X)
 2700 FØRMAT(1H 5HDATE F9.3,6H CASE F9.3, 6H MEMØ F9.3, 1X I8, 1X
     1 F10.2, 8X)
 2800 FØRMAT(1H 30HRESULTS ØF PRØGRAM 2542F DATE F9.3, 6H CASE F9.3,
     1 6H MEMØ F9.3, 3X)
 2900 FØRMAT(1H 18A4)
 3000 FØRMAT(1H0 8HPLØTIM = F10.4)
 3100 FØRMAT(1H 10A4)
      END
```

```
CBESSEL
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE BESSEL(XR,XI,C,N,ZNRO,ZNIO,
     1IT)
      DIMENSION ZNRO(25,25),ZNIO(25,25)
C
C
C         SUBROUTINE BESSEL CALCULATES FOR COMPLEX ARGUMENTS THE BESSEL
C       FUNCTIONS OF INTEGRAL ORDER ZERO THROUGH TWENTY-FOUR.  JNXBES
C       IS CALLED TO CALCULATE THE BESSEL FUNCTION OF MAXIMUM ORDER, N,
C       AND OF N-1.   THEN A RECURSIVE FORMULA IS USED TO CALCULATE
C       EACH DIMINISHING ORDER UNTIL THE ZEROTH ORDER IS DETERMINED.
C
      N1=N&1
      B=XR**2&XI**2
      IF (B)2,4,2
C
    4 ZNRO(N1,IT)=0.0D0
      ZNIO(N1,IT)=0.0D0
      ZNRO(N,IT)=0.0D0
      ZNIO(N,IT)=0.0D0
      GO TO 5
C
    2 CONTINUE
      CALL JNXBES(XR,XI,N,C,ZNRO(N1,IT),ZNIO(N1,IT))
      NN=N-1
      CALL JNXBES(XR,XI,NN,C,ZNRO(N,IT),ZNIO(N,IT))
      XRB=XR/B
      XIB=-XI/B
C
    5 DO 1 I=2,N
      NN=N-I
      XNN=FLOAT(NN&1)
      NN1=NN&1
      NN2=NN&2
      NN3=NN&3
      ZNRO(NN1,IT)=2.D0*XNN*(ZNRO(NN2,IT)*XRB-ZNIO(NN2,IT)*XIB)-ZNRO(NN3
     X,
     1IT)
    1 ZNIO(NN1,IT)=2.D0*XNN*(ZNRO(NN2,IT)*XIB&ZNIO(NN2,IT)*XRB)-ZNIO(NN3
     X,
     1IT)
      RETURN
      END
```

```
CCHNTB                                                                CHNTB   1
      IMPLICIT REAL*8 (A-H,O-Z)                                       CHNTB   2
      SUBROUTINE CHNTBL(DVALUE,ZTURNX,LCHNGE)                         CHNTB   3
C                                                                     CHNTB   4
      REAL*8 LAMDA,LA,NSL,NGL,NST,NGT,LAMDA1,LAMDA2,LA1,LA2,LA1F,LAMD1F CHNTB   5
C                                                                     CHNTB   6
      DIMENSION                                                       CHNTB   7
     1          TXCGD(50),TABI(50),TABIX(50),TABZ(50),TW(4,8),        CHNTB   8
     2          OCCUR(4000),NOCCUR(30),DVALUE(16),TABIX1(50),TABIX2(50) CHNTB   9
     3         ,TXCGD1(50),TXCGD2(50),TABI1(50),TABI2(50),TABZ1(50),  CHNTB  10
     4          TABZ2(50)                                             CHNTB  11
C                                                                     CHNTB  12
      EQUIVALENCE                                                     CHNTB  13
     1(OCCUR(004),BETA1 ),(OCCUR(005),BETA2 ),(OCCUR(006),BETA3 ),    CHNTB  14
     2(OCCUR(007),BETA4 ),(OCCUR(008),COST  ),(OCCUR(014),CP2    ),   CHNTB  15
     3(OCCUR(015),CPG   ),(OCCUR(022),DELRHO),(OCCUR(023),DELHC ),    CHNTB  16
     4(OCCUR(024),EPSIL ),(OCCUR(025),F      )                        CHNTB  17
      EQUIVALENCE                                                     CHNTB  18
     1(OCCUR(030),HREF  ),(OCCUR(032),LAMDA ),(OCCUR(033),LA     ),   CHNTB  19
     2(OCCUR(038),NSL   ),(OCCUR(039),NGL   ),(OCCUR(040),NST    ),   CHNTB  20
     3(OCCUR(041),NGT   ),(OCCUR(042),PI    ),(OCCUR(052),RN     ),   CHNTB  21
     4(OCCUR(053),RB    ),(OCCUR(054),RESINT),(OCCUR(058),RHC2   ),   CHNTB  22
     5(OCCUR(064),SINT  ),(OCCUR(066),SQCOST),(OCCUR(069),THETAC),    CHNTB  23
     6(OCCUR(070),TANT  ),(OCCUR(073),TWSTAG),(OCCUR(074),TWO    ),   CHNTB  24
     7(OCCUR(076),THETA ),(OCCUR(084),W     ),(OCCUR(085),WO     ),   CHNTB  25
     8(OCCUR(124),CMOIN1),(OCCUR(125),CMOIN2)                         CHNTB  26
      EQUIVALENCE                                                     CHNTB  27
     1(OCCUR(133),W1    ),(OCCUR(134),THETA1),(OCCUR(135),RN1    ),   CHNTB  28
     2(OCCUR(136),RB1   ),(OCCUR(137),LAMDA1),(OCCUR(138),LA1    ),   CHNTB  29
     3(OCCUR(139),W2    ),(OCCUR(140),THETA2),(OCCUR(141),RN2    ),   CHNTB  30
     4(OCCUR(142),RB2   ),(OCCUR(143),LAMDA2),(OCCUR(144),LA2    ),   CHNTB  31
     5(OCCUR(145),ZTURN ),(OCCUR(146),LA1F  ),(OCCUR(147),RB1F   ),   CHNTB  32
     6(OCCUR(149),TW1   ),(OCCUR(150),THET1F),(OCCUR(151),LAMD1F),    CHNTB  33
     7(OCCUR(168),TW2   ),(OCCUR(169),RN1F  ),(OCCUR(170),W1F    )    CHNTB  34
      EQUIVALENCE                                                     CHNTB  35
     1(OCCUR(190),FACTR2),(OCCUR(191),FACTR3),                        CHNTB  36
     2(OCCUR(192),FACTR4),(OCCUR(193),FACTR5),(OCCUR(194),FACTR6),    CHNTB  37
     3(OCCUR(195),FACTR7),(OCCUR(196),CMOIN ),(OCCUR(204),WTH    ),   CHNTB  38
     4(OCCUR(228),WTOTAL)                                             CHNTB  39
      EQUIVALENCE                                                     CHNTB  40
     1(OCCUR(0844), TXCGD(1)),(OCCUR(0894),  TABI(1)),                CHNTB  41
     2(OCCUR(0944), TABIX(1)),(OCCUR(0994),  TABZ(1)),                CHNTB  42
     3(OCCUR(2644),   TW(1,1)),                                       CHNTB  43
     4(OCCUR(2833),TXCGD1(1)),(OCCUR(2883),TXCGD2(1)),                CHNTB  44
     5(OCCUR(2933), TABI1(1)),(OCCUR(2983), TABI2(1)),                CHNTB  45
     6(OCCUR(3033),TABIX1(1)),(OCCUR(3083),TABIX2(1)),                CHNTB  46
     7(OCCUR(3133), TABZ1(1)),(OCCUR(3183), TABZ2(1))                 CHNTB  47
      EQUIVALENCE                                                     CHNTB  48
     1(NOCCUR(01),JJHOLD),(NOCCUR(04),MAXTAB),(NOCCUR(05),NOSEUP),    CHNTB  49
     2(NOCCUR(10), MHEAT),(NOCCUR(13),MATLNO),(NOCCUR(15),NGECM ),    CHNTB  50
     3(NOCCUR(16),MXTAB1),(NOCCUR(17),MXTAB2),(NOCCUR(20),MATLN1),    CHNTB  51
     4(NOCCUR(21),MATLN2)                                             CHNTB  52
C                                                                     CHNTB  53
      COMMON OCCUR,NOCCUR                                             CHNTB  54
C                                                                     CHNTB  55
C                                                                     CHNTB  56
C                                                                     CHNTB  57
C           SUBROUTINE CHNTBL ASSIGNS THE APPROPRIATE GEOMETRIC AND   CHNTB  58
C        MATERIAL PROPERTIES AS INITIAL VALUES AT THE REENTRY ALTITUDE CHNTB  59
```

```
C         AND AT ALTITUDE ZTURN, WHERE THE INTEGRATION IS RESTARTED
C         AFTER DISCONTINUOUS CHANGE IN CONFIGURATION.  THE TABLES WITH
C         SUBSCRIPT 1 ARE FOR THE INITIAL CONDITIONS AT REENTRY, THOSE
C         SUBSCRIPTED WITH A 2 ARE CONFIGURATION PARAMETERS AT ZTURN.
C
C
      JJHOLD=0
      GO TO(100,200),LCHNGE
  100 ZTURNX=ZTURN
C
C           GEOMETRY DESCRIPTION AT REENTRY ALTITUDE Z0
      DO 101 J=1,50
      TXCGD(J)=TXCGD1(J)
      TABI(J)=TABI1(J)
      TABIX(J)=TABIX1(J)
  101 TABZ(J)=TABZ1(J)
      CMQIN=CMQIN1
      MAXTAB=MXTAB1
      TWO=TW1
      JJ=151
      MATLNO=MATLN1
C
      GO TO (102,103,104),NGEOM
C
C           NGEOM=1   INPUT WEIGHT,CONE HALF ANGLE,NOSE AND BASE RADII
  102 W=W1
      THETA=THETA1
      RN=RN1
      RB=RB1
      LAMDA1=RN1/RB1
      SINTH1=DSIN(THETA1*0.01745329D0)
      COSTH1=DCOS(THETA1*0.01745329D0)
      LA1=RB1*(COSTH1-LAMDA1*(1.0D0-SINTH1))/SINTH1
      GO TO 300
C
C           NGEOM=2   INPUT WEIGHT,BASE RADIUS,CONE HALF ANGLE, AND
C             BLUNTNESS RATIO
  103 W=W1
      THETA=THETA1
      RB=RB1
      RN=RB1*LAMDA1
      SINTH1=DSIN(THETA1*0.01745329D0)
      COSTH1=DCOS(THETA1*0.01745329D0)
      LA1=RB1*(COSTH1-LAMDA1*(1.0D0-SINTH1))/SINTH1
      GO TO 300
C
C           NGEOM=3   INPUT WEIGHT,NOSE RADIUS,BASE RADIUS,AXIAL LENGTH
  104 W=W1
      RN=RN1
      RB=RB1
      TEM=LA1-RN1
      TEST=TEM*TEM-RN1*RN1&RB1*RB1
      IF(TEST.LT.0.0D0)GO TO 35
      THETA=180.0D0*ASINP((-(LA1-RN1)*RN1&RB1*SQRT(TEST))/
     1(TEST&RN1*RN1))/PI
      GO TO 300
C
  200 ZTURNX=-1.0D10
      LA1F=LA*12.0D0
      RB1F=RB*12.0D0
```

```
      THETIF=THETA*57.29578DC
      RNIF=RN*12.0DC
      LAMDIF=LAMDA
      WIF=WTOTAL
C
C           GEOMETRY DESCRIPTION AT ALTITUDE ZTURN
      DO 201 J=1,50
      TXCGD(J)=TXCGD2(J)
      TABI(J)=TABI2(J)
      TABIX(J)=TABIX2(J)
  201 TABZ(J)=TABZ2(J)
      CMQIN=CMQIN2
      MAXTAB=MXTAB2
      TWO=TW2
      JJ=170
      MATLNO=MATLN2
C
      GO TO(202,203,204),NGEOM
C
C           NGEOM=1   INPUT WEIGHT,CONE HALF ANGLE,NOSE AND BASE RADII
  202 W=W2
      THETA=THETA2
      RN=RN2
      RB=RB2
      LAMDA2=RN2/RB2
      SINTH2=DSIN(THETA2*0.C174532900)
      COSTH2=DCOS(THETA2*0.C174532900)
      LA2=RB2*(COSTH2-LAMDA2*(1.CDC-SINTH2))/SINTH2
      GO TO 300
C
C           NGEOM=2   INPUT WEIGHT,BASE RADIUS,CONE HALF ANGLE, AND
C         BLUNTNESS RATIO
  203 W=W2
      THETA=THETA2
      RB=RB2
      RN=RB2*LAMDA2
      SINTH2=DSIN(THETA2*0.C174532900)
      COSTH2=DCOS(THETA2*0.C174532900)
      LA2=RB2*(COSTH2-LAMDA2*(1.CDC-SINTH2))/SINTH2
      GO TO 300
C
C           NGEOM=3   INPUT WEIGHT,NOSE RADIUS,BASE RADIUS,AXIAL LENGTH
  204 W=W2
      RN=RN2
      RB=RB2
      TEM=LA2-RN2
      TEST=TEM*TEM-RN2*RN2&RB2*RB2
      IF(TEST.LT.0.CDC)GO TO 35
      THETA=180.CDO*ASINR((-(LA2-RN2)*RN2&RB2*SQRT(TEST))/
     1(TEST&RN2*RN2))/PI
  300 CONTINUE
C
C           MATLNO=1 IS TEFLON
C           MATLNO=2 IS LTALPHA
C           MATLNO=3 IS OTWR
C           MATLNO=4 IS PHENOLIC NYLON
C           MATLNO=5 IS CARBON PHENOLIC
C           MATLNO=6 IS AN INPUT MATERIAL
C
      IF(MATLNO.GT.6)WRITE(6,1031)MATLNO
```

```
      IF(MATLNO.GT.6)MATLNO=1
   55 GO TO (31,37,32,36,35,33),MATLNO
C
C              TEFLON PROPERTIES
   31   BETA1=0.D0
        BETA2=2.16D4
        BETA3=0.59D0
        BETA4=3.645D4
         HREF=0.D0
            F=485.D0
         RHO2=136.D0
       DELRHO=0.D0
          CP2=0.26D0
          CPC=0.D0
      NSL=0.528D0
      NST=0.282D0
          NGL=0.D0
          NGT=0.D0
        DELHC=0.D0
        EPSIL=0.4D0
      GO TO 34
C
C              LTALPHA PROPERTIES
   37 BETA1=0.D0
      BETA2=0.5417D0
      BETA3=0.5D0
      BETA4=1.6D4
      HREF=0.0D0
      F=207.0D0
      RHO2=66.4D0
      DELRHO=0.0D0
      CP2=0.52D0
      CPC=0.0D0
      NSL=0.57D0
      NST=0.456D0
      NGL=0.0D0
      NGT=0.0D0
      DELHC=0.0D0
      EPSIL=0.75D0
      GO TO 34
C
C              OTWR PROPERTIES
   32   BETA1=0.D0
        BETA2=.185D10
        BETA3=-0.50D0
        BETA4=1.16D5
         HREF=0.D0
            F=2565.D0
         RHO2=82.D0
       DELRHO=20.D0
          CP2=0.22D0
          CPC=0.423D0
          NSL=0.607D0
          NST=0.303D0
          NGL=0.847D0
          NGT=0.372D0
        DELHC=950.D0
        EPSIL=0.5D0
      GO TO 34
C
```

```
C           PHENOLIC NYLON PROPERTIES
   38 BETA1=0.0D0
      BETA2=0.0D0
      BETA3=0.0D0
      BETA4=0.0D0
       HREF=0.0D0
      F=400.0D0
      RHO2=30.0D0
      DELRHO=45.0D0
      CP2=0.38D0
      CPG=0.0D0
      NSL=0.52D0
      NST=0.20D0
      NGL=0.52D0
      NGT=0.20D0
      DELHC=0.0D0
      EPSIL=0.80D0
      GO TO 34
C
C           CARBON PHENOLIC PROPERTIES
   39 BETA1=0.0D0
      BETA2=9.1D5
      BETA3=0.29D0
      BETA4=1.61D5
      HREF=0.0D0
      F=11050.0D0
      RHO2=85.7D0
      DELRHO=0.0D0
      CP2=0.348D0
      CPG=0.0D0
      NSL=0.0D0
      NST=0.0D0
      NGL=0.0D0
      NGT=0.0D0
      DELHC=0.0D0
      EPSIL=0.8D0
      GO TO 34
C
C           INPUT MATERIAL PROPERTIES
   33  BETA1=OCCUR(JJ&1 )
       BETA2=OCCUR(JJ&2 )
      BETA3=OCCUR(JJ&3)
      BETA4=OCCUR(JJ&4)
      HREF=OCCUR(JJ&5)
      F=OCCUR(JJ&6)
      RHO2=OCCUR(JJ&7)
      DELRHO=OCCUR(JJ&8)
      CP2=OCCUR(JJ&9)
      CPG=OCCUR(JJ&10)
      NSL=OCCUR(JJ&11)
      NST=OCCUR(JJ&12)
      NGL=OCCUR(JJ&13)
      NGT=OCCUR(JJ&14)
      DELHC=OCCUR(JJ&15)
      EPSIL=OCCUR(JJ&16)
   34 CONTINUE
C
C           TESTING CONE HALF ANGLE WITH RANGE OF APPLICABILITY
      IF(THETA.LT.4.0D0)GO TO 57
      IF(THETA.LT.40.0D0)GO TO 5
```

```
   57 WRITE(6,1058)THETA                                                 CHNTB300
 1058 FORMAT(1H0 10X,19HINPUT ERROR. THETA=F8.3,40H IS OUTSIDE THE RANGECHNTB301
     1 OF 4 TO 40 DEGREES)                                                CHNTB302
      JJHOLD=-1                                                           CHNTB303
      RETURN                                                              CHNTB304
    5 CONTINUE                                                            CHNTB305
C                                                                         CHNTB306
C           DEFINING TRIGONOMETRIC QUANTITIES                             CHNTB307
      THETAD=THETA                                                        CHNTB308
      THETA=0.01745329D0*THETA                                            CHNTB309
      SINT=SIN(THETA)                                                     CHNTB310
      COST=COS(THETA)                                                     CHNTB311
      TANT=SINT/COST                                                      CHNTB312
      RESINT=1.D0/SINT                                                    CHNTB313
      SOCOST=COST*COST                                                    CHNTB314
C                                                                         CHNTB315
C           DEFINING INITIAL VALUES OF DVALUE ASSOCIATED WITH GEOMETRY    CHNTB316
      CVALUE(5)=W                                                         CHNTB317
      CVALUE(6)=RN                                                        CHNTB318
      CVALUE(7)=RB                                                        CHNTB319
      WO=W                                                                CHNTB320
      WTH=W                                                               CHNTB321
      CVALUE(16)=WTH                                                      CHNTB322
C                                                                         CHNTB323
C           DEFINING INITIAL VALUES FOR WALL TEMPERATURE                  CHNTB324
      TWSTAG=TWO                                                          CHNTB325
      TW(1,1)=TWO                                                         CHNTB326
      TW(4,1)=TWO                                                         CHNTB327
      DO 36 J=1,8                                                         CHNTB328
      TW(2,J)=TWO                                                         CHNTB329
   36 TW(3,J)=TWO                                                         CHNTB330
C                                                                         CHNTB331
C           FOLLOWING FACTORS ARE USED IN EVIL                            CHNTB332
      FACTR2=DLOG(1.11C5D7)                                               CHNTB333
      FACTR3=12.7657D0&2.9984DC                                           CHNTB334
      FACTR4=RHO2*CP2&CPG*DELRHO                                          CHNTB335
      FACTR5=DELRHO*DELHC                                                 CHNTB336
      FACTR6=RHO2*NSL&DELRHO*NGL                                          CHNTB337
      FACTR7=RHO2*NST&DELRHO*NGT                                          CHNTB338
C                                                                         CHNTB339
C           TESTS ON VALUES INPUT FOR OPTIONS                             CHNTB340
      IF(MATLNO.LT.1)WRITE(6,1031)MATLNO                                  CHNTB341
      IF(NOSEOP.GT.1)WRITE(6,1103)NOSEOP                                  CHNTB342
 1103 FORMAT(1H0 10X,24H*****WARNING*****NOSEOP=I3,43H IS NOT ALLOWED, NCHNTB343
     1OSEOP IS TAKEN TO BE = 0.)                                          CHNTB344
      IF(MHEAT.GT.1)WRITE(6,1104)MHEAT                                    CHNTB345
 1104 FORMAT(1H0 10X,23H*****WARNING*****MHEAT=I3,25H IS THE SAME AS MHECHNTB346
     1AT = 1.)                                                            CHNTB347
 1031 FORMAT(1H0 10X,25H*****WARNING***** MATLNO=I3,24H IS THE SAME AS MCHNTB348
     1ATLNO=1)                                                            CHNTB349
      RETURN                                                              CHNTB350
   35 JJHOLD=-1                                                           CHNTB351
      WRITE(6,1035)LCHNGE                                                 CHNTB352
 1035 FORMAT(1H0 10X,6HTABLE ,I2,28H GIVES NEGATIVE SQUARE ROOT.)         CHNTB353
      RETURN                                                              CHNTB354
      END                                                                 CHNTB355
```

```
CCLASSC
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE CLASSC(F321)
      COMMON OCCUR,NOCCUR
      COMMON/END/ITERM
      DIMENSION OCCUR(4000),NOCCUR(30), X(4000), A(516)
      EQUIVALENCE(OCCUR(1),X(1)),(OCCUR(301),A(1)),(NOCCUR(3),IOP)
      EQUIVALENCE (OCCUR(4000),LP)
C
C
C          SUBROUTINE CLASSC CONTAINS CLASSIC FUNCTIONS WHICH ARE
C       DESIGNED TO PROVIDE DIFFICULT TESTS OF HOW WELL A MINIMIZATION
C       ROUTINE WORKS.
C
C
      IF(IOP.EQ. 3) GO TO 300
      IF(IOP.EQ.4) GO TO 401
      IF(IOP.EQ.5) GO TO 501
      F= 100.0D0*(X(2)-X(1)**2)**2 & (1.0D0 - X(1))**2 &A(1)
      X(100) =F
      IF( IABS(IOP).EQ. 1) GO TO 999
C
  211 B= A(11) & (X(1)-A(7) )*(A(10) &(X(1)-A(7))*(A(9)&(X(1)-A(7))*
     1A(8)))
      C= A(16) & (X(1)-A(12))*(A(15) &(X(1)-A(12))*(A(14)&(X(1)-A(12))*
     1A(13)))
      X(101)= A(2)*(X(2)- B)
      X(102)= A(3)*(X(2)- C)
      GO TO 999
C
  300 FA =0.0D0
      FB =0.0D0
      FC =0.0D0
      FD =0.0D0
      FA = X(1)**2 &X(2)**2&2.D0*X(3)**2&X(4)**2 -5.0D0*X(1)-5.0D0*X(2)-
     121.0D0*X(3)& 7.0D0*X(4)& A(17)
      X(200) =FA
  301 FA2= 8.0D0& X(4)-X(3)&X(2)-X(1)-X(4)**2-X(3)**2-X(2)**2-X(1)**2
      X(201) =FA2
  302 FA3=10.0D0& X(4)&X(1) -2.0D0*X(4)**2 -X(3)**2- 2.0D0*X(2)**2
     1-X(1)**2
      X(202) =FA3
  303 FA4= 5.0D0&X(4)&X(2)-2.0D0*X(1) -2.0D0*X(1)**2-X(2)**2-X(3)**2
      X(203) =FA4
      GO TO 999
C
  401 F= 4.0D0*X(1)& X(2) &A(25)*(1.0D0/X(1) & 1.0D0/X(2) ) & A(26)
      X(300) =F
      GO TO 999
C
  501 F=A(1)*X(1)**2 &A(2)*X(1)*X(2) & A(3)*X(2)**2
     1&A(4)*X(1) & A(5)*X(2) &A(6)
      X(400)= F
      GO TO 999
C
  999 F321=0.0
      LP = 1
      RETURN
      END
```

```
CCOMP62
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE COMP62(H,RHO,ASOUND,T,P)
      DIMENSION TB(26),HB(26),BM(26),R(26),PB(26),TATM(26)
C
C         SUBROUTINE COMP62 COMPUTES THE NON-DIMENSIONAL DENSITY AND
C      PRESSURE, TEMPERATURE IN DEGREES RANKINE, AND SPEED OF SOUND
C      IN FEET PER SECOND OF THE 1962 STANDARD ATMOSPHERE.  FROM
C      Z=0.0 FEET TO Z=300000. FEET THE SUBROUTINE IS EXACT.  FROM
C      Z=300000. FEET TO 2,275,000. FEET. COMPARES TO LESS THAN 1/2
C      OF 1 PER CENT.  ABOVE Z=2,275,000. FEET CONSTANT VALUES ARE
C      USED.
C
      DATA TB/
     1 320.65D0,  288.15D0,  216.65D0,  216.65D0,  228.65D0,  270.65D0,
     X 270.65D0,
     2 252.65D0,  180.65D0,  180.65D0,  210.65D0,  235.65D0,  260.65D0,
     X 360.65D0,
     3 560.65D0,  960.65D0,  1110.65D0,  1210.65D0,  1350.65D0,  1550.65
     XD0,
     4 1830.65D0,  2160.65D0,  2290.65D0,  2420.65D0,  2590.65D0,  2700.
     X65D0/
C
      DATA HB/
     1 -5000.0D0,  0.0D0,  11000.0D0,  20000.0D0,  32000.0D0,  47000.0D0
     X,  52000.0D0,
     2 61000.0D0,  79000.0D0,  88743.0D0,  98451.0D0,  103294.0D0,  1081
     X29.0D0,
     3 117776.0D0,  127394.0D0,  146541.0D0,  156071.0D0,  165571.0D0,
     X184485.0D0,
     4 221967.0D0,  286476.0D0,  376312.0D0,  420240.0D0,  463526.0D0,
     X548230.0D0,
     5 630530.0D0/
C
      DATA BM/
     1 -0.0065D0,  -0.0065D0,  0.0D0,  0.001D0,  0.0028D0,  0.0D0,  -0.0
     X020D0,
     2 -0.004D0,  0.0D0,  0.0033902D0,  0.0051620900D0,  0.0051706310D0,
     3 0.0103659100,  0.0207943400,  0.0208910D0,  0.0157390D0,  0.0105263
     XD0,
     4 0.0074019300,  0.0053359000,  0.0043404800,  0.0036733600,  0.002
     X9593880D0,
     5 0.0030032800,  0.0020069900,  0.0013365700,  0.0D0/
C
      DATA R/
     1 0.193049997D+1, 0.122500263D+1, 0.363918856D-0, 0.880350657D-1,
     2 0.132250458D-1, 0.142753743D-2, 0.759434283D-3, 0.251091339D-3,
     3 0.200114310D-4, 0.317015171D-5, 0.497409776D-6, 0.211574012D-6,
     4 0.982986927D-7, 0.243629359D-7, 0.759159245D-8, 0.183655895D-8,
     5 0.115938455D-8, 0.804025792D-9, 0.434910767D-9, 0.156481698D-9,
     6 .358881794D-10, .650950819D-11, .312789872D-11, .157970712D-11,
     7                     0.464882739D-12, 0.154055260D-12/
C
      DATA PB/
     1 0.120676249D+4, 0.472681269D+3, 0.105576616D+3, 0.255403660D+2,
     2 0.404931098D+1, 0.517378375D+0, 0.275239632D-0, 0.849500768D-1,
     3 0.484093063D-2, 0.766885906D-3, 0.140309744D-3, 0.667954773D-4,
     4 0.343097486D-4, 0.117659643D-4, 0.569949880D-5, 0.236255541D-5,
     5 0.172430381D-5, 0.130345970D-5, 0.786596574D-6, 0.324727509D-6,
```

```
     6 0.8797634170-7, -0.1883396430-7, 0.9594460430-8, 0.5120562690-8,
     7 0.1612728610-8, 0.5571266340-9/
C
      DATA TWTM/
     1 28.964400,   28.964400,   28.964400,   28.964400,   28.964400,   28.96
     X4400,
     2 28.964400,   28.964400,   28.964400,   28.964400,   28.8800,   28.7500
     X,  28.5600,
     3 28.0700,   27.5800,   26.9200,   26.6600,   26.4000,   25.8500,   24.70
     XDO,  22.6600,
     4 19.9400,   18.8200,   17.9400,   16.8400,   16.1700/
C
      DATA AG /9.8066500/, AMO /28.964400/, ARR /8.31432D&3/,
     2                    AR /6356766.000/, CON1 /0.304800/, CON2 /1.8DO
     X/,
     3 CON3 /0.0021156D0/, CON4 /263.8305200/
C
C
      IF((H-2275000.000) -.0000010012,2,3
C
    3 RHO  =.1348651420-12
      ASOUND=3413.42933D0
        T   =2713.9566100
        P   =.1260680330-11
      GO TO 100
C
    2 ZZ   = H * CON1
                  AH      = AR * ZZ / (AR & ZZ)
            DO 7 I = 1,26
            IF(AH - HB(I)) 7,8,9
    9       IF(AH - HB(I&1)) 8,7,7
    8             K       = I
            GO TO 10
C
    7       CONTINUE
C
   10 WS4  =(AH -HB(K))
                  TM      = TB(K) &(BM(K) * (WS4))
      KK=K
      IF(KK.EQ.26)KK=25
      WTM=TWTM(KK)&(AH-HB(KK))*(TWTM(KK&1)-TWTM(KK))/(HB(KK&1)-HB(KK))
        T    =  TM * CON2*(WTM / AMO)
C
   12       IF(BM(K) - 0.DO) 15,16,15
C
   15 WS5  = (TB(K) / TM )
      WS6  = ( AG * AMO / (ARR  * BM(K)))
      RHO  =(R(K) * WS5 **(1.DO & WS6))/ R(2)
        P  =(PB(K)* WS5 **(WS6) ) * CON3
            GO TO 17
C
   16 WS7  = EXP ( - (  AG * AMO  * WS4)/( ARR * TB(K)))
      RHO  = R(K) * WS7/R(2)
        P  = PB(K)* WS7* CON3
C
   17 CONTINUE
      ASOUND = CON4          * SQRT  ( T / WTM )
  100       RETURN
      END
```

```
CCAVDON
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE DAVDON
      COMMON/IOPT/IPROC,IN,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON/FOPT/ERR,PRAND, FAC1, DELTA1
      COMMON /END/ ITERM
      COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FO,E, P,TO,RS,SL, Z,Q,A,EL,
     2DELTA,FAC,C(40,10)
      COMMON/SENSE/ISEN1,ISEN2
      COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC
C
C
C           SUBROUTINE DAVDON IS THE CONTROLLING SUBROUTINE FOR THE
C         CALCULATIONS OF THE DAVIDON VARIABLE METRIC METHOD OF MINIMI-
C         ZATION, WHICH DETERMINES NUMERICALLY THE LOCAL MINIMA OF
C         DIFFERENTIABLE FUNCTIONS OF SEVERAL VARIABLES.  IN THE PROCESS
C         OF LOCATING EACH MINIMUM, A MATRIX H WHICH CHARACTERIZES THE
C         BEHAVIOR OF THE FUNCTION ABOUT THE MINIMUM IS DETERMINED.
C
C
      FLIM  =0.0D0
      N=IN
      K=IRAND
      FAC = FAC1
      DELTA = DELTA1
      NC=0
      ITLIM=  LIMIT
      E    = ERR
C     VARIABLE METRIC MINIMIZATION
  101 MS=0
      IF(FAC)106,1066,106
  106 DO 1064 I=1,N
      DO 105 J=1,N
  105 H(I,J) =0.0D0
      H(I,I)=FAC
 1064 CONTINUE
      GO TO 109
C
 1066 DO 1068  I=1,N
 1067 DO 1068 J=I,N
 1068 H(J,I)=H(I,J)
C
  109 WRITE(6,5)
      WRITE(6,1)
  111 WRITE(6,6)  N, K, E, P, DELTA
  112 WRITE(6,7) (X(I),I=1,N)
  113 WRITE(6,8)
  114 DO 115 I=1,N
  115 WRITE (6,9) (H(I,J),J=1,N)
  116 M1=1
 1165 F=0.0
C
C
  117 CALL FCN(N,G,F,X,M1)
      IF(ITERM.NE.0) GO TO 149
C
 1175 IF(F-FLIM) 1393,1393,118
  118 L=1
```

```
 1185 IT=0
  119 WRITE (6,14)IT,MS,F
 1195 WRITE (6,7)(X(I),I=1,N)
  120 WRITE (6,4)
 1201 IF (NC)121,121,1202
 1202 DO 1209 J1=1,NC
 1203 CALL MATMP (N,N,H,C(1,J1),T)
 1204 CALL MATMP (1,N,T,C(1,J1),TO)
 1205 IF (TO-E)1209,1209,1206
 1206 DO 1208 I=1,N
 1207 DO 1208 J=1,N
 1208 H(I,J)=H(I,J)-T(I)*T(J)/TO
 1209 CONTINUE
C
  121 CALL READY
      IF(ITERM.NE.0) GO TO 149
C
  122 L=L
  123 GO TO (139,159,137,126),L
  124 L=2
  125 GO TO 1201
C
  126 CALL AIM
      IF(ITERM.NE.0) GO TO 149
C
  127 L=L
  128 GO TO (129,135,137),L
C
  129 CALL FIRE
      IF(ITERM.NE.0) GO TO 149
C
  130 L=L
  131 GO TO (135,132,126),L
  132 L=1
C
  133 CALL DRESS
C
      FD= F-FLIM
      IF((FD.LE.0.000).OR.(IT.GT.ITLIM) ) GO TO 142
 1335 L=L
  134 GO TO (124,162),L
  135 L=2
  136 GO TO 133
  137 L=3
  138 GO TO 133
  159 L=4
  1.1 GO TO 133
  139 IF(ISEN1) 1395,1395,1393
 1393 WRITE (6,7)(X(I),I=1,N)
 1395 CALL STUFF
  140 L=L
  141 GO TO (1165,142),L
  142 WRITE (6,10)
      IF(IT.LE.ITLIM) GO TO 143
      WRITE(6,900) IT
  900 FORMAT(1H /10X,'PROGRAM HAS NOT CONVERGED AFTER', I3,'ITERATIONS'
     1, 1H /)
  143 WRITE (6,11)
  144 DO 145 I=1,N
  145 WRITE (6,9)(H(I,J),J=1,N)
```

```
  146 WRITE (6,13)F,GS
  147 WRITE (6,7)(X(I),I=1,N)
  148 WRITE (6,12)(G(I),I=1,N)
C
C
  149 M1=3
  150 CALL FCN(N,G,F,X,M1)
C
C
  151 IF(ISEN1) 152,152, 157
  152 CONTINUE
  156 GO TO 157
    1 FORMAT (72H IDENTIFICATION TITLE
   1                                    )
    2 FORMAT (3I6,F6.0)
    3 FORMAT (6D12.5)
    4 FORMAT (20H0- - - - - - - - - - -)
    5 FORMAT (29H1VARIABLE METRIC MINIMIZATION)
    6 FORMAT (3H0N=I2,4H  K=I2,4H  E=1PE14.5,4H  P=E14.5,8H  DELTA=E14.5
   1)
    7 FORMAT (3H0X=1P8E14.5/(3H0  8E14.5))
    8 FORMAT (2H0H)
    9 FORMAT (1H01P8E14.5/(1H08E14.5))
   10 FORMAT (13H0FINAL VALUES)
   11 FORMAT (13H0ERROR MATRIX)
   12 FORMAT (3H0G=1P8E14.5/(3H0  8E14.5))
   13 FORMAT (3H0F=1PE14.5,5H  GS=E14.5)
   14 FORMAT (4H0IT I4,7H  STEP I4,4H  F=1PE14.5)
  157 CONTINUE
  158 CONTINUE
  162 RETURN
      END
```

```
CDEREQ
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE DEREQ(DVAL,ZUSE,DERIV,LL)
C
      REAL*8 MDOT,LAMDA,MINF
C
      DIMENSION ALPTAB(75),CDTAB(75),WHTAB(75),WCDTAB(75),HTAB(75)
      DIMENSION OCCUR(4000),NOCCUR(30),MDOT(4,8)
      DIMENSION DVAL(16),DERIV(16)
C
      EQUIVALENCE
     1(OCCUR(001),AREF  ),(OCCUR(002),ALPHA ),(OCCUR(003),ALPHIM),
     2(OCCUR(016),CD    ),(OCCUR(026),GAMF  ),(OCCUR(032),LAMDA ),
     3(OCCUR(035),MINF  ),(OCCUR(043),P     ),(OCCUR(044),PHI   ),
     4(OCCUR(045),PSI   ),(OCCUR(050),Q     ),
     5(OCCUR(052),RN    ),(OCCUR(053),RB    ),(OCCUR(059),RNDOT ),
     6(OCCUR(060),RBDOT ),(OCCUR(065),SMR   ),(OCCUR(071),THEALP),
     7(OCCUR(073),TWSTAG),(OCCUR(074),TWO   ),(OCCUR(080),TIMER ),
     8(OCCUR(082),V     ),(OCCUR(084),W     ),(OCCUR(086),WDOT  ),
     9(OCCUR(087),XR    ),(OCCUR(090),XBAR  ),(OCCUR(091),Z     )
      EQUIVALENCE
     1(OCCUR(188),AWREF ),(OCCUR(199),YR    ),(OCCUR(200),PSIALP),
     2(OCCUR(204),WTH   ),(OCCUR(237),XUP   ),
     3(OCCUR(2708), MDOT(1,1)),(OCCUR(3233),HTAB(1)),
     4(OCCUR(3308),WHTAB(1)),(OCCUR(3383),CDTAB(1)),
     5(OCCUR(3458),WCDTAB(1)),(OCCUR(3646),ALPTAB(1))
      EQUIVALENCE
     1(NOCCUR(03),  MOPT),(NOCCUR(05),NOSEOP),(NOCCUR(10), MHEAT),
     2(NOCCUR(18),MAXCD ),(NOCCUR(19),MAXWCD),(NOCCUR(30),INALPH)
C
      COMMON OCCUR,NOCCUR
C
C
C
C          SUBROUTINE DEREQ CALLS IN SUBROUTINES WHICH CALCULATE THE
C       DERIVATIVES INTEGRATED BY SUBROUTINE ADM4RK.
C
C
C          INTERNAL PROGRAM VARIABLES ARE SET EQUAL TO CORRESPONDING
C       INTEGRATED VALUES FROM INTEGRATION SUBROUTINE ADM4RK.
    1 V=DVAL(1)
      GAMF=DVAL(2)
      TIME=DVAL(3)
      Z=ZUSE
      XR=DVAL(4)
      W=DVAL(5)
      RN=DVAL(6)
      RB=DVAL(7)
      PSI=DVAL(8)
      THEALP=DVAL(9)
      PHI=DVAL(10)
      Q=DVAL(11)
      SMR=DVAL(12)
      P=DVAL(13)
      YR=DVAL(14)
      PSIALP=DVAL(15)
      WTH=DVAL(16)
      TIMER=TIME
      IF(Z.LT.0.0D0)GO TO 2
```

```
C
C        SUBROUTINE PRELIM DOES PRELIMINARY CALCULATIONS OF GEOMETRIC,
C     FLOW FIELD, AND THRUSTING PARAMETERS.
   10 CONTINUE
      CALL PRELIM(LL)
      IF(LL.EQ.6)RETURN
C
C        PROGRAM WILL NOT RUN WHEN THE FREE STREAM MACH NUMBER IS
C     LESS THAN 5.
      IF(MINF.GE.5.D0)GO TO 20
      WRITE(6,1001)
 1001 FORMAT(1H0 10X,15HMINF IS LESS THAN 5)
      LL=6
      RETURN
C
C        PROGRAM WILL NOT RUN WHEN THE BLUNTNESS RATIO IS GREATER
C     THAN 0.6.
   20 IF(LAMDA.LE.0.6D0)GO TO 22
      WRITE(6,1002)
 1002 FORMAT(1H0 10X,25HLAMDA IS GREATER THAN 0.6)
      LL=6
      RETURN
   22 CONTINUE
C
C        NO HEATING OR MASS LOSS CALCULATIONS IN RAREFIED FLOW.
C        HEATING AND MASS LOSS CALCULATIONS ARE ONLY PERFORMED IN
C     CONTINUUM FLOW AND IN THE FAIRING REGIME BETWEEN STRONG
C        INTERACTION AND CONTINUUM WHERE XBAR IS LESS THAN XUP.
      IF(XBAR.LT.XUP)GO TO 3
    8 TWSTAG=TWO
      DERIV(5)=0.D0
      DERIV(6)=0.D0
      DERIV(7)=0.D0
      MDOT(2,8)=0.D0
      MDOT(3,8)=0.D0
      GO TO 4
C
C        IF MOPT=0 THE ENTIRE MASS LOSS AND HEATING BLOCK OF EQUATIONS
C     IS BYPASSED.
    3 IF(MOPT.EQ.0)GO TO 6
C
C        SUBROUTINE AERODY PERFORMS AERO. HEATING CALCULATIONS WHEN
C     MOPT = 1.
      CALL AERODY
      IF(MHEAT.EQ.0)GO TO 8
C
C        SUBROUTINE MASSLO CONTROLS THE CALLING OF SUBROUTINE EVIL
C     WHICH CALCULATES MASS LOSS RATES, RECESSION RATES, AND WALL
C     TEMPERATURE.   TOMALO INTEGRATES THE MASS LOSS RATES OVER THE
C     VEHICLE TO OBTAIN THE RATE OF CHANGE IN WEIGHT DUE TO ABLATION.
C        THE VALUE OF THE FOLLOWING DERIVATIVE IS CALCULATED IN TOMALO
C        DERIV(5)=W DOT
      CALL MASSLO
    5 CALL TOMALO
      DERIV(5)=WDOT
C
C        SUBROUTINE NOSEBL CALCULATES VEHICLE SHAPE CHANGE WHEN
C     NOSTOP = 1.  THE VALUES OF THE FOLLOWING DERIVATIVES ARE
C     CALCULATED IN NOSEBL.
    6 IF(NOSTOP.EQ.1)GO TO 7
```

```
      DERIV(6)=0.D0
      DERIV(7)=0.D0
      GO TO 4
    7 CALL NOSEBL
      DERIV(6)=RNDOT
      DERIV(7)=RBDOT
C
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 ROT TC CALCULATES THE ANGLE OF ATTACK BY ONE OF
C      TWO METHODS DEPENDING ON THE VALUE OF THE INPUT LOPT.
C      LOPT = 0 IS AN UNCOUPLED THREE DEGREE OF FREEDOM CALCULATION.
C      LOPT = 2 IS A SIMPLIFIED ANGLE OF ATTACK SOLUTION REQUIRING NO
C       INTEGRATION.
    4 CONTINUE
      IF(INALPH.GT.0.0D0) GO TO 17
C
C         ANGLE OF ATTACK IS CALCULATED BY SUBROUTINE ROTATE.
      CALL ROTATE(DERIV,LL)
      IF(LL.EQ.6)RETURN
      GO TO 19
   17 CONTINUE
C         THE ANGLE OF ATTACK IS A TABULAR INPUT
      LLL=-1
      ALPHA=TABLE(Z,HTAB,ALPTAB,INALPH,LLL)/57.29578D0
      ALPRIM=ALPHA
      DO 18 IJ = 8,13
      DERIV(IJ) = 0.0D0
   18 CONTINUE
   19 CONTINUE
C
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
      CALL DRAGCO
      GO TO 14
   13 LLL=-1
      CD=TABLE(Z,HTAB,CDTAB,MAXCD,LLL)
   14 CONTINUE
C
C         MAXWCD=NUMBER OF VALUES IN TABLE OF DRAG  COEFF. INCREMENT,
C       WCDTAB, WHICH IS ADDED TO THE CD OBTAINED FROM EITHER DRAGCO OR
C       INPUT CDTAB.   WCDTAB IS BASED ON THE INPUT CONSTANT REFERENCE
C       AREA AWREF, BUT IS MULTIPLIED BY THE RATIO AWREF/AREF BEFORE
C       BEING ADDED TO CD.
      IF(MAXWCD.EQ.0)GO TO 15
      LLL=-1
      WIRECD=TABLE(Z,WHTAB,WCDTAB,MAXWCD,LLL)
      CD=CD+WIRECD*AWREF/AREF
C
C         SUBROUTINE TEQUAT CALCULATES THE TRANSLATIONAL TRAJECTORY
C       DERIVATIVES.
   15 CONTINUE
      CALL TEQUAT(DERIV)
C
C         THE TIME DEPENDENT DERIVATIVES ARE BEING CHANGED TO ALTITUDE
C       DEPENDENT DERIVATIVES.  DERIV(3) = D(TIME)/D(ALTITUDE).
```

```
      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)Z                                                    DEREQ186
 1000 FORMAT(1H0 10X,14HZ IS NEGATIVE (E12.5,1H))                       DEREQ187
      RETURN                                                            DEREQ188
      END                                                               DEREQ189
```

```
CDIVMLT
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE DIVMLT(A,B,C,D,E,F)
C
C         SUBROUTINE DIVMLT CALCULATES THE REAL AND IMAGINARY COMPO-
C       NENTS OF THE FIRST ORDER NEUMANN FUNCTION USING THE ZEROTH
C       ORDER NEUMANN FUNCTION COMPONENTS.
C
      R=C**2&D**2
      IF (R)1,2,1
    2 E=0.0D0
      F=0.0D0
      GO TO 3
C
    1 CONTINUE
      G=C/R
      H=-D/R
      E=A*G-B*H
      F=A*H&B*G
    3 RETURN
      END
```

```
CDRAGCO
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE DRAGCO
C
      REAL*8 LAMDA,LAMWD,LA,MINF,MDOT,ME,MLSTAR,LAMW
C
      DIMENSION  A(514),B(21),CDFINF(2,2,2),TW(4,8),MDOT(4,8),
     1          OCCUR(4000),NOCCUR(30)
C
      EQUIVALENCE
     1(OCCUR(001),AREF  ),(OCCUR(002),ALPHA ),(OCCUR(003),ALPRIM),
     2(OCCUR(009),CODRAG),(OCCUR(010),CAPL  ),(OCCUR(011),COSLAM),
     3(OCCUR(016),CD    ),(OCCUR(017),CPE   ),(OCCUR(018),CPW   ),
     4(OCCUR(021),D     ),(OCCUR(026),GAMF  ),(OCCUR(028),GAMMA ),
     5(OCCUR(029),HSRTO ),(OCCUR(032),LAMDA ),(OCCUR(033),LA    ),
     6(OCCUR(035),MINF  ),(OCCUR(036),ME    ),(OCCUR(042),PI    ),
     7(OCCUR(048),PE    ),(OCCUR(049),PINF  )
      EQUIVALENCE
     1(OCCUR(055),RHOINF),(OCCUR(056),RHOIN1),(OCCUR(061),RHOE  ),
     2(OCCUR(062),REYL  ),(OCCUR(064),SINT  ),(OCCUR(067),SINTM ),
     3(OCCUR(069),THETAD),(OCCUR(070),TANT  ),(OCCUR(072),TINF  ),
     4(OCCUR(074),TWO   ),(OCCUR(075),T     ),(OCCUR(077),TCRIT ),
     5(OCCUR(078),TECON ),(OCCUR(079),TE    ),(OCCUR(080),TIMER )
      EQUIVALENCE
     1(OCCUR(082),V     ),(OCCUR(083),VE    ),(OCCUR(086),WDOT  ),
     2(OCCUR(089),XBARST),(OCCUR(090),XBAR  ),(OCCUR(091),Z     ),
     3(OCCUR(092),ZTR   ),(OCCUR(093),ZETA  ),(OCCUR(095),GRATE ),
     4(OCCUR(098),CDP   ),(OCCUR(099),CDB   ),(OCCUR(100),CDI   ),
     5(OCCUR(101),CDPO  ),(OCCUR(126),XBAR1 ),(OCCUR(148),TWST  ),
     6(OCCUR(197),FACTR9),(OCCUR(225),HWBAR )
      EQUIVALENCE
     1(OCCUR(234),DELCDP),(OCCUR(235),DECDFP),(OCCUR(236),DECFTC),
     2(OCCUR(237),XUP   ),(OCCUR(238),XLOW  ),(OCCUR(239),X1UP  ),
     3(OCCUR(240),X1LOW ),(OCCUR(246),TZTEST)
      EQUIVALENCE
     1(OCCUR(0301),      A(1)),(OCCUR(0823),      B(1)),
     2(OCCUR(2644),  TW(1,1)),(OCCUR(2708),MDOT(1,1)),
     3(OCCUR(2793),CDFINF(1,1,1))
      EQUIVALENCE
     1 (NOCCUR(03),  MUPT),(NOCCUR(07),  LCPT),(NOCCUR(10), MHEAT)
C
      COMMON OCCUR,NOCCUR
C
C
C
C         SUBROUTINE DRAGCO CALCULATES THE DRAG COEFFICIENT WHEN INPUT
C       DRAG COEFFICIENT OPTION IS NOT USED.
C
C         FREE MOLECULE FLOW REGIME.      XBAR1 ISGREATER THAN OR EQUAL
C       TO X1UP.
C         STRONG INTERACTION FLOW REGIME.   XBAR1 IS LESS THAN X1LOW
C       AND XBAR GREATER THAN OR EQUAL TO XUP.
C         FAIRING REGIME BETWEEN FREE MOLECULE AND STRONG INTERACTION.
C       XBAR1 IS LESS THAN X1UP AND GREATER THAN OR EQUAL TO X1LOW.
C
C
      PEPINF=PE/PINF
      SQRT3=1.7320508075688770O
      TEMLAM=1.0D0-COSLAM*COSLAM
```

```
      F1K=.9D0&SINTM*(-.119D0&.010800*SINTM)
      ALPHAD=180.0D0*ALPHA/PI
C
C          CURVE FIT FOR PRESSURE DRAG COEFF. AT ZERO ANGLE OF ATTACK.
C        THE SET OF COEFFICIENTS UTILIZED IS DETERMINED BY THE VALUE OF
C        THETAD, CONE HALF ANGLE IN DEGREES.
      N=312
      IF(THETAD.GT.20.0D0)GO TO 200
      N=247
      IF(THETAD.GT.10.0D0)GO TO 200
      N=211
  200 X=1.0D0/MINF
      ZZ=1.0D0/THETAD
      Y=LAMDA
      CDPO=0.0D0
      DO 201 I=1,3
      II=I-1
      ISUB1=N&II
      TEMI=X**II
      DO 201 J=1,3
      JJ=J-1
      ISUB2=ISUB1&3*JJ
      TEMJ=1.0D0
      IF(JJ.NE.0)TEMJ=Y**JJ
      DO 201 K=1,4
      KK=K-1
  201 CDPO=CDPO&A(ISUB2&9*KK)*TEMJ*TEMI*ZZ**KK
C
C          CDPLEO IS A SHARP-CONE PRESSURE DRAG COEFFICIENT FOR THE
C        GIVEN THETA.
      CDPLEO=2.0D0*PE/(RHOINF*V*V)
C
C          CALCULATION OF MULTIPLIER WHICH CORRECTS CDP FOR ANGLES OF
C        ATTACK LESS THAN OR EQUAL TO 4 DEGREES.
      IF(DABS(ALPHAD).GT.4.0D0)GO TO 220
      CDLCDO=1.0D0
      IF(ALPHAD.EQ.0.0D0)GO TO 230
      X=THETAD
      Y=DABS(ALPHAD)
      ZZ=LAMDA
      CDLCDO=0.0D0
      DO 202 I=1,4
      II=I-1
      ISUB1=421&II
      TEMI=X**II
      DO 202 J=1,3
      JJ=J-1
      ISUB2=ISUB1&4*JJ
      TEMJ=Y**JJ
      DO 202 K=1,3
      KK=K-1
      TEMK=1.0D0
      IF(KK.NE.0)TEMK=ZZ**KK
  202 CDLCDO=CDLCDO&A(ISUB2&12*KK)*TEMI*TEMJ*TEMK
      IF(CDLCDO.LT.1.0D0)CDLCDO=1.0D0
      GO TO 230
C
C          CALCULATION OF MULTIPLIER WHICH CORRECTS CDP FOR ANGLES OF
C        ATTACK LESS THAN 40 DEGREES BUT GREATER THAN 4 DEGREES.
```

```
  220 TEML=DABS(ALPHAD)
C          IF ANGLE OF ATTACK IS GREATER THAN 40 DEGREES THEN ALPHA OF
C        40 DEGREES IS USED IN CURVE FIT.
      IF(TEML.GT.40.0D0)TEML=40.0D0
      X=DLOG10(THETAD)
      Y=DLOG10(TEML)
      ZZ=LAMDA
      FUN=0.0D0
      DO 222 I=1,4
      II=I-1
      ISUB1=348&II
      TEMI=X**II
      DO 222 J=1,3
      JJ=J-1
      ISUB2=ISUB1&4*JJ
      TEMJ=Y**JJ
      DO 222 K=1,3
      KK=K-1
      TEMK=1.0D0
      IF(KK.NE.0)TEMK=ZZ**KK
  222 FUN=FUN&A(ISUB2&12*KK)*TEMI*TEMJ*TEMK
      IF(FUN.LT.0.0D0)FUN=0.0D0
      CDLCDO=10.0D0**FUN
  230 CONTINUE
C
      CDP=CDPO*CDLCDO
C
      IF(XBAR.LT.XLOW)GO TO 1
C          WHEN XBAR HAS VALUE LESS THAN XLOW, ALL RAREFIED AIR
C        SOLUTIONS ARE BYPASSED AND ONLY CONTINUUM CALCULATIONS ARE
C        PERFORMED.
C
      TEMTSG=TINF*(1.D0&0.5D0*(GAMMA-1.D0)*MINF*MINF)
      TEM = TWST/TEMTSG
      TEM3=DLOG(XBARST)
      SUM=0.0D0
      IF(THETAD.LT.15.0D0)GO TO 204
C
C          SHARP CONE STRONG INTERACTION DRAG COEFF. FOR CONE HALF ANGLE
C        GREATER THAN OR EQUAL TO 15 DEGREES.
      DO 203 I=1,2
      II=I-1
      ISUB1=384&II
      TEM1=TEM3**II
      DO 203 J=1,4
      JJ=J-1
      ISUB2=ISUB1&2*JJ
      TEM2=THETAD**JJ
      DO 203 K=1,2
      KK=K-1
      ISUB=ISUB2&8*KK
  203 SUM=SUM&A(ISUB)*TEM1*TEM2*TEM**KK
      COST=DEXP(SUM)
      GO TO 3
C
C          SHARP CONE STRONG INTERACTION DRAG COEFF. FOR CONE HALF ANGLE
C        LESS THAN 15 DEGREES.
  204 CONTINUE
      DO 2 I=1,2
      II=I-1
```

```
      ISUB1=200&II
      TEM1=TEM3**II
      DO 2 J=1,2
      JJ=J-1
      ISUB2=ISUB1&2*JJ
      TEM2=THETAD**JJ
      DO 2 K=1,2
      KK=K-1
      ISUB=ISUB2&4*KK
    2 SUM=SUM&A(ISUB)*TEM1*TEM2*TEM**KK
      COST=EXP(SUM)
C
C           FREE MOLECULE DRAG COEFFICIENT WITH ACCOMMODATION COEFFICIENT
C        ZETA.
    3 CDF=    EXP(-SINTM*SINTM)
      CDF=CDF*(1.D0/(SINTM*SQRT(PI))&0.5D0*SQRT(TWST/TINF)/(MINF*MINF))
      CDF=CDF&2.D0& 1.D0/(MINF*MINF)&SINT*SQRT(PI*TWST  /TINF)/MINF
      CDF=ZETA*CDF
C
C           NEWTONIAN DRAG COEFFICIENT.
      CON=2.D0*SINT*SINT
C
C           CAPP IS THE SHARP CONE PROBABILITY DISTRIBUTION AS A FUNCTION
C        OF KNUDSEN NUMBER.
      LAMW=CODRAG*SQRT(2.25D0*PI*TWST/(GAMMA*TINF))/MINF
      LAMWD=LAMW/D
      IF(LAMWD.LT.0.04D0)GO TO 19
      CAPP=0.D0
      TEM=DLOG(LAMW/D)
      DO 18 I=1,21
   18 CAPP=CAPP&B(I)*TEM**(I-1)
      GO TO 23
   19 CAPP=0.506D0-0.147D0*(DLOG10(0.04D0/LAMWD))
C
   23 CONTINUE
C           SHARP CONE FREE MOLECULE DRAG COEFFICIENT.
      CDTRFM=CAPP*(CDF-CON)&CON
      IF(LAMDA.LT.1.D0-3)GO TO 30
C
C           FREE MOLECULE DRAG COEFF. ON SPHERICAL NOSE
      CDFMS=0.0D0
      TEM=DLOG(REYL*D/(CAPL*MINF*FACTR9))
      IF(DABS(TEM).LE.1.D0-4)TEM=1.D0-4
      IF(TEM.GT.5.298D0)TEM=5.298D0
      DO 31 J=1,20
   31 CDFMS=CDFMS&A(J&173)*TEM**(J-1)
      CDFMS=2.0D0*CDFMS
C
C           BLUNT CONE FREE MOLECULE TOTAL DRAG COEFFICIENT.
      CDFM=CDFMS*COSLAM*COSLAM&CDTRFM*(1.0D0-LAMDA*LAMDA)
      GO TO 32
C
C           SHARP CONE FREE MOLECULE TOTAL DRAG COEFFICIENT.
   30 CDFM=CDTRFM
C
C           BLUNT CONE STRONG INTERACTION DRAG COEFFICIENT.
   32 CDS=(COST-COPLEO)*(1.0D0-LAMDA*LAMDA)*COS(ALPRIM)&CDP
      IF(XBAR1.LT.XIUP)GO TO 34
C
C           FREE MOLECULE TOTAL DRAG COEFFICIENT.
```

```
      CD=CDFM
      RETURN
C
   34 IF(XBAR1.GE.X1LOW)GO TO 35
      IF(XBAR.LE.XUP)GO TO 5
C
C           STRONG INTERACTION TOTAL DRAG COEFFICIENT.
      CD=CDS
      RETURN
C
C           FAIRING REGION BETWEEN STRONG INTERACTION AND FREE MOLECULE
C        FLOW
   35 W1=(XBAR1-X1LOW)/(X1UP-X1LOW)
      CD=CDFM*W1&CDS*(1.0D0-W1)
      IF(XBAR.GT.XUP)RETURN
      XUP=XBAR
      XLOW=XBAR-2.0D0
      IF(XBAR.LE.2.0D0)XLOW=0.5D0
      RETURN
C
C
C           CONTINUUM FLOW DRAG CALCULATIONS.
C
C
C           KEY FOR SKIN FRICTION DRAG COEFF. DESCRIPTION
C        SHARP=1           BLUNT=2
C        TURBULENT=1       LAMINAR=2
C        WITH BLOWING=1    NO BLOWING=2
C        CDFINF(SHARP OR BLUNT, LAM. OR TURB., BLOWING OR NO BLOWING)
C
C           BASE DRAG COEFFICIENT.
    1 CONTINUE
    5 Y=DABS(WDOT)/(RHOIN1*V*AREF)
      IF(Y.LT.0.0D0)Y=0.0D0
      IF(Y.GT.0.06D0)Y=0.06D0
      X=1.0D0/MINF
      IF(MINF.GT.24.0D0)X=1.0D0/24.0D0
      N=487
      IF(MINF.GE.7.0D0)N=499
      PBPINF=0.0D0
      DO 8 I=1,4
      II=I-1
      ISUB1=N&II
      TEMI=X**II
      DO 8 J=1,3
      JJ=J-1
      ISUB=ISUB1&4*JJ
      TEMJ=1.0D0
      IF(JJ.NE.0)TEMJ=Y**JJ
    8 PBPINF=PBPINF&A(ISUB)*TEMI*TEMJ
      CDB=(1.0D0-PBPINF)*(X**2)/0.7D0
C
    9 CONTINUE
      IF(XBAR.GE.XLOW)GO TO 10
      IF(Z.GE.ZTR)GO TO 10
C
C           TURBULENT DRAG CALCULATIONS.
C
C           SCHULTZ GRUNOW SKIN FRICTION DRAG COEFFICIENT.
      HE=CPE*TE
```

```
      HSTAHE=0.50D0&0.5D0*CPW*TW(3,8)/HE&0.099D0*ME*ME*(GAMMA-1.0D0)
      HSTAR=HSTAHE*HE
      IF(HSTAR.LE.1110.0D0)GO TO 21
      TEM=PE/2116.0D0
      IF(TEM.GE.1.001)TEM=1.001
      TSTAR=0.0D0
      DO 20 I=1,4
      II=I-1
      ISUB1=162&II
      TEMI=HSTAR**II
      DO 20 J=1,3
      JJ=J-1
      ISUB=ISUB1&4*JJ
   20 TSTAR=TSTAR&A(ISUB)*TEMI*(TEM**JJ)
      GO TO 22
   21 TSTAR=3.5964D0*HSTAR
   22 CONTINUE
      IF(TSTAR.LT.0.D0)CALL EXIT
      MUSTAR=32.2D0*2.27D-8*TSTAR**1.5D0/(TSTAR&198.6D0)
      TAU=TSTAR*(1.0D0-0.125D0*DLOG10(PE/2116.D0))/1.8D0
      ZTAU=2.5D0&0.1D0*DTANH(TAU/500.0D0-7.D0)&0.4D0*DTANH(TAU/1000.D0-
     17.D0)&DTANH(TAU/2500.D0-5.8D0)
      RHOSTA=(39.65D0*PE)/(2116.D0*ZTAU*TSTAR)
C
      IF(CAPL.GE.2.D0)GO TO 12
      A(118)=-4.4666D0
      A(119)=156.D0
      A(120)=-665.D0
      GO TO 13
   12 A(118)=0.D0
      A(119)=0.D0
      A(120)=0.D0
      DO 14 I=1,7
      II=I-1
      TEMI=CAPL**II
      A(118)=A(118)&A(I&400)*TEMI
      A(119)=A(119)&A(I&407)*TEMI
   14 A(120)=A(120)&A(I&414)*TEMI
   13 TEM1=DLOG10(RHOSTA*VE/MUSTAR)
      TEM2=A(120)&TEM1*(A(119)&A(118)*TEM1)
      REYSTA=RHOSTA*VE*CAPL/MUSTAR
C
C        CFO= NO BLOWING SKIN FRICTION COEFF. FOR USE IN MASS LOSS
C          CORRECTION.
      CFO=.37D0*1.15D0*RHOSTA*VE*VE/(V*V*RHOIN1*DLOG10(REYSTA)**2.584D0)
      CFO=CFO*RHOIN1*V*V/(RHOE*VE*VE)
      COFINF(1,1,2)=0.5D0*.852D0*VE*VE*RHOSTA/(RHOIN1*TANT*V*V*TEM2)
C
C          BLOWING CORRECTION TO SKIN FRICTION DRAG COEFF.
      COFINF(1,1,1)=COFINF(1,1,2)/(1.D0&1.2D0*MDOT(3,8)/(RHOE*VE*CFO))
C
C          BLUNTNESS CORRECTION TO SKIN FRICTION DRAG COEFF.
      IF(LAMDA.LT.0.32D0)COFINF(2,1,1)=COFINF(1,1,1)*(1.0D0-LAMDA*
     1(0.8D0&0.052D0*MINF))
      IF(LAMDA.GE.0.32D0)COFINF(2,1,1)=COFINF(1,1,1)*(3.764D0-0.01664D0*
     1MINF)
      COFINF(2,1,2)=COFINF(2,1,1)*COFINF(1,1,2)/COFINF(1,1,1)
C
C          INDUCED PRESSURE DRAG COEFFICIENT.
      HW=HWBAR
```

```
      IF((MOPT*MHEAT).EQ.0)HW=0.24D0*TWO/33.86D0
      HR=0.9D0*HSRTO
      HWHR=HW/HR
      TEM =RHOE*VE*CFO
      DTHE1=CFO*(2.0D0*MDOT(3,8)/TEM &1.0DC/(1.0D0&1.2D0*MDOT(3,8)/TEM))
     1/3.8D0
      DTHE2=DTHE1*(0.547D0*HWHR&ME*(0.53D0&0.68D0*HWHR)&ME*ME*(0.083D0&
     10.106D0*HWHR))
      DTHE=DTHE2&1.6D0*MDOT(3,8)/(3.6D0*RHOE*VE)
      CDLCDP=1.11D0*TEMLAM*CDPLEO*DTHE*F1K*SQRT3*ME
      CDI=CDLCDP
C
C           VALUE OF CDSUM FOR FULLY TURBULENT FLOW
      CDSUMT=CDFINF(2,1,1)&CDLCDP
      CDSUM=CDSUMT
      IF(LOPT.GE.3)GO TO 36
      IF(TIMER.GE.TZTEST)GO TO 36
C
C           FAIRED VALUE FOR CDSUM BETWEEN LAMINAR AND TURBULENT FLOW
      TEM1 =CFOZTR*LA/1.5D0
      TEM2=CFO*LA**1.6D0/1.8D0
      TEM3=1.0D0-(TIMER-TZTR)/TF
      PBAR=(TEM1*TEM3**1.5D0-TEM2*TEM3**1.8D0)/(TEM1-TEM2)
      CDI=CDILAM*PBAR&(1.0D0-PBAR)*CDI
      CDFINF(2,1,1)=CDFINF(2,2,1)*PBAR&(1.0DC-PBAR)*CDFINF(2,1,1)
      CDFINF(2,1,2)=CDFINF(2,2,2)*PBAR&(1.0D0-PBAR)*CDFINF(2,1,2)
      CDSUM=CDI&CDFINF(2,1,1)
C
C           AVERAGE ANGLE OF ATTACK OVER CYCLE USED IN CORRECTING CDP.
   36 CONTINUE
      IF(TCRIT.LE.T)GO TO 15
      IF(T.LT.TECON)GO TO 15
      CD=CDSUM&CDB&CDPO*(1.0D0&GRATE/T)
      RETURN
C
C           TOTAL TURBULENT DRAG COEFFICIENT
   15 CONTINUE
      CD=CDSUM&CDB&CDP
      RETURN
C
C
C           LAMINAR DRAG CALCULATIONS.
   10 HSTAHE=0.5D0&0.5D0*CPW*TW(2,8)/(CPE*TE)&0.0935D0*(GAMMA-1.0D0)*ME*
     1ME
      REYSTA=REYL
      CDFINF(1,2,2)=1.53D0*SQRT(PEPINF/REYL)*(VE/V)**1.5D0/TANT
      CDFINF(1,2,2)=CDFINF(1,2,2)*(HSTAHE*CPE*TE/(.2398D0*TINF))**(-.185
     XD0)
C          CFO= NO BLOWING SKIN FRICTION COEFF. FOR USE IN MASS LOSS
C       CORRECTION.
      CFO=1.15D0*TANT*CDFINF(1,2,2)*RHOINI*V*V/(1.53D0*RHOE*VE*VE)
C
C           BLOWING CORRECTION TO SKIN FRICTION DRAG COEFF.
      CDFINF(1,2,1)=CDFINF(1,2,2)/(1.D0&2.000D0*MDOT(2,8)/(RHOE*VE*CFO))
C
C           BLUNTNESS CORRECTION TO SKIN FRICTION DRAG COEFF.
      TEM=DLOG10(REYL)
      IF(LAMDA.GE.0.2D0)N=472
      IF(LAMDA.LT.0.2D0)N=457
      SUM=0.D0
```

```
      DO 16 J=1,3
      JJ=J-1
      ISUB1=N&5*JJ
      TEMJ=1.DO
      IF(JJ.NE.0)TEMJ=LAMDA**JJ
      DO 16 I=1,5
      II=I-1
      ISUB=ISUB1&II
   16 SUM=SUM&A(ISUB)*TEMJ*TEM**II
      CDFINF(2,2,1)=CDFINF(1,2,1)*SUM
      CDFINF(2,2,2)=CDFINF(2,2,1)*CDFINF(1,2,2)/CDFINF(1,2,1)
C
C           DELCDP,DECDFP, AND DECFTC ARE RESPECTIVELY THE LAMINAR
C        INDUCED PRESSURE DRAG COEFFICIENT, PRESSURE INDUCED AND TRANS-
C        VERSE CURVATURE INDUCED SKIN FRICTION DRAG COEFFICIENTS.
      HW=HWBAR
      IF((MOPT*MHEAT).EQ.0)HW=0.24D0*TW(2,8)/33.86D0
      HR=0.9D0*HSRTO
      TEM =RHOE*VE*CFO
      DTHE1=CFO*(2.0D0*MDOT(2,8)/TEM&1.0D0/(1.0D0&1.25D0*MDOT(2,8)/TEM))
     1/(2.0D0*SQRT3)
      FME2=ME*ME
      HWHR=HW/HR
      DTHE2=DTHE1*(A(283)&A(284)*FME2&HWHR*(A(285)&A(286)*FME2&HWHR*(
     1A(287)&A(288)*FME2&HWHR*(A(289)&A(290)*FME2))))
      DTHE=DTHE2&MDOT(2,8)/(3.0D0*RHOE*VE)
      DELCDP=1.33D0*TEMLAM*CDPLEO*DTHE*F1K*SQRT3*ME
      DECDFP=1.5D0*DTHE*TEMLAM            *CDFINF(1,2,2)*F1K
     1*(-0.823D0&0.524D0*TW(2,8)/TE&0.438*FME2)/(ME*SQRT(CAPL))
      DINF=0.058D0&0.968*TW(2,8)/(TINF*MINF*MINF)
      DECFTC=1.5D0*(0.517D0&0.913D0*TW(2,8)/TE&0.0484D0*FME2)
     1*TEMLAM*DTHE*CDFINF(1,2,2)/(FME2*TANT*DINF*SQRT(3.0D0*CAPL))
      CDI=DELCDP&DECDFP&DECFTC
C
      IF(LOPT.GE.3)GO TO 4
C
C           DEFINING QUANTITIES USED IN FAIRING BETWEEN LAMINAR AND
C        TURBULENT FLOW REGIMES.
      CFOZTR=CFO
      TZTR=TIMER
      TF=1.0D0/(0.48667D-4*V*DABS(SIN(GAMF)))
      CDILAM=CDI
      TZTEST=TZTR&TF
    4 CONTINUE
C
      IF(XBAR.GE.XLOW)GO TO 33
      IF(TCRIT.LE.T)GO TO 17
      IF(T.LT.TECON)GO TO 17
C
C           AVERAGE ANGLE OF ATTACK OVER CYCLE USED IN CORRECTING CDP.
      CD=CDB&CDI&CDFINF(2,2,1)&CDPO*(1.D0&GRATE/T)
      RETURN
C
C           FULLY LAMINAR FLOW.
   17 CD=CDB&CDI&CDFINF(2,2,1)&CDP
      RETURN
C
C           FAIRING BETWEEN STRONG INTERACTION AND LAMINAR FLOW.
   33 CDLAM=CDB&CDI&CDP&CDFINF(2,2,1)
      W2=(XBAR-XLOW)/(XUP-XLOW)
```

```
      CD=CDS*W2&CDLAM*(1.0D0-W2)                                        DRAGC480
      RETURN                                                            DRAGC481
      END                                                               DRAGC482
```

```
CDRESS
      SUBROUTINE DRESS
        IMPLICIT REAL*8 (A-H,O-Z)
       COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FO,E, P,TO,RS,SL, Z,D,A,EL,
     2DELTA,FAC,C(40,10)
       COMMON/SENSE/ISEN1,ISEN2
       COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC
C
C
C          SUBROUTINE DRESS, PART OF THE DAVIDON MINIMIZATION METHOD,
C        MODIFIES THE METRIC MATRIX H ON THE BASIS OF INFORMATION
C        OBTAINED ABOUT THE FUNCTION ALONG THE DIRECTION UNDER CONSIDER-
C        ATION, S. THE NEW H MUST RETAIN THE INFORMATION WHICH THE
C        PRECEDING ITERATIONS HAVE GIVEN ABOUT THE FUNCTION.
C
C
C
      L=L
      GO TO (500,525,519,510),L
  500 CALL MATMP (N,N,H,G,X)
  501 M=1
  502 CALL MATMP (M,N,X,G,TO)
  503 TP1=SL-GSS**2/TO-E
  504 IF (TP1)524,505,505
  505 DO 507 I=1,N
  506 DO 507 J=1,N
  507 H(I,J)=H(I,J)-X(I)*X(J)/TO
  508 DELTA=DELTA*(EL*GSS/TO)
  509 TO=EL/GSS
C
C
  510 DO 512 I=1,N
  511 DO 512 J=1,N
  512 H(I,J)=H(I,J)&TO*S(I)*S(J)
C
C
  519 F=FB
  520 DO 522 I=1,N
  521 G(I)=GB(I)
  522 X(I)=T(I)
  529 CALL OVERFL(LCP021)
      GO TO (531,530,530),LCP021
  530 CALL OVERFL(LCP021)
      GO TO (531,513,513),LCP021
  513 WRITE (6,1)IT,MS,F,GS
  514 IF(ISEN1) 513,515, 517
  515 WRITE (6,2)(X(I),I=1,N)
       WRITE(6,7)
       DO 101 I = 1,N
  101  WRITE(6,8) (H(I,J),J=1,N)
  516 WRITE (6,3)DELTA
  517 WRITE (6,4)
  518 IT=IT&1
 5165 L=1
C
  523 RETURN
C
  524 WRITE (6,5)
C
```

```
  525 TP1=EL*SL/GSS                                                DRESS 60
  526 DELTA=DELTA*TP1                                              DRESS 61
  527 TO=(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 (4H0IT I4,7H  STEP I4,4H  F=1PE14.5,5H  GS=E14.5)     DRESS 77
    2 FORMAT (3H0X=1P8E14.5/(3H0  8E14.5))                         DRESS 78
    3 FORMAT (7H0DELTA=1PE14.5)                                    DRESS 79
    4 FORMAT (20H0- - - - - - - - - -)                             DRESS 80
    5 FORMAT (9H0COLINEAR)                                         DRESS 81
    6 FORMAT (9H0OVERFLOW)                                         DRESS 82
    7 FORMAT (13H0ERROR MATRIX)                                    DRESS 83
    8 FORMAT (1H01P8E14.5/(1H08E14.5))                             DRESS 84
    9 FORMAT (3H0F=1PE14.5)                                        DRESS 85
   10 FORMAT (3H0G=1P8E14.5/(3H0  8E14.5))                         DRESS 86
   11 FORMAT (6E12.5)                                              DRESS 87
      END                                                          DRESS 88
```

```
CDRLIM
      REAL * 8  R(2) , A(2)
      COMPLEX  *16  C , C2
      SUBROUTINE DRLIM(C,R )
       EQUIVALENCE  (C2 , A )
C
C        SUBROUTINE DRLIM TAKES THE REAL AND IMAGINARY PARTS OF A
C      COMPLEX NUMBER AND SETS THEM EQUAL TO TWO REAL NUMBERS.
C
       C2 = C
      R (1) = A(1)
      R (2) = A(2)
        RETURN
     END
```

```
CEFFECT
CEFFECT JUNE 3
      SUBROUTINE EFFECT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON OCCUR,NOCCUR
      COMMON /CIOCUR/ IOCCUR(320)
      COMMON /CPCCUR/ PCCUR(11770)
      DIMENSION IOP(90), NOCCUR(30), OCCUR(4000)
       DIMENSION  SRS(9)
       DIMENSION  DUMMY(100)
      EQUIVALENCE (IOCCUR(00001), IOP   )
      EQUIVALENCE (PCCUR(11757), PFD   )
      EQUIVALENCE (PCCUR(11758), HDIF  )
      EQUIVALENCE (OCCUR(3962), PD    )
       EQUIVALENCE (PCCUR(11598), SRS   )
       EQUIVALENCE (OCCUR(3901), DUMMY )
      EQUIVALENCE (OCCUR(03965), SIGMA)
      DATA A, CON, EPS, B / .7071067811865475, .05D0, .0000100,
     * .0227501319481791D /
C
C
C          SUBROUTINE EFFECT CALCULATES THE QUANTITY PD, WHICH IS THE
C        PROBABILITY THAT A DECOY WILL BE DISCRIMINATED AS BEING A
C        DECOY.
C
C
      PD = 0.D0
C
       DO 11  I = 1, 9
       IF (I.GT.3) GO TO 5
      IF(IOP(I&3).EQ.1)PD=PD&SRS(I)*OCCUR(I&3952)
       GO TO 11
5      IF(IOP(I&24).EQ.1) PD = PD&SRS(I)*OCCUR(I&3952)
11    CONTINUE
C
      PD = DSQRT(PD/HDIF)
      SIGMA = PD
C
C
C
      TOLD = -2.D0
      Y = B
      K = 0
10    K = K & 1
      TX = A*(TOLD & DABS(TOLD)*CON)
      Z = .5D0*(1.D0 & DERF(TX))
      TNEW = TOLD -CON*DABS(TOLD)*(Y-PFD)/(Z-Y)
      W = .5D0*(1.D0&DERF(A*TNEW))
      IF(DABS(W-Y).LE.EPS*DABS(W)) GO TO 99
       IF(K.EQ.20) GO TO 90
      TOLD = TNEW
      Y = W
      GO TO 10
   90  PD = 1.0D0
      WRITE(6,100)
       RETURN
99    CONTINUE
C
C
```

```
C
      PD = .5D0 * (1.0D0 & DERF(A*(TNEW & PD)))
C
      RETURN
 100  FORMAT(1H0, 16HERROR IN EFFECT   )
      END
```

```
CEVIL
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE EVIL(ZZ)
C
      REAL*8 MDOT,NST,MDTWR,K1,K2,LBAR
C
      DIMENSION OCCUR(400),NOCCUR(30),TW(4,8),QDOT(4,8),MDOT(4,8)
C
      EQUIVALENCE
     1(OCCUR(004),BETA1 ),(OCCUR(005),BETA2 ),(OCCUR(006),BETA3 ),
     2(OCCUR(007),BETA4 ),(OCCUR(014),CP2   ),(OCCUR(022),DELRHO),
     3(OCCUR(024),EPSIL ),(OCCUR(025),F     ),(OCCUR(029),HSRTJ ),
     4(OCCUR(030),HREF  ),(OCCUR(031),HS    ),
     5(OCCUR(040),NST   ),(OCCUR(047),PS    ),(OCCUR(058),RHO2  ),
     6(OCCUR(068),SPD   ),(OCCUR(074),TW1   ),(OCCUR(091),Z     ),
     7(OCCUR(092),ZTP   ),(OCCUR(132),TINIT ),
     8(OCCUR(190),FACTR2),(OCCUR(191),FACTR3),(OCCUR(192),FACTR4),
     9(OCCUR(193),FACTR5),(OCCUR(194),FACTR6),(OCCUR(195),FACTR7)
      EQUIVALENCE
     1(OCCUR(225),HWBAR ),(OCCUR(2644),TW(1,1)),
     2(OCCUR(2676),QDOT(1,1)),(OCCUR(2708),MDOT(1,1))
      EQUIVALENCE
     1(NOCCUR(01),JJHOLD),(NOCCUR(02),KKHOLD),(NOCCUR(13),MATLNO)
C
      COMMON OCCUR,NOCCUR
C
C
C
C        SUBROUTINE EVIL CALCULATES THE MASS LOSS RATE - MDOT, THE
C      WALL RECESSION RATE - SDOT, 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 LTWR 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 MDOT AND SDOT AS A FUNCTION OF
C      COLD WALL HEATING FROM AERODY, AND TW AS A FUNCTION OF SDOT.
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.
      J=JJHOLD
      K=KKHOLD
      ICOUNT=0
      IF(J.NE.2)GO TO 12
      TW(3,K)=TW(2,K)
      TW(4,1)=TW(2,1)
   12 CONTINUE
      IF(MATLNO.LT.3)GO TO 14
      IF(MATLNO.EQ.4)GO TO 52
      IF(MATLNO.GE.6)GO TO 14
      IF(Z.GT.ZTR)GO TO 52
   14 CONTINUE
      ZZ=TW(J,K)
      CS=0.0500*TW(J,K)625000.0/TW(J,K)
```

```
    1 ICOUNT=ICOUNT&1
      TEM=DLOG(BETA2*ZZ**BETA3)-BETA4/ZZ
      SPD=BETA1*ZZ&EXP(TEM)
      TK1=FACTR2-1.11120 5/ZZ
      K1=EXP(TK1)
      PBAR=PS/2117.D0
      IF(K1.LT.(1.0D-36))K1=0.0D0
      K2=SQRT(12.7657D0*K1*K1&4.D0*K1*2.99340D*(K1&4.D0*PBAR)*(FACTR3))
      K2=K2-12.7657D0*K1
      K2=0.5D0*K2/(K1&4.0D0*PBAR)
      GAM1=13.654D0&469.585D0*K2
      GAM2=0.25601 2D0&0.5558D-2*K2
      GAM3=0.5345D-5-0.427D-6*K2
      HWBAR=GAM1&ZZ*(GAM2&GAM3*ZZ)
      HWBAR=HWBAR/33.86D0
      E1BAR=HWBAR/35.69D0
      XCOM=0.349D0*HSRTO
      E2BAR=(XCOM&0.5D0*HWBAR)/(XCOM&17.945D0)
      E3BAR=0.95D0-(HWBAR-HREF)/HSRTO
      IF(JJHOLD.EQ.2)GO TO 2
      IF(JJHOLD.GE.3)GO TO 3
      E1=-0.037D0
      E2=0.D0
      GO TO 4
    2 E1=0.D0
      E2=-0.185D0
      GO TO 4
    3 E1=0.D0
      E2=-0.502D0
    4 QDOTOF=QDOT(JJHOLD,KKHOLD)*(E1BAR**E1)*(E2BAR**E2)
      IF(QDOTOF.LE.0.0D0)GO TO 77
      TEM=ZZ-TINIT
      IF(Z.GE.ZTR)FBAR=SPD*HS*FACTR6/QDOTOF
      IF(Z.LT.ZTR)FBAR=SPD*HS*FACTR7/QDOTOF
      PHIB=EXP(-FBAR*(1.D0&0.618D0*FBAR))
      QDOTT=0.47583D-12*EPSIL*ZZ**4
      QDOTC=QDOTOF*E3BAR*PHIB
      IF(MATLNO.EQ.5)GO TO 55
      QDOTS=SPD*RHO2*F
      LBAR=SPD*(TEM*FACTR4&FACTR5)
      RBAR=QDOTC-QDOTT-QDOTS
      ARGU=LBAR-RBAR
   51 CONTINUE
      IF(ICOUNT.GT.100)GO TO 50
    5 IF(DABS(ARGU).LT.1.0D0)GO TO 50
    6 IF(DABS(ARGU).LT.DABS(0.01D0*LBAR))GO TO 50
    7 IF(ARGU)8,50,9
    8 IF(DS)11,50,10
    9 IF(DS)10,50,11
   11 DS=-0.5D0*DS
   10 ZZ=ZZ&DS
      GO TO 1
   50 ZZ=DMAX1(ZZ,0.0D0)
   77 IF(QDOTOF.LE.0.0D0)SPD=0.0D0
      SPD=DMAX1(SPD,0.0D0)
      QDOT(J,K)=SPD*(RHO2&DELRHO)
      RETURN
C
C
   52 IF(MATLNO.EQ.5)GO TO 54
```

```
C
C          QTWR CALCULATIONS FOR LAMINAR FLOW
      QDOTCW=QDOT(JJHOLD,KKHOLD)
      IF(QDOTCW.LT.1.7D1)GO TO 56
      IF(QDOTCW.LT.1.56D1)GO TO 58
      IF(QDOTCW.LT.2.5CD2)GO TO 59
      IF(QDOTCW.LT.3.CD3)GO TO 62
      MDOT(J,K)=QDOTCW*(1.CDC-1.5D3/HS)/(3.64D3&8.1D0*HSRTO)
      GO TO 57
   62 MDOT(J,K)=-1.05650025D-3&7.61118699D-5*QDCTCW&3.34251700D-8*QDOTCW
     1**2-6.91682422D-12*QDOTCW**3
      GO TO 57
   56 MDOT(J,K)=C.0D0
      GO TO 57
   58 MDOT(J,K)=C.CC0210C&C.CCC15DC*(QDOTCW-13.0D0)
      GO TO 57
   59 MDOT(J,K)=-1.2742433SD-3&1.36C71670D-4*QDCTCW-1.09991516D-6*QDOTCW
     1**2&7.98275747D-9*QDOTCW**3-1.65210579D-11*QDOTCW**4
   57 MOTWR=MDOT(J,K)
      IF(QDOTCW.LT.1.CD3)GO TO 75
      IF(QDOTCW.LT.3.0D3)GO TO 72
      SPD=MOTWR/(RHO2&DFLRHO)
      GO TO 73
   75 SPD=1.6115D-6*QDOTCW-5.23741D-4
      GO TO 73
   72 SPD=-1.11196760D-4&4.C3376719D-7*QDOTCW&9.70131261D-10*QDOTCW**2-
     12.45527564D-13*QDOTCW**3
   73 IF(SPD.LT.1.CD-13)SPD=1.CD-13
      SOTWR=SPD
      IF(MATLNO.EQ.4)GO TO 53
      SPDL=DLOG10(SPD)
      IF(SPDL.LT.-1.2583D1)SPDL=-12.583D0
      ZZ=6.34634512D3&5.5C628796D2*SPDL&1.96585366D1*SPDL**2
      IF(ZZ.GT.4.85D3)ZZ=485C.CD0
      RETURN
C
C          PHENOLIC NYLON CALCULATIONS
   53 CONTINUE
      QDOTCW=QDOT(JJHOLD,KKHOLD)
      QDOTCW=DMAX1(QDOTCW,C.CDC)
      IF(QDOTCW.LE.1.0D2)GO TO 60
      IF(QDOTCW.GT.3.CD3)GO TO 74
      MDOT(J,K)=-1.62367647D-3&1.78922793D-4*QDCTCW&1.32113696D-8*
     1QDOTCW**2-5.C8747513D-12*QDOTCW**3
      GO TO 61
   60 MDOT(J,K)=10.0D0**(-2.52288D0&7.3759D-3*QDOTCW)
      GO TO 61
   74 MDOT(J,K)=QDOTCW*(1.CDC-1.7D3/HS)/(1.845D3&1.1D1*HSRTO)
   61 CONTINUE
      IF(MDOT(J,K).LT.1.CD-4)MDOT(J,K)=0.0D0
      IF(MOTWR.LT.1.0D-5)MOTWR=1.0D-5
C          PHENOLIC NYLON SDOT IS RATIOED FROM QTWR SDOT
      SPD=(SOTWR*MDOT(J,K))/MOTWR
      IF(SPD.LT.1.0D-5)SPD=1.CD-5
      SPDL=DLOG10(SPD)
      IF(SPD.LT.1.CD-4)GO TO 76
      ZZ=6.94474035D3&6.45347146D2*SPDL-1.48589173D2*SPDL**2
      IF(ZZ.GT.4.85D3)ZZ=485C.0DC
      RETURN
   76 ZZ=250.0D0&1756.0DC*(SPDL&5.0DC)
```

```
      RETURN
C
C           CARDON PHENOLIC CALCULATIONS FOR LAMINAR FLCW
   54 CONTINUE
      QDOTCW=QDOT(JJHOLD,KKHOLD)
      IF(QDOTCW.LT.8.ODC)GO TO 64
      IF(QDOTCW.LT.1.1?D1)GO TO 65
      IF(QDOTCW.LT.2.31D1)GO TO 66
      IF(QDOTCW.LE.1.CD3)GO TO 67
      MDOT(J,K)=1.2328400?D-2&3.3CR58843D-5*QDOTCW&4.6?267808D-9*CDOTCW*
     1*2
      GO TO 68
   64 MDOT(J,K)=C.ODO
      GO TO 69
   65 MDOT(J,K)=6.578?D-4-1.55773D-4*QDOTCW&9.?485D-6*QDCTCW**2
      GO TO 69
   66 MDOT(J,K)=1.1D-4&5.76C22D-5*(QDCTCW-1.13D1)
      GO TO 68
   67 MDOT(J,K)=-1.73043255D-3&1.25965766D-4*QDCTCW-8.569390320-8*QDOTCW
     1**2&4.655483120-12*QDOTCW**3
   68 CONTINUE
      IF(QDOTCW.LT.2.47D1)SPD=1.CD-18
      IF(QDOTCW.LT.2.47D1)GO TO 69
      IF(QDOTCW.LT.2.75D2)GO TO 70
      IF(QDOTCW.LT.1.CD3)GO TO 71
      SPD=-3.196484D-4&3.065D-7*QDOTCW&2.171792D-10*QDCTCW**2
      GO TO 69
   70 SPD=-2.122965590-&1.18955951D-7*QDCTCW&1.40098706D-10*CDOTCW**2
      GO TO 69
   71 SPD=7.66757321D-5-4.55C33115D-7*QDOTCW&1.55581653D-9*QDOTCW**2
     1-8.4921539CD-13*QDOTCW**3
   69 CONTINUE
   6? SPDL=DLOG1C(SPD)
      ZZ=6.6874113403&4.45431845D2*SPDL&1.2099142301*SPDL**2
      IF(ZZ.GT.6.16D3)ZZ=616C.ODC
      RETURN
C
C           CARBON PHENOLIC CALCULATIONS FOR TURBULENT FLCW
   55 CONTINUE
      QDNET=QDOTC-QDOTT
      ABAR=C22*(616C.ODC-TWC)&F
      HBAR=HST
      QSTAR=ABAR&BBAR*HSRTO
      MDOT(J,K)=QDNET/QSTAR
      IF(QDNET.LT.C.ODC)MDOT(J,K)=C.ODO
      SPD=(?.8D?*MDOT(J,K))/RHO2
      IF(SPD.LE.1.CD-1C)SPD=1.CD-10
      SPDL=DLOG10(SPD)
      IF(SPDL.LT.-1.805E01)SPDL=-18.05800
      ZZ=6.6874113403&4.4643184502*SPDL&1.2099142301*SPDL**2
      RETURN
      END
```

```
C FCN
      SUBROUTINE FCN(N,G,F,X,M1)
        IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CIOCUR/ IOCCUR(320)
      COMMON/BLK1/ISO(5),MS,ISO1(6)
      COMMON /END/ ITERM
      COMMON/OOPT/DELX(20)
      COMMON OCCUR, NOCCUR
      DIMENSION X(40),G(40),FO(40)
      DIMENSION OCCUR(4000), NOCCUR(30)
      DIMENSION IOP(96),NPLOT(5),NPSAVE(14)
      EQUIVALENCE(NOCCUR(14),NPRINT), (NOCCUR(24), NPLOT(1))
      EQUIVALENCE (IOCCUR(200), IOP   )
C
C
C         SUBROUTINE FCN MUST COMPUTE THE FUNCTION F AND THE GRADIENT
C       VECTOR G (I.E., THE VECTOR WHOSE ELEMENTS ARE THE FIRST
C        PARTIAL DERIVATIVES OF F WITH RESPECT TO THE PARAMETERS ),
C        GIVEN THE N COORDINATES OF THE POINT X.
C
C
      DO 416 I = 1, 6
      NPSAVE(I) = IOP(I&63)
      IF(I.LT.6)NPSAVE(I&6)=NPLOT(I)
  416 IF(I.LT.4)NPSAVE(I&11)= IOP(I)
C
      IF(M1.NE.3) GO TO 5
C
      NPRINT = 1
      CALL FEV(N,X,F)
      NPRINT = 0
      GO TO 106
C
    5 CALL FEV(N,X,F)
      IF(ITERM.NE.0) GO TO 106
C
      DO 417 I = 1, 6
      IOP(I&63) = 0
      IF(I.LT.6) NPLOT(I) = 0
  417 IF(I.LT.4) IOP(I) = 0
   10 DO 105  I=1,N
      X(I)=X(I)&DELX(I)
      CALL FEV(N,  X, FM)
      IF(ITERM.NE.0) GO TO 106
      G(I) = (FM -F)/DELX(I)
      X(I)=X(I)-DELX(I)
  105 CONTINUE
C
  106 DO 418 I = 1, 6
      IOP(I&63) = NPSAVE(I)
      IF(I.LT.6) NPLOT(I) = NPSAVE(I&6)
  418      IF(I.LT.4) IOP(I) = NPSAVE(I&11)
C
      IF(M1. NE. 1) GO TO 1000
      IRC=1
      IC=1
 1000 WRITE(6,915) IC, MS , F, (X(I),I=1,N)
C
  915 FORMAT(1H /10X,'PRINT FCN',5X, 'PASS',I5, 5X,'RANDOM STEP',I5,5X
```

```
     1'F=',1PE20.8/ 9X 'X' 1P5F20.8/3(10X,1P5F20.8/))
      WRITE(6,916) (G(I),I=1,N)
 916  FORMAT(1H0 9X 'G' 1P5F20.8/(1H 10X 1P5F20.8))
      IC=IC&1
      RETURN
      END
```

```
CFEV
      SUBROUTINE FEV(N,X,VAL)
        IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /IDNOS/ ID1(50), ID2(50)
      COMMON OCCUR, NOCCUR
      COMMON/IOPT/IPROC,IN,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON/MIN/ALOW(20),UP(20), AMULT(20),CALOW(20),CUP(20)
      COMMON/CCPN/D
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      COMMON/END/ITERM
      DIMENSION X(20), PNLTY(20)
      DIMENSION OCCUR(4000), NOCCUR(30)
      EQUIVALENCE(OCCUR(4000),LP)
      EQUIVALENCE (NOCCUR(14),NPRINT)
C
C
C          SUBROUTINE FEV EVALUATES THE PENALTY FUNCTION WHICH IS BEING
C        OPTIMIZED FOR THE POINT DESCRIBED BY THE N VALUES OF THE X
C        ARRAY.
C
C
      ITERM=0
      VAL =0.0D0
C
      DO 10  I=1,N
      IZ=ID1(I)
      OCCUR(IZ)=X(I)
   10 CONTINUE
C
      IF(ICOM(1).NE.1) GO TO 15
      CALL CLASSC(VAL)
      GO TO 22
C
15     CALL  SCREEN(N,X,ISUC,VAL,D)
      IF(ISUC.LT. 0) GO  TO 100
C
C
       CALL  F123(VAL)
C
C
      IF(LP.LT.6) GO TO 20
      ITERM = 1
      GO TO 100
C
   20 CALL MISC (N,X,VAL)
      CALL  EFFECT
C
C
C
   22  IF(NCONS.EQ.0)GO TO 100
      WRITE(6,1000)
1000  FORMAT(1H0'  IZ'4X'LOWER BOUND'4X'UPPER BOUND'6X'OCCUR(IZ)'8X
     1'PENALTY')
C
C
      DO 50 I=1,NCONS
      PNLTY(I) = 0.0D0
      IZ=  ID2(I)
      IF(OCCUR(IZ).GT.CALOW(I))    GO TO 25
```

```
      PNLTY(I) = AMULT(I)*DABS(OCCUR(IZ)-CALOW(I))**IEX
      WRITE(6,1100) IZ,CALOW(I),CUP(I),OCCUR(IZ),PNLTY(I)
      GO TO 50
   25 IF(OCCUR(IZ).LT.CUP(I)) GO TO 27
      PNLTY(I) = AMULT(I)*DABS(OCCUR(IZ)-CUP(I))**IEX
27    WRITE(6,1100) IZ,CALOW(I),CUP(I),OCCUR(IZ),PNLTY(I)
1100  FORMAT(1H I4, 1P4E15.7)
   50 VAL = VAL  & PNLTY(I)
C
C
      D = VAL
      IF(VAL.GT.0.0D0)GO TO 100
      ITERM= -1
C
  100 CONTINUE
      WRITE(6,900) VAL, (X(I), I=1,N)
C
900   FORMAT(1H '*FEV* '2X'F ='1PE15.7,' X ='5E15.7/(1H 29X1P5E15.7))
  999 RETURN
      END
```

```
CFIB1
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE FIB1  (AA,BB,NF,NMIMAX,ACCUR,NFUNC,XMIMAX,YMIMAX)
C
C
C          SUBROUTINE FIB1  UTILIZES A FIBONACCI SEARCH TECHNIQUE TO
C        FIND THE MAXIMUM OR MINIMUM OF A ONE VARIABLE UNIMODAL
C        FUNCTION WITHIN A DEFINED REGION (AA,BB).
C
C
      IW=6
      DIMENSION F(40)
       ITERM = 0
      A=AA
      B=BB
      IF (NMIMAX) 10,11,11
   10 JMIMAX=1
      GO TO 20
   11 JMIMAX=2
   20 IF (B-A) 21,22,23
   21 TEMP=A
      A=B
      B=TEMP
      GO TO 23
   22 L=1
   25 WRITE (IW,26) L
   26 FORMAT(17HO  ERROR OF TYPE  I2)
      ITERM = 1
      GO TO 170
   23 R=B-A
      IF (NFUNC) 31,30,35
   30 IF (ACCUR) 31,31,40
   31 L=2
      GO TO 25
   35 NFUNC2=NFUNC&2
      R1=(1.0D0&SQRT (5.0D0))/2.0D0
      R2=(1.0D0-SQRT (5.0D0))/2.0D0
      ROACC=(R1**NFUNC2-R2**NFUNC2)/(R1-R2)
      GO TO 45
   40 ROACC=R/ACCUR
   45 IF (ROACC-2.0D0) 50,50,51
   50 XMIMAX=(A&B)/2.0D0
       YMIMAX = FUN1(XMIMAX,NF)
      GO TO 170
   51 IF (ROACC-3.0D0) 60,60,61
   60 XL=A&(B-A)/3.0D0
      XR=A&2.0D0*(B-A)/3.0D0
       YL = FUN1(XL,NF)
      IF(ITERM.NE.0) GO TO 160
       YR = FUN1(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      IF(YL-YR) 70,70,71
   70 XMIMAX=XR
      YMIMAX=YR
      GO TO 170
   71 XMIMAX=XL
      YMIMAX=YL
      GO TO 170
   61 K=3
```

```
      E(1) = 1.000
      E(2)=1.000
   74 E(K)=E(K-1)&E(K-2)
      IF (RDACC-E(K)) 80,80,75
   75 IF (K-40) 77,76,76
   76 L=3
      GO TO 25
   77 K=K&1
      GO TO 74
   80 N=K
      XL=A&E(N-2)*R/E(N)
      XR=A&E(N-1)*R/E(N)
       YL = FUN1(XL,NF)
      IF(ITERM.NE.0) GO TO 160
       YR = FUN1(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      J=1
   90 NJ=N-J
      NJONE=N-J-1
      NJTWO=N-J-2
      GO TO (100,101), JMIMAX
  100 IF (YR-YL) 130,130,110
  101 IF (YR-YL) 110,110,130
  110 IF (J-N&3) 120,160,160
  120 B=XR
      A=A
      R=B-A
      XR=XL
      XL=A&E(NJTWO)*R/E(NJ)
      YR=YL
       YL = FUN1(XL,NF)
      IF(ITERM.NE.0) GO TO 160
  125 J=J&1
      GO TO 90
  130 IF (J-N&3) 140,150,150
  140 B=B
      A=XL
      R=B-A
      XL=XR
      XR=A&E(NJONE)*R/E(NJ)
      YL=YR
       YR = FUN1(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      GO TO 125
  150 XMIMAX=XR
      YMIMAX=YR
      GO TO 170
  160 XMIMAX=XL
      YMIMAX=YL
  170 RETURN
      END
```

```
CFIRE
      SUBROUTINE FIRE
        IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BLKD/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FO,E, P,TO,RS,SL, Z,Q,A,EL,
     2DELTA,FAC,C(40,10)
      COMMON/BLKI/M,N,L,LS,MI,MS,NS,IT,K,NC
      COMMON /END/ ITERM
C
C
C        SUBROUTINE FIRE, PART OF THE DAVIDON MINIMIZATION CALCULA-
C     TIONS, EVALUATES THE PENALTY FUNCTION AND ITS GRADIENT AT THE
C     INTERPOLATED POINT AND DETERMINES IF THE LOCAL MINIMUM HAS BEEN
C     SUFFICIENTLY WELL LOCATED.  IF SO, THE RATE OF CHANGE OF THE
C     GRADIENT IS EVALUATED BY INTERPOLATING FROM ITS VALUES AT THE
C     END POINTS OF THE INTERVAL BEING CONSIDERED AND ITS VALUE AT
C     THE INTERPOLATED POINT.
C
C
  400 MI=2
  401 CALL FCN(N,GB,FB,T,MI)
      IF(ITERM.EQ.0) GO TO 402
C
      DO 10 I = 1,N
10    X(I) = T(I)
      GO TO 412
C
  402 M=1
  403 CALL MATMP (M,N,S,GB,GSB)
      TP1=  DMIN1(F,FP)
  405 IF (TP1-FB&E)418,406,406
  406 TP1=A/(1.0-A)
  407 TP2=(1.0-A)/A
  408 TO=GSB*(TP1-TP2)
  409 IF(DABS(TO)-Q) 413,410,410
  410 GSS=2.0*Q
  411 L=1
C
  412 RETURN
C
  413 GSS=TO&2.0*Q
  414 DO 415 I=1,N
  415 G(I)=(GB(I)-G(I))*TP1&(GP(I)-GB(I))*TP2
C
  416 L=2
  417 GO TO 412
C
  418 IF (F-FP)419,428,428
  419 WRITE(6,1)
  420 EL=(1.0-A)*EL
  421 FP=FB
  422 GSP=GSB
  423 DO 425 I=1,N
  424 XP(I)=T(I)
  425 GP(I)=GB(I)
C
  426 L=3
  427 GO TO 412
C
```

```
428 WRITE(6,2)
429 FL=FL*A
430 F=FB
431 GS=GSB
432 DO 434 I=1,N
433 X(I)=T(I)
434 G(I)=GB(I)
435 GO TO 426
  1 FORMAT (10HCMOVE LEFT)
  2 FORMAT (11HCMOVE RIGHT)
    END
```

```
CFLOWE
      SUBROUTINE FLOWE (ENET,B1,IND)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /NINPUT/ AKW,RHOW,CRHOW,DELWH,      TABL,DHCHEM,RHUSL,
     1   RTG,BCON,ACON,     B21,B22,B23,     C(159)
      COMMON /TBLS12/ ERNTBL(8,10),ERNRTB(10),ERNUTB(8),
     1   ENTABL(25,9),RSTABL(9),HSTABL(25),
     2   ETABL(12,11),THTTBL(11),EMCTBL(12),
     3        D(11,5,6),XYZTBL(11,3)
      COMMON /FPLTMK/ BZ,HN,THETAC,DB,GAMMAE,COTA,COVA,PICIPU,PU,
     1   AMU,RHOU,UU,HU,AMUU,AMINEC,RHOIC,HINEC,UINEC,AMBLSD,
     2   AMABLD,Z,SC,ZBLT
      EQUIVALENCE (OCCUR(3563),CCON)
      DIMENSION OCCUR(4000)
      DIMENSION NOCCUP(30)
      COMMON OCCUP,NOCCUP
      EQUIVALENCE
     1   (RHO,RHOU),(V,UU)
C
C
C
C         SUBROUTINE FLOWE CALCULATES THE ELECTRON DENSITY TRANSITION,
C      N SUB ET  (E/CC), AND THE DECAY RATE, B1.
C
C
C
C       INPUT TABLES
C
C       ERNTBL  = EQUILIBRIUM NORMAL SHOCK ELECTRON DENSITY TABLE
C                 (TABLE F)
C       ERNRTB  = RHO, DENSITY ARGUMENT OF TABLE F (LBM/FT-3)
C       ERNUTB  = VELOCITY ARGUMENT OF TABLE F  (1000 FT/SEC)
C
C       ENTABL  = EQUILIBRIUM ELECTRON DENSITY TABLE   (TABLE B)
C       HSTABL  = H/RTO, ENTHALPY ARGUMENT OF TABLE B, NON-DIMENSIONAL
C       RSTABL  = RHO/RHOO-DENSITY ARGUMENT OF TABLE B, NON-DIMENSIONAL
C
C       ETABL   = TABLE OF MACH NUMBER (TABLE E)
C       EMCTBL  = MC, CONE MACH NUMBER-ARGUMENT OF TABLE E
C       THTTBL  = THETAC, CONE HALF ANGLE-ARGUMENT OF TABLE E, DEGREES
C
C       D       = ELECTRON DENSITY AS A FUNCTION OF NORMALIZED ENTHALPY
C                 AND AIR DENSITY FOR 1000 PPM SODIUM SEED (TABLE D)
C       XDTABL  = RHO/RHOO, AIR DENSITY ARGUMENT OF TABLE D, NON-
C                 DIMENSIONAL
C       YDTABL  = H/RTO, ENTHALPY ARGUMENT OF TABLE D, NON-DIMENSIONAL
C       ZDTABL  = MRAT, THE RATIO OF ABLATED MASS LOSS RATE TO THE MASS
C                 LOSS RATE IN THE BOUNDARY LAYER, ARGUMENT OF TABLE D
C
C
C
C         THE FOLLOWING ARE INPUT QUANTITIES.
C
C       AKW = HEATSHIELD CONDUCTIVITY,(BTU/FT-HR-DEG. R)
C       AMABLD = MASS ABLATED (LBM/ SEC)
C       AMBLSD = MASS SWALLOWED BY BOUNDARY LAYER (LBM/SEC)
C       AMINEC = CONE MACH NUMBER
C       AMU = UPSTREAM MACH NUMBER
```

```
C          AMUU = FREE STREAM VISCOSITY (LBM/FT-SEC)
C          B7 = SCALE HEIGHT (1,000 FT)
C          B21 = SCALING CONSTANT
C          B22 = SCALING CONSTANT
C          B23 = SCALING CONSTANT
C          CDTA = TOTAL DRAG*AREA (FT-SQ)
C          CDVA = VISCOUS DRAG*AREA (FT-SQ)
C          CRHOW = HEATSHIELD SPECIFIC HEAT,(BTU/LB-DEG. R)
C          C(59) = PRESET CONSTANT
C          C(60) = PRESET CONSTANT
C          C(67) =  PRESET CONSTANT
C          C(69) =  PRESET CONSTANT
C          C(83-93) = PRESET CONSTANTS
C          C(100) = PRESET CONSTANT
C          C(115-125) = PRESET CONSTANTS
C          C(130-136) = PRESET CONSTANTS
C          C(159) = PRESET CONSTANT
C          C(160) = PRESET CONSTANT
C          C(164) = PRESET CONSTANT
C          C(165) = PRESET CONSTANT
C          C(169) = PRESET CONSTANT
C          DB = BASE DIAMETER,(FT)
C          DELWH = HEATSHIELD THICKNESS,(IN)
C          DHCHEM = CHEMICAL ENTHALPY OF HEATSHIELD (FT-SQ / SEC-SQ)
C          GAMMAE = ENTRY ANGLE (RAD)
C          HINFC = CONE ENTHALPY (FT-SQ / SEC-SQ)
C          HU = UPSTREAM STATIC ENTHALPY (FT-SQ/SEC-SQ)
C          PICIPU = RATIO OF CONE TO UPSTREAM PRESSURE
C          PU = UPSTREAM PRESSURE (LB/FT-SQ)
C          RHOIC = CONE DENSITY (LBM/ FT-3)
C          RHOSL = SEA LEVEL DENSITY (LBM/FT-3)
C          RHOU  = UPSTREAM DENSITY,(LBM/FT-3)
C          RHOW = HEATSHIELD DENSITY,(LBM/FT-3)
C          RN = NOSE RADIUS (FT)
C          RIO = REFERENCE ENTHALPY (FT-SQ/SEC-SQ)
C          SC = LENGTH OF CONICAL FRUSTUM (FT)
C          TABL = ABLATION TEMPERATURE OF HEATSHIELD (DEG. K)
C          THETAC = CONE ANGLE (RAD)
C          UINFC = CONE VELOCITY (1000 FT/SEC)
C          UU  = UPSTREAM VELOCITY,(1000 FT/SEC)
C          Z = ALTITUDE (1,000 FT)
C          ZBLT = BOUNDARY LAYER TRANSITION ALTITUDE (1,000 FT)
C
C
      PI = 3.141592654
C
C
C          AKV = HEATING CONSTANT
C          REUD = REYNOLDS NUMBER UPSTREAM
C
      REUD = 1.D3*RHOU *UU *DB /AMUU
      AKV = AKW*UU *DSIN(GAMMAE) / (RHOW*CRHOW*DELWH**2.D0)
      ZKEEP = Z
      IF (Z.GE.ZBLT) GO TO 1
      Z = ZBLT
      GO TO 1
C
C
C          TWC = TEMPERATURE WALL CONE (DES. K)
C
```

```
    3 Z = ZKEEP
      TWC = C(160)*(AKV*(ZBLT-Z ))**C(159)/((SC*DEXP(Z/BZ)/DSIN(THETAC)*
     1*2)**.5D0*(AKW/(DELWH*UU **3.D0)))&TWC
      GO TO 2
    1 TWC = 278.D0& C(60)*(AKV*Z )**C(59) / ((SC*DEXP(Z/BZ)/DSIN(THETAC)*
     1*2)**.5D0*(AKW/(DELWH*UU **3.D0)))
      IF (ZKEEP.LT.ZBLT)  GO TO 3
C
C
C        AMBLDS = MASS SWALLOWED BY BOUNDARY LAYER (LBM/ SEC)
C        TW = TEMPERATURE WALL (DEG. K)
C        TWSP = TEMPERATURE WALL SPHERE (DEG. K)
C
    2 TWSP = 278.D0 &C(60)*(AKV*Z )**C(59) / (AKW*DB**.5D0*DEXP(Z /(4.D0
     1    *BZ)) / (UU **1.5D0*DELWH))
      AMBLDS = AMBLSD & AMABLD
      IF ((TABL.LE.TWC.AND.2.D0*RN.LT.0.9999*DB).OR. (TABL.LE.TWSP
     1    .AND.2.D0*RN.GE.0.99D0*DB))   TW=TABL
      IF ( 2.D0*RN.LT.0.99D0*DB.AND.TWC.LT.TABL)   TW=TWC
      IF ( 2.D0*RN.GE.0.99D0*DB.AND.TWSP.LT.TABL)   TW=TWSP
C
C
C        CDSS2A= DRAG*AREA 2ND ENTROPY LAYER (FT-SQ)
C        HW= ENTHALPY AT THE WALL (FT-SQ/ SEC-SQ)
C
       HW = 1.087D4*TW
      CDSS2A = CDTA-CDVA
      IF ( IND.NE.1)  GO TO 203
      WRITE (6,103) REUD,AKV,TWC,TWSP,AMBLDS,TW,HW,CDSS2A
  103 FORMAT (//9X,4HREUD,12X,3HAKV,11X,3HTWC,10X,4HTWSP,9X,6HAMBLDS,
     1    10X,2HTW,14X,2HHW,7X,6HCDSS2A/2X,8D14.5)
C
C
C        AMRAT = RATIO ABLATION TO BOUNDARY LAYER AIR
C        THETSI = SHOCKANGLE UNIVERSAL (RAD)
C
  203 CONTINUE
      AMRAT = AMABLD/AMBLDS
      IF (PICIPU   .GT.(19.D0/6.D0))   GO TO 7
      THETSI =          DARSIN(1.D0/AMU )
      GO TO 8
    7 THETSI =          DARSIN(1.D0/AMU *(6.D0/7.D0*(PICIPU   )-13.D0/7.D0)
     1    **.5D0)
    8 THETSI = DMAX1(THETSI,THETAC)
C
C
C        AMSS2D = MASS FLOW 2ND ENTROPY LAYER (LBM/ SEC)
C        B35 = NSRN FACTOR
C        THETS2 = SHOCKANGLE MAX (RAD)
C        THTS2B = AVERAGE SHOCK ANGLE FOR SECOND ENTROPY LAYER (RAD)
C        U2C = VELOCITY CONE-2ND ENTROPY LAYER (KFT / SEC)
C
      AMSS2D = 2.D3*RHOU *RN**2.D0*DCOTAN(THETSI)**2*UU
      U2C = UU -5.D2*RHOU *UU **2.D0*CDSS2A/AMSS2D
      U2C = DMAX1(U2C,1.D0)
      AI=.286D0&1.C290D0*AMU **2.D0*(1.D0-(U2C/UU )**2.D0)
      THETS2 = DARSIN(1.D0/AMU *((AI&(2.86D0&AI)**.5D0)/2.D0)**.5D0)
      THETS2 = DMAX1(THETS2,THETAC)
      THTS2B = .5D0*(THETS2&THETSI)
      B35  = C(119)*UU **C(120)
```

```
      IF ( IND.NE.1)  GO TO 204
      WRITE (6,104) AMRAT, THETS1,AMSS2D,U2C,A1,THETS2,THTS2D,B35
  104 FORMAT (/7X,5HAMRAT,10X,6HTHETS1,8X,6HAMSS2D,9X,3HU2C,11X,2HA1,
     1  8X,6HTHETS2,10X,6HTHTS2D,8X,3HB35/2X,8D14.5)
C
C
C        AMNDS = MASS FLOW NOSE (LBM/ SEC)
C        SRAT = RATIO BOUNDARY SWALLOWING TO NOSE MASS FLOW
C
  204 CONTINUE
      AMNDS = 3.141592654*RN**2.D0*C(116)*RHOU *UU
      IF (AMBLSD.GE.AMNDS) GO TO 9
      SRAT= AMBLSD/AMNDS
      GO TO 10
    9 SRAT= 1.D0
C
C
C        ENERN = ELECTRON DENSITY AT NOSE CAP (E/CC)
C        ENRN = NO. OF ELETRONS PRODUCED BY NOSE CAP (E/ SEC)
C        H2C = ENTHALPY CONE-2ND ENTROPY LAYER(FT-SQ/ SEC-SQ)
C        P2C = PRESSURE CONE-2ND ENTROPY LAYER (LB/FT-SQ)
C        RHO2C = DENSITY CONE-2ND ENTROPY LAYER (LBM/FT-3)
C
   10 CALL AR2DIM (10, B,RHO,ERNRTB,V,ERNUTB,ENERN,ERATBL)
      ENRN =   30.48D0**3.D0*AMNDS*SRAT*ENERN/RHOU
      P2C = (7.D0*AMU **2.D0*DSIN(THTS2D) **2.D0-1.D0)*PU /6.D0
      H2C  =   HU *P2C*(P2C&6.D0*PU )/(PU *(6.D0*P2C&PU ))
      RHO2C = RHOU *(6.D0*P2C&PU )/(P2C&6.D0*PU)
      IF (AMBLSD.GE.AMSS2D) GO TO 11
C
C
C        HC = ENTHALPY CONE (FT-SQ / SEC-SQ)
C        PC = PRESSURE CONE (LB/FT-SQ)
C        RHOC =  DENSITY CONE (LBM/FT-3)
C        SCHEM = CHEMICAL LENGTH
C        UC = VELOCITY CONE (KFT / SEC)
C
      RHOC = RHO2C
      HC =   H2C
      PC = P2C
      UC = U2C
      SCHEM=RHO2C**C(117)*SC*AMBLSD/(U2C*AMSS2D)
      GO TO 12
   11 RHOC = RHO1C
      HC =   HINFC
      PC = P1C1PU*PU
      UC = UINFC
      SCHEM = RHO2C**C(117)*SC*AMSS2D/(U2C*AMBLSD)   &
     1  RHO1C**C(117)*SC*(AMBLSD-AMSS2D)/(UINFC*AMBLSD)
C
C
C        ENSRN = NO. OF ELECTRONS NOSE CAP -EXP AT SHOULDER (E/ SEC)
C
   12 ENSRN = C(122)*ENRN*AMNDS*(1.D0-C(121) & (1.D0&ENRN*SCHEM)
     1  *C(121)*DEXP(-B35  * SCHEM)) /(AMNDS&C(123)*SCHEM*ENRN
     2  *TN**C(124))
      IF ( IND.NE.1)  GO TO 205
      WRITE (6,105) AMNDS,SRAT,ENERN,ENRN,P2C,U2C,RHO2C,RHOC
  105 FORMAT (/8X,5HAMNDS,11X,4HSRAT,9X,5HENERN,9X,4HENRN,11X,3HP2C,
     1  10X,3HU2C,11X,5HRHO2C,8X,4HRHOC/2X,8D14.5)
```

```
  205 CONTINUE
C
C
C        ENEQH = EQUILIBRIUM ELECTRON DENSITY (E/CC)
C
      RHOC12 = RHOC / RHOSL
      HCRTO = HC/RTO
      IF (AMRAT.EQ.0.D0)  GO TO 13
      ENEQH = AR3DIM(0,11, 5, 4,RHOC12,HCRTO,AMRAT,XYZTBL)
      GO TO 14
   13 CALL AR2DIM ( 9,25,RHOC12,RSTABL,HCRTO,HSTABL,ENECH,ENTABL)
   14 CONTINUE
C
C
C        ENBL = NO. OF ELECTRONS LEAVING BOUNDARY LAYER (E/SEC)
C        ENEBL = ELECTRON DENSITY IN THE BOUNDARY LAYER (E/CC)
C
      ENEBL = ENEQH*(1.D0-DEXP(-B22*SCHEM**C(118)))*
     1   (B21 & B23 * (UC-22.D0)/22.D0)
      ENBL =   30.48D0**3.D0*ENEBL*AMBLDS/PC
      IF (IND.NE.1)  GO TO 206
      WRITE (6,106) HC,PC,UC,SCHEM,ENSRN,ENEQH,ENEBL,ENBL
  106 FORMAT(/11X,2HHC,12X,2HPC,12X,2HUC,13X,5HSCHEM,8X,5HENSRN,10X,
     1    5HENEQH,10X,5HENEBL,8X,4HENBL/2X,8D14.5)
  206 CONTINUE
C
C
C        AM2 = MACH NUMBER 2ND ENTROPY LAYER
C        AM2C = MACH NUMBER CONE-2ND ENTROPY LAYER
C        AMINF = FREE STREAM MACH NUMBER
C        ENS = NO. OF ELECTRONS ENTERING WAKE NECK (E / SEC)
C        H2 = ENTHALPY 2ND ENTROPY LAYER (FT-SQ / SEC-SQ)
C        RHO2 = DENSITY 2ND ENTROPY LAYER (LBM/FT-3)
C        U2 = VELOCITY OF SECOND ENTROPY LAYER (1000 FT/SEC)
C
      ENS = ENBL & ENSRN
      AM2C = U2C*AMU /UU *(HL/H2C)**.5D0
      AM2C = DMAX1(AM2C,1.D0)
      THETCD = THETAC*57.29577951D0
      CALL AR2DIM (11,12,THETCD,THITBL,AM2C,EMCTBL,AM2,ETABL)
      CALL AR2DIM (11,12,THETCD,THITBL,AMINFC,EMCTBL,AMINF,ETABL)
      RHOINF = RHOIC*(((1.D0&.2D0*AMINFC**2.D0)/(1.D0&.2D0*AMINF**2.D0))
     1    **2.5D0
      HINF = HINFC*((1.D0&.2D0*AMINFC**2.D0) / (1.D0&.2D0*AMINF**2.D0))
      UINF = UINFC*AMINF /AMINFC*(HINF/HINFC)**.5D0
      RHO2 = RHO2C*((1.D0&.2D0*AM2C**2.D0) / (1.D0&.2D0*AM2**2.D0))
     1    **2.5D0
      H2 = H2C*((1.D0&.2D0*AM2C**2.D0) / (1.D0&.2D0*AM2**2.D0))
      U2 = U2C*AM2/AM2C*(H2/H2C)**.5D0
      IF (AMBLSD.GE.AMSS2D)  GO TO 21
C
C
C        AMS = MACH NUMBER SHOULDER
C        HS = ENTHALPY SHOULDER (FT-SQ / SEC-SQ)
C        RES = REYNOLDS NUMBER SHOULDER
C        RHOS = DENSITY SHOULDER (LBM/FT-3)
C        US = VELOCITY SHOULDER (KFT / SEC)
C
      AMS = AM2
      RHOS = RHO2
```

```
      HS = H2
      US = U2
      GO TO 22
   21 AMS = AMINF
      RHOS = RHOINF
      HS = HINF
      US = UINF
   22 RES = RHOS*AMS*RCUD*C(169)/(RHOL *AMU )*(HS/HU )**C(67)
      IF ( IND.NE.1)  GO TO 207
      WRITE (6,107) ENS,AM2C,AM2,RHO2,H2,U2,AMS,RHOS
  107 FORMAT (/10X,3HENS,11X,4HAM2C,10X,3HAM2,10X,4HRHO2,11X,2HH2,11X,
     1   2HU2,13X,3HAMS,9X,4HRHOS/2X,8D14.5)
  207 CONTINUE
C
C
C        ENEN = ELECTRON DENSITY AT THE NECK (E/CC)
C        HN = ENTHALPY NECK (FT-SQ / SEC-SQ)
C        HNIHS = RATIO NECK ENTHALPY TO SHOULDER ENTHALPY
C
      CS = C(164)&C(165)*DSIN(THETAC)**.25D0*RES**(-3.D0/8.D0)
     1   *AMS**(-.5D0)
      HN = HW& CS*(HS-HW&.5D&*US**2.D0)-DHCHEM
      HNIHS = 2.D&*HN/(2.D0*HS&1.D&*US**2.D0)
      ENEN = RHOS*ENS*HS/(30.48D&**3.D)*AMHLDS*HN)
      IF ( IND.NE.1)  GO TO 208
      WRITE (6,108) HS,US,RES,HN,HNIHS,RHOS,ENS,ENEN
  108 FORMAT(/10X,2HHS,13X,2HUS,11X,3HRES,11X,2HHN,10X,5HHNIHS,10X,
     1   4HRHOS,12X,3HENS,9X,4HENEN/2X,8D14.5)
  208 CONTINUE
C
C
C        RETS = SHOULDER REYNOLDS NUMBER BASED ON THETA
C        THETSS = SHOULDER WAKE MOMENTUM THICKNESS (FT)
C        THTBLS = INITIAL WAKE MOMENTUM THICKNESS BASED ON SHOULDER
C        CONDITIONS (FT)
C
      THTBLS = (CDTA*RHOU /(2.D0*PI*RHOS))**.5D0*(UU /US)
      IF (AMBLSD.GE.AMSS2D) GO TO 23
      THETSS =  (RHOU /(2.D0*PI*RHOS)*(CDVA& AMSS2D*CDSS2A/AMBLSD))
     1   **.5D0*(UU /US)
      GO TO 24
   23 THETSS = THTBLS
   24 RETS = THETSS*RES/DB
      IF (Z.GT.ZBLT)  GO TO 25
C
C
C        AKE = GTCHEM FACTOR
C        B1 = DECAY RATE
C        B5 =  NET SCALING FACTOR
C        B11 = ELECTRON DECAY RATE FACTOR
C        GT = FUNCTION AERO
C
      GT = 0.D0
      GO TO 26
   25 GT = (1.D0-HN*(1.D0-HN/HS) /(2.D&*C(90)*RHOS*THETSS*US**2.D0
     1   *RETS*(1.D0&HNIHS)))**(-1.D0)
   26 B5 = C(87)*(1.D0& C(84)*AMRAT**C(85)&C(86)*AMRAT**C(87)
     1   &C(88)*(RHOS/RHOSL)**C(89))
      B11 = C(91)&C(92)*AMRAT**C(93)
      B1 =(AMUU *(C(115)&B11*(RHOS/RHOSL)**C(125))/(1.D&*US**2.D)*THTBLS
```

```
     1     ))*CCON**BCON
      AKE = (THETSS)**(-2.DC)-7.5D28*RHOS*C(103)/ENEN
      IF (IND.NE.1)  GO TO 209
      WRITE (6,109) THTBLS,THETSS,RETS,GT,B5,B11,B1,AKE
  109 FORMAT(/7X,6HTHTBLS,10X,6HTHETSS,9X,4HRETS,10X,2HGT,12X,2HB5,
     1    11X,3HB11,14X,2HB1,10X,3HAKE/2X,8D14.5)
  209 CONTINUE
      IF (Z.GT.ZBLT)  GO TO 27
C
C
C        ENET = TRANSITION ELECTRON DENSITY (E/CC)
C        GTCHEM = FUNCTION CHEMICAL
C
      GTCHEM = 1.D0
      GO TO 28
   27 GTCHEM = (1.D0&HN/(4.DC*C(90)*US*AKE*RHOS*THETSS*RETS))**(-1.D0)
   28 ENET =(1.D-3**C(134)*AMUU *B5*RHOS**C(133)*GT/(THETSS*US**C(134))
     1    *(1.D0 -DEXP(-DABS(HN)*  (C(135)  &C(136)*DABS(HN)**C(130)
     2    &C(131)*DABS(HN)**C(132))))   &  1.1D27*RHOS*C(103)
     3    &ENEN*GTCHEM)*CCON**ACON
      IF (IND.NE.1)  GO TO 210
      WRITE (6,110) GTCHEM,ENET
  110 FORMAT (/7X,6HGTCHEM,11X,4HENET/2X,2D14.5//)
C
C
  210 CONTINUE
  113 CONTINUE
      RETURN
      END
```

```
CFMIMAX
      IMPLICIT  REAL*8 (A-H, O-Z)
      FUNCTION FMIMAX(X,NF)
      COMMON/MINSK/N,K,ALB(20),UB(20),XX(20),F1
      COMMON/IOPT/IPROC,IN,NCONS,IPNT,IFX,LIMIT,IRAND
      COMMON/FOPT/ERR,RRAND, FAC, DELTA
C
C
C           IN THE FIBONACCI OPTIMIZATION TECHNIQUE, FUNCTION FMIMAX
C         IS SET EQUAL IN TURN TO EACH OF THE N UNIMODAL FUNCTIONS OF
C         ONE VARIABLE BEING MINIMIZED (MAXIMIZED) IN ACCORDANCE WITH
C         THE VALUE OF NF.
C
C
        GO TO (1,1,3), NF
 1    CALL FFV(1,X, VAL)
      FMIMAX  = VAL
      GO TO 1000
 3    XX(1) = X
      CALL FFV(2,XX,VAL)
      FMIMAX = VAL
 1000 RETURN
      END
```

```
CFUN1
      FUNCTION FUN1(XX,N)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /CSIG/ XTT,WS1,XODP,DWST,SPH2,XLMS,XMSP,XSSP,B1,ZCR2,ZCR4,
     1XB20,DWBST1,SPH4,BC,DDW,WS4,SIGMDS,IND2
C
C
C           FUNCTION FUN1 COMPUTES THE PULSE SHAPE RADAR RETURN AS A
C       FUNCTION OF DISTANCE BEHIND THE LEADING EDGE OF THE RADAR
C          PULSE.
C
C
      SIG1= 0.0D0
      SIG2= 0.0D0
      SIG3= 0.0D0
      SIG4= 0.0D0
      XXTT= XX -  XTT
C
C          Y1 -  REGION AFFECTED BY OVERDENSE SCATTERING
  701 Y1 = DMIN1(WS1,XODP)
      Y1 = DMIN1(Y1,XXTT)
      WS2 = XTT & XODP & WS1 - XX
      Y1 = DMIN1(Y1,WS2)
      IF(XXTT)702,703,703
  703 IF(XXTT-WS1-XODP)704,704,702
C
C          SIG1 - PORTION OF PULSE SHAPE DUE TO OVERDENSE RETURN
  704 SIG1 = Y1 * DWST * SPH2
  702 WS2 = 2.0D0 * XLMS
C
C          Y2 - REGION AFFECTED BY FIRST PART OF MULTIPLE SCATTERING
       Y2 = DMIN1(WS1,WS2)
      WS2 = XXTT - XODP
       Y2 = DMIN1(WS2,Y2)
      WS2 = XTT & XMSP & WS1 -XX
       Y2 = DMIN1(Y2,WS2)
      IF(XXTT - XODP)705,706,706
  706 IF(XXTT - WS1 - XMSP)707,707,705
C          SIG2 - PORTION OF PULSE SHAPE DUE TO CONSTANT PORTION OF
C
C                   MULTIPLE SCATTERING REGION
  707 SIG2 = 0.5D0 * Y2 * DWST * SPH2
  705 WS3 = 0.5D0 * WS1
      IF(WS1 - (XX - XTT - XMSP))708,709,709
C
C          Y31 - UPPER LIMIT OF REGION AFFECTED BY SECOND PART OF
C                  MULTIPLE SCATTERING
  708  Y31 = XX - WS1 - XTT
      GO TO 710
  709  Y31 = XMSP
  710 IF(XX-(XTT & XSSP))711,711,712
C
C          Y32 - LOWER LIMIT OF REGION AFFECTED BY SECOND PART OF
C                  MULTIPLE SCATTERING
  711   Y32 = XXTT
      GO TO 713
  712  Y32 = XSSP
  713 IF(WS1 - (XXTT - XSSP))714,715,715
C
```

```
C        Y41 - UPPER LIMIT OF REGION AFFECTED BY SINGLE SCATTERING
  715  Y41 = XSSP
       GO TO 716
  714  Y41 = XXTT-WS2
C
C        Y42 - LOWER LIMIT OF REGION AFFECTED BY SINGLE SCATTERING
  716   Y42 = XXTT
       IF(XXTT - XMSP)720,721,721
  721  IF(XXTT - (WS1 & XSSP))717,717,720
  717  EX1 =(DEXP(-B1* Y31) -DEXP(-B1* Y32))* ZCR2/16.0D0
       EX2 =(DEXP(-4.0D0* B1* Y31) - DEXP(-4.0D0*B1 * Y32))
C
C        THE FOLLOWING TEST IS USED TO PREVENT UNDERFLOW
       IF(DABS(EX2) - 1.0D-6)9007,9007,9008
 9007  EX2 = 0.0D0
 9008  EX2 = 0.25D0 * ZCR4 * EX2 * ZCR4
       IF(XB20)719,719,718
C
C        SIG3 - PORTION OF PULSE SHAPE DUE TO VARIABLE PORTION OF
C                MULTIPLE SCATTERING REGION
  719  SIG3 = DWBSTI * (EX1 & EX2) & (Y32 - Y31) * SPH4 * B0 * DDW
       GO TO 720
  718  SIG3 = DWBSTI * (EX1 & EX2) & 2.0D0 * WS4 *(DEXP(-0.5D0 * B1 *
     1        XB20*Y31)- DEXP(-0.5D0* B1* XB20*Y32))* B0 * SPH2**(2.0D0-
     1 XB20)/(B1*XB20) *DWST
  720  IF(XXTT - XSSP)730,722,722
  722  EX1 = DEXP(-4.0D0* B1* Y41)- DEXP(-4.0D0* B1* Y42)
       EX2 = DEXP(-B1* Y41) - DEXP(-B1* Y42)
C
C        SIG4 - PORTION OF RETURN DUE TO SINGLE SCATTERING REGION
       SIG4 = 0.25D0*DWBSTI* ZCR4*EX1*ZCR4 & DDW/B1 *(SPH2/16.0D0
     1   *DWST/DDW & B0) *ZCR2 * EX2
C
C        SIG - PULSE SHAPE - RADAR RETURN AS A FUNCTION OF DISTANCE
C              BEHIND THE LEADING EDGE OF THE RADAR PULSE
  730  SIG = SIG1 & SIG2 & SIG3 & SIG4
       IF(IND2-1)30005,10005,30005
10005 WRITE (6,20005) Y1,SIG1,Y2,SIG2,Y31
20005 FORMAT(10X,7H    Y1=,1PE15.7,6H  SIG1=,1PE15.7,6H    Y2=,1PE15.7,10H
     1     SIG2=,1PE15.7,6H  Y31=,1PE15.7)
10006 WRITE (6,20006) Y32,Y41,Y42,SIG3,SIG4
20006 FORMAT(10X,7H    Y32=,1PE15.7,6H   Y41=,1PE15.7,6H   Y42=,1PE15.7,10H
     1     SIG3=,1PE15.7,6H SIG4=,1PE15.7)
10010 WRITE (6,20010) XX,SIG
20010 FORMAT(2X,3HX=,1PE15.7,6H   SIG=,1PE15.7)
30005  GO TO (100,101),N
C
  100    FUN1 = SIG
C
       RETURN
C
  101  FUN1 = (SIGMDS - SIG)**2
C
       RETURN
       END
```

NOT REPRODUCIBLE

```
CF123
      SUBROUTINE F123(XXX)
        IMPLICIT REAL*8 (A-H,O-Z)
C
C
C        SUBROUTINE F123 CAN OPTIONALLY  -  (1) READ DATA FOR A
C      REENTRY VEHICLE TRAJECTORY FROM INPUT OR CALL SUBROUTINE VIXEN
C      TO CALCULATE THIS DATA.  (2) CALL VIXEN TO CALCULATE THE DATA
C      FOR A DECOY TRAJECTORY.  (3) COMPARE THE DATA FOR A REENTRY
C      VEHICLE TRAJECTORY WITH THE DATA FOR A DECOY TRAJECTORY AND
C      SAVE THE DIFFERENCES.  (4) CALL INTGRL TO CALCULATE AND PLOT
C      A CORRIDOR INTEGRAL.  (5) CALL INTGRL TO CALCULATE THE SIGMA
C      SQUARE INTEGRAL TO BE USED BY SUBROUTINE EFFECT.  (6) CALL
C      INTGRL TO CALCULATE A TABLE OF INFLUENCE COEFFICIENTS AND TO
C      PRODUCE INFLUENCE COEFFICIENT PLOTS AT SPECIFIED ALTITUDES.
C
C
      COMMON OCCUR,NOCCUR
      COMMON /CIOCCUR/ IOCCUR(320)
      COMMON /CPOCCUR/ POCCUR(11770)
       COMMON /TTMCOM/ TTM
      COMMON /INFTST/ NDECOY
       REAL * 8 LA1, LA2, MDOT, MW, MX, MY, MZ
      DIMENSION ALMAX(200),ALMIN(200),BCB(40),BCD(40),BCV(40),BCWL1(40)
      DIMENSION BCWL2(40),BCWL3(40),BCWR1(40),BCWR2(40),BCWR3(40)
       DIMENSION  TBMAT(3), TT(200)
      DIMENSION BETAPL(160),BS(160),COFINF(2,2,2),DB(160),DD(160)
      DIMENSION DS(160),DV(160),DVAR(50),DVH(50),DVL(50),DWL1(160)
      DIMENSION DWL2(160),DWL3(160),DWR1(160),DWR2(160),DWR3(160)
      DIMENSION EMAX(200),EMIN(200),H(40),HOLD(3,50),ICP(90),MDOT(4,8)
      DIMENSION NCOMDV(50),NOCCUR(30),NPV(160),OCCUR(4000),PEPSB(6)
      DIMENSION ODOT(4,8),SAVE(9,160,3),SB(40),SD(40),SV(40),SWL1(40)
      DIMENSION SWL2(40),SWL3(40),SWR1(40),SWR2(40),SWR3(40),TAMAX(200)
      DIMENSION TAMIN(200),TCB(40),TCD(40),TCV(40),TCWL1(40),TCWL2(40)
      DIMENSION TCWL3(40),TCWR1(40),TCWR2(40),TCWR3(40),TM(140)
      DIMENSION TPLOT(160),TS(160),TTM(200),VPLOT(160),VS(160)
      DIMENSION VDGPLT(160),WL1P(160),WL1S(160),WL2P(160),WL2S(160)
      DIMENSION WL3P(160),WL3S(160),WR1P(160),WR1S(160),WR2P(160)
      DIMENSION WR2S(160),WR3P(160),WR3S(160),ZMAX(200),ZMIN(200)
      DIMENSION ZPLOT(160),ZS(160)
      EQUIVALENCE (OCCUR(00229), ALALT )
      EQUIVALENCE (OCCUR(01044), ALMAX )
      EQUIVALENCE (OCCUR(01244), ALMIN )
      EQUIVALENCE (OCCUR(00002), ALPHA )
      EQUIVALENCE (OCCUR(00003), ALPRIM)
      EQUIVALENCE (OCCUR(00122), ALST  )
      EQUIVALENCE (OCCUR(00128), CASE  )
      EQUIVALENCE (OCCUR(00099), CDB   )
      EQUIVALENCE (OCCUR(02793), COFINF)
      EQUIVALENCE (OCCUR(00100), CDI   )
      EQUIVALENCE (OCCUR(00098), CDP   )
      EQUIVALENCE (OCCUR(00101), CDPO  )
      EQUIVALENCE (OCCUR(00202), CM    )
      EQUIVALENCE (OCCUR(00203), CN    )
      EQUIVALENCE (OCCUR(00094), CODLAM)
      EQUIVALENCE (OCCUR(00216), CPSZET)
      EQUIVALENCE (OCCUR(00215), CTHZET)
      EQUIVALENCE (OCCUR(00127), DATE  )
      EQUIVALENCE (OCCUR(00235), DECDEP)
```

```
      EQUIVALENCE (OCCUR(00236), DECPTC)
      EQUIVALENCE (OCCUR(00234), DFLCOP)
      EQUIVALENCE (OCCUR(00219), DELY  )
      EQUIVALENCE (OCCUR(00220), DELZ  )
      EQUIVALENCE (OCCUR(00129), EMO   )
      EQUIVALENCE (OCCUR(00197), FACTR9)
      EQUIVALENCE (OCCUR(02244), FMAX  )
      EQUIVALENCE (OCCUR(02444), FMIN  )
      EQUIVALENCE (OCCUR(00105), GAMEO )
      EQUIVALENCE (OCCUR(00020), GAMMA )
      EQUIVALENCE (OCCUR(00139), LA1   )
      EQUIVALENCE (OCCUR(00144), LA2   )
      EQUIVALENCE (OCCUR(02769), MDOT  )
      EQUIVALENCE (OCCUR(00117), MW    )
      EQUIVALENCE (OCCUR(00210), MX    )
      EQUIVALENCE (OCCUR(00211), MY    )
      EQUIVALENCE (OCCUR(00212), MZ    )
      EQUIVALENCE(OCCUR(4001),LP)
      EQUIVALENCE (OCCUR(02801), PEPSO )
      EQUIVALENCE (OCCUR(00112), PHIO  )
      EQUIVALENCE (OCCUR(00223), PSIZET)
      EQUIVALENCE (OCCUR(00114), PSIG  )
      EQUIVALENCE (OCCUR(02676), QDOT  )
      EQUIVALENCE (OCCUR(00110), QG    )
      EQUIVALENCE (OCCUR(00060), RBDOT )
      EQUIVALENCE (OCCUR(00136), RB1   )
      EQUIVALENCE (OCCUR(00142), RB2   )
      EQUIVALENCE (OCCUR(00059), RNDOT )
      EQUIVALENCE (OCCUR(00135), RN1   )
      EQUIVALENCE (OCCUR(00141), RN2   )
      EQUIVALENCE (OCCUR(00116), SIG   )
      EQUIVALENCE (OCCUR(00221), SINGO )
      EQUIVALENCE (OCCUR(00111), SMRO  )
      EQUIVALENCE (OCCUR(00218), SPSZET)
      EQUIVALENCE (OCCUR(00217), STHZET)
      EQUIVALENCE (OCCUR(01444), TAMAX )
      EQUIVALENCE (OCCUR(01644), TAMIN )
      EQUIVALENCE (OCCUR(03643), TBMAT )
      EQUIVALENCE (OCCUR(00201), TH    )
      EQUIVALENCE (OCCUR(00113), THEALO)
      EQUIVALENCE (OCCUR(00224), THEZET)
      EQUIVALENCE (OCCUR(00246), TZTEST)
      EQUIVALENCE (OCCUR(00086), WDOT  )
      EQUIVALENCE (OCCUR(01844), ZMAX  )
      EQUIVALENCE (OCCUR(02044), ZMIN  )
      EQUIVALENCE (PCCUR(06181), BCB   )
      EQUIVALENCE (PCCUR(06101), BCD   )
      EQUIVALENCE (PCCUR(06021), BCV   )
      EQUIVALENCE (PCCUR(06501), BCWL1 )
      EQUIVALENCE (PCCUR(06581), BCWL2 )
      EQUIVALENCE (PCCUR(06661), BCWL3 )
      EQUIVALENCE (PCCUR(06261), BCWR1 )
      EQUIVALENCE (PCCUR(06341), BCWR2 )
      EQUIVALENCE (PCCUR(06421), BCWR3)
      EQUIVALENCE (PCCUR(00641), BTTAPL)
      EQUIVALENCE (PCCUR(03841), BS    )
      EQUIVALENCE (PCCUR(02081), DB    )
      EQUIVALENCE (PCCUR(01921), DD    )
      EQUIVALENCE (PCCUR(03681), DS    )
      EQUIVALENCE (PCCUR(01761), DV    )
```

```
      EQUIVALENCE (PCCUR(11707), DVAP  )
      EQUIVALENCE (PCCUR(11607), DVH   )
      EQUIVALENCE (PCCUR(11657), DVL   )
      EQUIVALENCE (PCCUR(02721), DWL1  )
      EQUIVALENCE (PCCUR(02881), DWL2  )
      EQUIVALENCE (PCCUR(03041), DWL3  )
      EQUIVALENCE (PCCUR(02241), DWR1  )
      EQUIVALENCE (PCCUR(02401), DWR2  )
      EQUIVALENCE (PCCUR(02561), DWR3  )
      EQUIVALENCE (PCCUR(05841), H     )
      EQUIVALENCE (PCCUR(07101), HOLD  )
      EQUIVALENCE (PCCUR(07261), SAVE  )
      EQUIVALENCE (PCCUR(06821), SB    )
      EQUIVALENCE (PCCUR(06781), SD    )
      EQUIVALENCE (PCCUR(06741), SV    )
      EQUIVALENCE (PCCUR(06581), SWL1  )
      EQUIVALENCE (PCCUR(07621), SWL2  )
      EQUIVALENCE (PCCUR(07661), SWL3  )
      EQUIVALENCE (PCCUR(06861), SWR1  )
      EQUIVALENCE (PCCUR(06901), SWR2  )
      EQUIVALENCE (PCCUR(06941), SWR3  )
      EQUIVALENCE (PCCUR(06221), TCB   )
      EQUIVALENCE (PCCUR(06141), TCD   )
      EQUIVALENCE (PCCUR(06061), TCV   )
      EQUIVALENCE (PCCUR(06541), TCWL1 )
      EQUIVALENCE (PCCUR(06621), TCWL2 )
      EQUIVALENCE (PCCUR(06701), TCWL3 )
      EQUIVALENCE (PCCUR(06301), TCWR1 )
      EQUIVALENCE (PCCUR(06381), TCWR2 )
      EQUIVALENCE (PCCUR(06461), TCWR3 )
      EQUIVALENCE (PCCUR(05561), TI    )
      EQUIVALENCE (PCCUR(03201), TS    )
      EQUIVALENCE (PCCUR(03521), VS    )
      EQUIVALENCE (PCCUR(00001), TPLOT )
      EQUIVALENCE (PCCUR(00481), VCPLT)
      EQUIVALENCE (PCCUR(00321), VPLOT )
      EQUIVALENCE (PCCUR(01281), WL1P  )
      EQUIVALENCE (PCCUR(01441), WL2P  )
      EQUIVALENCE (PCCUR(01601), WL3P  )
      EQUIVALENCE (PCCUR(04481), WL1S  )
      EQUIVALENCE (PCCUR(04641), WL2S  )
      EQUIVALENCE (PCCUR(04801), WL3S  )
      EQUIVALENCE (PCCUR(00801), WR1P  )
      EQUIVALENCE (PCCUR(00961), WR2P  )
      EQUIVALENCE (PCCUR(01121), WR3P  )
      EQUIVALENCE (PCCUR(04001), WR1S  )
      EQUIVALENCE (PCCUR(04161), WR2S  )
      EQUIVALENCE (PCCUR(04321), WR3S  )
      EQUIVALENCE (PCCUR(00161), ZPLOT )
      EQUIVALENCE (PCCUR(03361), ZS    )
      EQUIVALENCE(IOCCUR(00308), II    )
      EQUIVALENCE (IOCCUR(00301), IREF  )
      EQUIVALENCE (IOCCUR(00001), IOP   )
      EQUIVALENCE(IOCCUR(00310), JJ    )
      EQUIVALENCE (IOCCUR(00313), K     )
      EQUIVALENCE (IOCCUR(302), LPLOT )
      EQUIVALENCE (IOCCUR(00303), MODE  )
      EQUIVALENCE (IOCCUR(00091), NCOMDV)
      EQUIVALENCE(IOCCUR(00312), NCON  )
      EQUIVALENCE (IOCCUR(304), NCP   )
```

```
      EQUIVALENCE (IOCCUR(CC305), NDECOY)
      EQUIVALENCE(IOCCUR(CC306), NDVCH )
      EQUIVALENCE (NOCCUR(3C), INALPH)
      EQUIVALENCE (NOCCUR(C7), LOPT  )
      EQUIVALENCE (NOCCUR(1C), MHEAT)
      EQUIVALENCE (NOCCUR(C3), MOPT  )
      EQUIVALENCE (NOCCUR(CC5), NOSEOP)
      EQUIVALENCE (NOCCUR(15), NGEOM)
      EQUIVALENCE (NOCCUR(22), NTHRST)
C
C
C
      XXX = 0.D0
C
C            CALLING THE TIMER SUBROUTINE TO RECORD THE START OF
C          CALCULATIONS BY THE PROGRAM.
  10C CALL TIMERS(1,TIMTAB)
C
  40C  NDC = NDECOY
      NDVC = NDVCH
C
C SAVE OCCUR(133-145), OCCUR(2C5-2C9) AND OCCUR(222) IN DVAR(1-19)
      CALL SAVEDV(1,DVAR,OCCUR)
      IF(IREF.NE.2) GO TO 497
      IF(NDECOY.EQ.1) GO TO 497
      DO 111 I = 1, NDVCH
      NCON = NCONDV(I)
C THE VALUE OF THE ITH DESIGN VARIABLE TO BE PERTURBED IS SAVED IN
C
C HOLD(1,I).  THIS VALUE WAS USED TO CALCULATE BASIC DECOY TRAJECTORY.
      HOLD(1,I) = OCCUR(NCON)
C
C HOLD(2,I) SAVES THE PERTURBED VALUE OF THIS VARIABLE TO BE USED FOR
C CALCULATING THE TRAJECTORY OF THE FIRST COMPARISON DECOY.
      HOLD(2,I) = DVL(I)
C
C HOLD(3,I) SAVES THE PERTURBED VALUE OF THIS VARIABLE TO BE USED FOR
C THE TRAJECTORY OF THE SECOND COMPARISON DECOY.
 111  HOLD(3,I) = DVH(I)
C
C FOR ONE PERTURBATION ONLY, THIS VALUE MUST BE SAVED IN HOLD(2,I).
C IF TWO PERTURBATIONS ARE MADE, THE SMALLER VALUE MUST BE SAVED IN
C             HOLD(2,I) AND THE LARGER IN HOLD(3,I).
 497  DO 777 II = 1,NDVC
      IF(IREF.NE.2) GO TO 498
      IF(NDECOY-2)401,402,4C3
C
C A BASIC DECOY TRAJECTORY WILL BE CALCULATED.
 401  NLOW = 1
      NHIGH = 1
      GO TO 404
C
C ONE COMPARISON DECOY TRAJECTORY WILL BE CALCULATED.
 402  NLOW = 2
      NHIGH = 2
      GO TO 404
C
C TWO COMPARISON DECOY TRAJECTORIES WILL BE CALCULATED.
 403  NLOW = 2
      NHIGH = 3
```

```
C
C  OCCUR(NCON) IS THE LOCATION OF THE DESIGN VARIABLE BEING PERTURBED.
 4 6   NCON = NCOMDV(II)
C
C RESTORE DVAR(1-19) TO OCCUR(133-145), OCCUR(205-209) AND OCCUR( 222)
      CALL SAVEDV(2,DVAR,OCCUR)
      GO TO 499
C
C IF IREF EQUALS ONE A REENTRY VEHICLE TRAJECTORY WILL BE CALCULATED.
C IF IREF EQUALS THREE REFERENCE VEHICLE TRAJECTORY DATA MUST BE INPUT.
498   NLOW = 1
      NHIGH = 1
499   DO 444 KK = NLOW,NHIGH
      IF(IREF.NE.2) GO TO 501
C
C PUT LOWER VALUE OF IITH DESIGN VARIABLE TO BE PERTURBED IN OCCUR ARRAY
      IF(KK.EQ.2) OCCUR(NCON) = DVL(II)
C
C PUT UPPER VALUE OF IITH DESIGN VARIABLE TO BE PERTURBED IN OCCUR ARRAY
      IF(KK.EQ.3) OCCUR(NCON) = DVH(II)
501   CONTINUE
      IF(MODE.NE.1) CASE = CASE & .001D0
      WRITE(6,1100)CASE,DATE,EMO
      IF(IREF.EQ.3) GO TO 28
C
C           TESTING THE INPUT QUANTITIES.
      IF(LOPT.LE.4)GO TO 53
      WRITE(6,1053)LOPT
      LOPT=1
   53 IF(NTHRST.LE.2)GO TO 54
      WRITE(6,1054)NTHRST
      NTHRST=0
   54 CONTINUE
      IF(NGEOM.LT.1)WRITE(6,1102)NGEOM
      IF(NGEOM.GT.3)WRITE(6,1102)NGEOM
      IF(NGEOM.LT.1)NGEOM=1
      IF(NGEOM.GT.3)NGEOM=1
      IF(INALPH.EQ.0)GO TO 5
      IF((LOPT*(LOPT-2)).EQ.0)WRITE(6,1077)LOPT
    5 CONTINUE
      IF(NOPT.NE.0)GO TO 55
      IF(MHEAT.NE.0)WRITE(6,1055)MHEAT
      IF(NOSEOP.NE.0)WRITE(6,1056)NOSEOP
   55 IF(MHEAT.NE.0)GO TO 56
      IF(NOSEOP.NE.0)WRITE(6,1057)NOSEOP
   56 CONTINUE
C
C           DEFINING NUMERICAL FACTORS.
      CODLAM=7.6489D-9*MW/(SIG*SIG)
      FACTR9=SQRT(GAMMA)
      ALALI=0.0D0
C
C           SETTING INPUT OPTIONS EQUAL TO A STORAGE QUANTITY IN ORDER
C         TO CARRY THEM OVER TO THE FOLLOWING INPUT CASE, UNLESS THEY ARE
C         REDEFINED.
      LOPTO=LOPT
      NTHRTO=NTHRST
      INALPO=INALPH
C
C           CHANGING THE UNITS FROM INCHES (INPUT UNITS) TO FEET (UNITS
```

NOT REPRODUCIBLE

```
C        USED INTERNALLY).
      RN1=RN1/12.0DC
      RB1=RB1/12.0DC
      LA1=LA1/12.0DC
      RN2=RN2/12.0DC
      RB2=RB2/12.0DC
      LA2=LA2/12.0DO
      DELZ=DELZ/12.0DC
      DELY=DELY/12.0DC
C
C          SETTING THE DRAG COEFFICIENT QUANTITIES TO ZERC.
      CDFINF(2,1,1)=0.DC
      CDFINF(2,2,1)=0.DC
      CDFINF(2,1,2)=0.DC
      CDFINF(2,2,2)=0.DC
      CDB=0.DO
      CDP=0.DC
      CDPC=0.DO
      CDI=0.DC
      DELCDP=0.0DO
      DECDFP=0.0DO
      DECFTC=0.0DC
      TZTEST=0.0DC
C
C          SETTING THE ROTATIONAL QUANTITIES TO ZERO.
      IF(LOPT.EQ.0)GO TO 201
        CP=0.0DC
        CN=0.0DC
      PSIC=0.0DO
        QC=0.0DO
      SMPC=0.0DC
      PHIC=0.0DO
      IF(LOPT.EQ.1)THEALC=0.0DC
  201 CONTINUE
      IF(LOPT.NE.0)GO TO 205
      DO 200 J=1,200
      TAMAX(J)=0.DO
       ZMAX(J)=0.DC
       FMAX(J)=0.DC
      ALMAX(J)=0.DC
      TAMIN(J)=0.DO
       ZMIN(J)=0.DC
       FMIN(J)=0.DO
  200 ALMIN(J)=0.DC
  205 CONTINUE
C
C          FINDING THE TRIGONOMETRIC FUNCTIONS OF THE THRUST OFFSET
C        ANGLES.
      CTHZET=COS(0.017453290DC*THEZET)
      STHZET=SIN(0.017453290DC*THEZET)
      CPSZET=COS(0.017453290DC*PSIZET)
      SPSZET=SIN(0.017453290DC*PSIZET)
C
C          CHANGING THE UNITS OF ANGLES FROM DEGREES TO RADIANS.
      GAMFO=0.017453290DC*GAMFO
      SINGO=DSIN(GAMFO)
       PHIC=0.017453290DC*PHIC
       PSIC=0.017453290DC*PSIC
      THEALO=0.017453290DC*THEALO
      ALST=1.017453290DC*ALST
```

```
C
C          SETTING THE MASS LOSS AND HEATING QUANTITIES TO ZERO.
      QDOT(1,1)=0.0D0
      QDOT(4,1)=0.0D0
      DO 3 J=1,9
      MDOT(1,J)=0.0D0
      MDOT(2,J)=0.0D0
      MDOT(3,J)=0.0D0
      MDOT(4,J)=0.0D0
      PERSB(J)=0.0D0
      QDOT(2,J)=0.0D0
    3 QDOT(3,J)=0.0D0
C
      WDOT=0.0D0
      PNDOT=0.0D0
      RBDOT=0.0D0
C
C          CALLING IN THE TIMER SUBROUTINE TO RECORD THE CALLING IN OF
C       SUBROUTINE VIXEN.
      CALL TIMERS(2,TIMTAB)
C
C          CALLING IN SUBROUTINE VIXEN WHICH PERFORMS THE CALCULATIONS.
      CALL VIXEN
C
C       PERFORM THE CALCULATIONS CALLED FOR BY VIXEN.
C          CALLING IN THE TIMER SUBROUTINE TO MEASURE THE TIME TO
C       CALCULATE THE TRAJECTORY.
      CALL TIMERS(-2,TIMTAB)
C
C
C          CALLING IN THE TIMER SUBROUTINE TO MEASURE THE TOTAL TIME TO
C       READ IN THE INPUT QUANTITIES AND PERFORM THE CALCULATIONS.
      CALL TIMERS(-1,TIMTAB)
C
C          SETTING THE THRUST QUANTITIES TO ZERO.
      TH=0.0D0
      MX=0.0D0
      MY=0.0D0
      MZ=0.0D0
      TBMAT(1)=0.0D0
      TBMAT(2)=0.0D0
      TBMAT(3)=0.0D0
C
C          CHANGING UNITS ON THE GEOMETRIC QUANTITIES FROM FEET BACK TO
C       INCHES.
      RN1=RN1*12.0D0
      RB1=RB1*12.0D0
      LA1=LA1*12.0D0
      RN2=RN2*12.0D0
      LA2=LA2*12.0D0
      RB2=RB2*12.0D0
      DELY=DELY*12.0D0
      DELZ=DELZ*12.0D0
C
C          CHANGING UNITS FROM RADIANS TO DEGREES TO PREPARE FOR NEW
C       CASE.
      ALST=ALST/0.01745329D0
C
C          SETTING THE OPTIONS EQUAL TO THE STORAGE INPUT VALUES TO
C       PREPARE FOR A NEW CASE.
```

```
      INALPH=INALPC
      NTFRST=NTHRTC
      LOPT=LOPTC
      IF(IREF.EQ.2) GO TO 30
 20   CALL HEADER(3)
C
C SAVE REENTRY VEHICLE TRAJECTORY DATA FOR POSSIBLE COMPARISON WITH THE
C TRAJECTORY OF A DECOY.
      DO 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) WR1S(I) = WR1P(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),WR1S(I),WR2S(I),WR3S(I),
     1 WL1S(I),WL2S(I),WL3S(I), I=1,LPLOT)
      LP=LP
      IF(LP.LT.6)GO TO 407
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,DVAR,OCCUR)
      RETURN
 30   CALL HEADER(NLOW)
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.
      DO 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) DD(I) = DS(I) -VOGPLT(I)
      IF(IOP(66).EQ.1) DB(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) = WR1S(I) -WR1P(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,
     1LPLOT)
      IF(IOP(74).EQ.1)WRITE(6,1001)(ZPLOT(I),TPLOT(I),WR1P(I),WR2P(I),
     1WR3P(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 RV OR DECOY.
      WRITE(6,406)  LP
      CALL SAVEDV(2,DVAR,OCCUR)
      WRITE(6,4666) (DV(I),DD(I),DB(I),I=1,LPLOT)
4666  FORMAT(1H0 8X7HDELTA VRX7HDELTA D3X7HDELTA B/(1H 1P3E15.7))
      GO TO 777
 507  CONTINUE
C
C THE SEQUENCE OF STATEMENTS ENDING AT 50 COMBINES THE H AND ZPLOT ARRAY
C THE ZPLOT 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 CORRIDOR 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 TTM 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)) 35,36,37
35    JB = J & 1
       TT(K) = H(J)
      CALL LINFIT(2,2,ZPLOT(I),TPLOT(I),WW,XX)
      TTM(K)=WW*H(J)&XX
      GO TO 44
36    JB = J  & 1
       TT(K) = ZPLOT(I)
       TTM(K)=TPLOT(I)
      GO TO 50
37     TT(K) = ZPLOT(I)
       TTM(K)=TPLOT(I)
      GO TO 50
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) = DB(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
      NDDCOY = 1
       IF(KK.EQ.NHIGH.AND.NDECOY.EQ.2) NDDCOY = 2
       IF(KK.EQ.NHIGH.AND.NDECOY.EQ.3) NDDCOY = 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
```

```
C DIFFERENCES VS THE VALUE OF THE DESIGN VARIABLE BEING VARIED
C THERE WILL BE THREE POINTS ON EACH GRAPH.
      IF(IOP(73).EQ.0) GO TO 600
      IF(IOP(64).EQ.1)
     1CALL  INTGRL(1,2,1,3915,3953,BCV,DV,TCV,SV,8H VELCCTY)
      IF(IOP(65).EQ.1)
     1CALL  INTGRL(2,2,4,3916,3954,BCD,DD,TCD,SD,8H DECELRN)
      IF(IOP(66).EQ.1)
     1CALL  INTGRL(3,3,7,3917,3955,BCB,DB,TCB,SB,8H BALLSTC)
600   IF(IOP(74).EQ.0) GO TO 444
      IF(IOP(67).EQ.1)
     1CALL  INTGRL(22,6,10,3941,3956,BCWL1,DWL1,TCWL1,SWL1,8H WAKE L1)
      IF(IOP(68).EQ.1)
     1CALL  INTGRL(23,6,13,3942,3957,BCWL2,DWL2,TCWL2,SWL2,8H WAKE L2)
      IF(IOP(69).EQ.1)
     1CALL  INTGRL(24,6,16,3943,3958,BCWL3,DWL3,TCWL3,SWL3,8H WAKE L3)
      IF(IOP(70).EQ.1)
     1CALL  INTGRL(25,6,19,3944,3959,BCWR1,DWR1,TCWR1,SWR1,8H WAKE R1)
      IF(IOP(71).EQ.1)
     1CALL  INTGRL(26,6,22,3945,3960,BCWR2,DWR2,TCWR2,SWR2,8H WAKE R2)
      IF(IOP(72).EQ.1)
     1CALL  INTGRL(27,6,25,3946,3961,BCWR3,DWR3,TCWR3,SWR3,8H WAKE R3)
444      CONTINUE
C
777   CONTINUE
C
C
C
      CALL SAVEDV(2,DVAR,DCCLR)
      RETURN
C
406   FORMAT(1H0 'CASE FAILED. TRY NEXT DESIGN VARIABLE' ,'LP =',I12)
700   FORMAT(1H0 'REENTRY VEHICLE CHARACTERISTICS')
900   FORMAT(1H0 'DECOY CHARACTERISTICS')
1000  FORMAT(
     11H07X'ALTITUDE'11X'TIME'7X'VELOCITY'3X'DECELERATION'11X'BETA'/
     2(1H F15.1,4F15.2))
1001  FORMAT(1H07X8HALTITUDE11X4HTIME8X7HWAKE R18X7HWAKE R28X7HWAKE R3
     2 8X7HWAKE L18X7HWAKE L28X7HWAKE L3/(1H F15.1,F15.2,1P3E15.7,
     3  0P3F15.2))
1800   FORMAT(1H1'INTEGRAL OF ',1X,A8,' =',1PE16.7)
1100   FORMAT(1H110X'CASE ',F9.3,5X'DATE ',F7.2,5X'MEMO ',F7.1,5X'2542E')
1052 FORMAT(1H0 10X,24H*****WARNING***** NGEOM=I3,23H IS THE SAME AS NG
     1EOM=1)
1053 FORMAT(1H0,10X,'*****WARNING***** LOPT=',I3,' IS NOT ALLOWED, LOPT
     1 HAS BEEN SET EQUAL TO 1')
1054 FORMAT(1H0,10X,'*****WARNING***** NTHRUST=',I3,' IS NOT ALLOWED,
     1NTHRUST HAS BEEN SET EQUAL TO 0')
1055 FORMAT(1H0 10X,35H*****WARNING***** MOPT=0 AND MHEAT=I3,23H IS THE
     1 SAME AS MHEAT=0)
1056 FORMAT(1H0 10X,36H*****WARNING***** MOPT=0 AND NCSECP=I3,24H IS THE
     1E SAME AS NOSECP=0)
1057 FORMAT(1H0 10X,37H*****WARNING***** MHEAT=0 AND NCSECP=I3,24H IS T
     1HE SAME AS NOSECP=0)
1077 FORMAT(1H0,10X,'******WARNING******'/ 11X,'INPUT ALPHA OPTION OVER
     1RIDES LOPT=',I3,' ROTATIONAL CALCULATIONS')
      END
```

```
CGIMAX
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE  GIMAX(AA,BB,NF,NMIMAX,ACCUR,NFUNC,XMIMAX,YMIMAX)
      IW=6
      DIMENSION E(100)
      COMMON /END/ ITERM
C
C
C          SUBROUTINE GIMAX, THE CONTROLLING SUBROUTINE FOR THE TWO
C       VARIABLE FIBONACCI SEARCH TECHNIQUE, OPTIMIZES THE SECONDARY
C        INDEPENDENT VARIABLE OF THE FUNCTION. AT EACH EVALUATION POINT
C        IN THIS OPTIMIZATION, GIMAX SUMMONS FUNCTION GMIMAX WHICH
C        CALLS FOR THE OPTIMIZATION OF THE PRIMARY VARIABLE.
C
C
      A=AA
      B=BB
      IF (NMIMAX) 10,11,11
   10 JMIMAX=1
      GO TO 20
   11 JMIMAX=2
   20 IF (B-A) 21,22,23
   21 TEMP=A
      A=B
      B=TEMP
      GO TO 23
   22 L=1
   25 WRITE (IW,26)  L
   26 FORMAT( 17H0  ERROR OF TYPE  I2)
      ITERM = 1
      GO TO 170
   23 R=B-A
      IF (NFUNC) 31,30,35
   30 IF (ACCUR) 31,31,40
   31 L=2
      GO TO 25
   35 NFUNC2=NFUNC&2
      R1=(1.0D0&SQRT (5.0D0))/2.0D0
      R2=(1.0D0-SQRT (5.0D0))/2.0D0
      ROACC=(R1**NFUNC2-R2**NFUNC2)/(R1-R2)
      GO TO 45
   40 ROACC=R/ACCUR
   45 IF (ROACC-2.0D0) 50,50,51
   50 XMIMAX=(A&B)/2.0D0
      YMIMAX=GMIMAX(XMIMAX,NF)
      GO TO 170
   51 IF (ROACC-3.0D0) 60,60,61
   60 XL=A&(B-A)/3.0D0
      XR=A&2.0D0*(B-A)/3.0D0
      YL=GMIMAX(XL,NF)
      IF( ITERM.NE.0) GO TO 160
      YR=GMIMAX(XR,NF)
      IF( ITERM.NE.0) GO TO 15[illegible]
      IF(YL-YR ) 70,70,71
   70 XMIMAX=XR
      YMIMAX=YR
      GO TO 170
   71 XMIMAX=XL
      YMIMAX=YL
```

```
      GO TO 170
   61 K=3
      F(1) = 1.0D0
      F(2)=1.0D0
   74 F(K)=F(K-1)&F(K-2)
      IF (ROACC-F(K)) 80,80,75
   75 IF (K-40) 77,76,76
   76 L=3
      GO TO 25
   77 K=K&1
      GO TO 74
   80 N=K
      XL=A&F(N-2)*R/F(N)
      XR=A&F(N-1)*R/F(N)
      YL=GMIMAX(XL,NF)
      IF(ITERM.NE.0) GO TO 160
      YR=GMIMAX(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      J=1
   90 NJ=N-J
      NJONE=N-J-1
      NJTWO=N-J-2
      GO TO (100,101), JMIMAX
  100 IF (YR-YL) 130,130,110
  101 IF (YR-YL) 110,110,130
  110 IF (J-N&3) 120,160,160
  120 B=XR
      A=A
      R=B-A
      XR=XL
      XL=A&F(NJTWO)*R/F(NJ)
      YR=YL
      YL=GMIMAX(XL,NF)
      IF(ITERM.NE.0) GO TO 160
  125 J=J&1
      GO TO 90
  130 IF (J-N&3) 140,150,150
  140 B=B
      A=XL
      R=B-A
      XL=XR
      XR=A&F(NJONE)*R/F(NJ)
      YL=YR
      YR=GMIMAX(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      GO TO 125
  150 XMIMAX=XR
      YMIMAX=YR
      GO TO 170
  160 XMIMAX=XL
      YMIMAX=YL
  170 RETURN
      END
```

```
CGMIMAX
      IMPLICIT REAL*8(A-H, O-Z)
      FUNCTION GMIMAX(X,NF)
      COMMON /END/ ITERM
      COMMON /XXSAVE/ X1SAVE(40), X2SAVE(40), KOUNT
      COMMON/MINSK/N,K,ALB(20),UB(20),XX(20),F1
      COMMON/IOPT/IPROC,IN,NCONS,IPNT,ICX,LIMIT,IRAND
      COMMON/EOPT/ERR,PRAND, FAC, DELTA
C
C
C          FUNCTION GMIMAX, PART OF THE TWO VARIABLE FIBONACCI SEARCH
C        TECHNIQUE, MINIMIZES THE PRIMARY VARIABLE FOR EACH STEP IN THE
C        OPTIMIZATION OF THE SECONDARY VARIABLE.
C
C
 2    XX(2) = X
      CALL MIMAX(ALB(1),UB(1),3,-1,ERR,LIMIT,XX(1),VAL)
      GMIMAX  = VAL
      KOUNT = KOUNT & 1
      IF(KOUNT.GT.40) ITERM = 1
      IF(KOUNT.GT.40) RETURN
      X1SAVE(KOUNT) = XX(1)
      X2SAVE(KOUNT) = XX(2)
 1000 RETURN
      END
```

```
CGRAM
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE GRAM
      COMMON/IOPT/IPROC,NMAX,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON/BLKO/HH(1600),X(40),D(40),E(40),DP(40),A(40),
     1 B(40),GS,GSP,GTP,GSS,GTT,GSB,FZZ,FP,FB,FC,Z,PZZ,TC,RS,SL,ZZZ,
     1OZZ,AXX,FL,
     2DFLXX,FAK,CXX(40,10)
      COMMON OCCUR,NOCCUR
      DIMENSION OCCUR(4000),NOCCUR(30),BB(20,20),C(20,20)
      EQUIVALENCE(HH(1),BB(1)),(HH(401),C(1))
C
C
C          SUBROUTINE GRAM USES THE GRAM-SCHMIDT ORTHOGONALIZATION
C        PROCESS.  GIVEN A MATRIX C AND A COLUMN VECTOR D, THIS ROUTINE
C        DEFINES A SET OF VECTORS, A, WHICH ARE USED TO CALCULATE AN
C        ORTHOGONAL SET OF VECTORS, B, TO BE USED AS THE ROW VECTORS OF
C        A MATRIX BB.  THE INPUT MATRIX C IS THEN REDEFINED AS THE
C        INVERSE (TRANSPOSE) OF THE MATRIX BB.
C
C
      DO 1 II = 1,NMAX
      A(II) = 0.D0
      B(II) = 0.D0
      DO 1 JJJ = 1,NMAX
      BB(II,JJJ) = 0.D0
    1 CONTINUE
C
C
      DO 107 I=1,NMAX
C
      DO 101 L=1,NMAX
      A(L) = 0.D0
      DO 101 N=1,NMAX
      A(L)=A(L)&C(L,N)*D(N)
  101 CONTINUE
C
      DO 102 L=1,NMAX
      IF(I-1) 111,111,112
  111 B(L)=A(L)
      GO TO 102
  112 IL=I-1
      DO 103 J=1,IL
      AB = 0.D0
      DO 104 K=1,NMAX
      AB=AB&A(K)*BB(J,K)
  104 CONTINUE
      B(L)=A(L)-AB*BB(J,L)
  103 CONTINUE
C
  102 CONTINUE
C
      BMAGS = 0.D0
      DO 106 L=1, NMAX
      BMAGS=BMAGS&B(L)**2
  106 CONTINUE
      BMAG=SQRT(BMAGS)
C
C
```

```
      DO 107 L=1,NMAX
      BB(I,L)=B(L)/BMAG
  107 CONTINUE
C
C
C        C MATRIX IS INVERSE OF BB MATRIX. THIS INVERSE IS THE TRANSPOSE
      DO 108 I=1,NMAX
      DO 108 J=1, NMAX
      C(J,I)=BB(I,J)
  108 CONTINUE
C
C
      WRITE(6,5)
    5 FORMAT(1H018HC MATRIX FROM GRAM)
C
      DO 3 I = 1,NMAX
    3 WRITE(6,2) (C(I,J),J=1,NMAX)
    2 FORMAT(1H 6(F20.10))
C
      RETURN
      END
```

```
CHEADER
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE HEADER(K)
      COMMON OCCUR, NOCCUR
      DIMENSION OCCUR(4000), NOCCUR(30)
      EQUIVALENCE(OCCUR(4000),LP)
C
C
C          SUBROUTINE HEADER PRINTS THE TYPE OF VEHICLE WHOSE TRAJECTORY
C       IS BEING PROCESSED ALONG WITH THE VALUES OF NINETEEN DESIGN
C       PARAMETERS, THE VALUE OF NGEOM, AND THE VALUE OF LP, AN ERROR
C       TEST PARAMETER.
C
C
      K = K
      GO TO(10, 20, 30), K
10    WRITE(6,700)
      GO TO 40
20    WRITE(6,800)
      GO TO 40
30    WRITE(6,900)
40    CONTINUE
      WRITE(6,1000)(OCCUR(I),I=133,145),(OCCUR(I),I=205,209),OCCUR(222),
     1NOCCUR(15),LP
 1000 FORMAT(1H0'            W1        THETA1            RN1           RB1       L
     1AMDA1           LA1            W2        THETA2            RN2           RB2'
     2/1H 10E12.2/1H
     *            '         LAMDA2          LA2         ZTURN           ZON
     3     ZOFF           THO          TON          TOFF           ISP  NGEOM
     4 LP'/1H 2E12.2,1PE12.4,0P6E12.2,2I6)
700   FORMAT(1H0 'BASIC DECOY CHARACTERISTICS')
800   FORMAT(1H0 'COMPARISON DECOY CHARACTERISTICS')
900   FORMAT(1H0 'REFERENCE REENTRY VEHICLE CHARACTERISTICS')
      RETURN
      END
```

```
CINFCOF
      SUBROUTINE INFCOF(DC,IZ,XO)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL *4 DC, XO
      COMMON OCCUR, NOCCUR
      COMMON /CPOCCUR/ POCCUR(11770)
      COMMON /CIOCCUR/ IOCCUR(320)
      DIMENSION DC(2),H(3,50),IZ(19),NOCCUR(30),OCCUR(4000)
       DIMENSION  XO(40)
      DIMENSION     S(9,160,3), T(160), Z(160)
      EQUIVALENCE (POCCUR(07101), H       )
      EQUIVALENCE (POCCUR(07251), S       )
      EQUIVALENCE (POCCUR(00001), T       )
      EQUIVALENCE (POCCUR(00161), Z       )
      EQUIVALENCE(IOCCUR(000309), II      )
      EQUIVALENCE(IOCCUR(000310), JJ      )
      EQUIVALENCE (IOCCUR(00305), NDECOY)
      EQUIVALENCE (IOCCUR(00302), LPLOT )
      EQUIVALENCE(IOCCUR(000312), NCON   )
C
C
C           SUBROUTINE INFCOF CALCULATES A TABLE OF INFLUENCE COEFFI-
C        CIENTS AND POSSIBLY SECOND DERIVATIVES AT EACH ALTITUDE FOR
C        WHICH VIXEN STORES TRAJECTORY INFORMATION.
C
C
      DO 27 I = 1, 19
      JP = I
      IF(IZ(I).EQ.NCON) GO TO 31
27    CONTINUE
C
      JP = 0
31    CONTINUE
C
      JP = JP & JP
      X21 = H(2,II)-H(1,II)
      IF(NDECOY.EQ.2) GO TO 100
      X32 = H(3,II)-H(1,II)
      X31 = (H(3,II)-H(2,II))/2.DO
      GO TO 500
 100  AMID= .5DO*(H(1,II)&H(2,II))
      IF(JP.EQ.0)GO TO 102
      WRITE(6,1000, XO(JP-1)  , XO(JP) ,(H(K,II),K=1,2), AMID
      GO TO 103
 102  WRITE(6,1050)NCON   ,(H(K,II),K=1,2) ,AMID
 103  WRITE(6,1100) DC
      WRITE(6,1200)
      DO 222 I = 1, LPLOT
      XA = S(JJ,I,2) -S(JJ,I,1)
      XA = XA/X21
170   WRITE(6,1300)Z(I),T(I),(S(JJ,I,K),K=1,2),XA
222   CONTINUE
      RETURN
C
C
 500  AMID = .5DO*(H(1,II)&H(2,II))
      BMID = .5DO*(H(1,II)&H(3,II))
      CMID = .5DO*(H(2,II)&H(3,II))
      IF(JP.EQ.0)GO TO 502
```

```
      WRITE(6,950) XQ(JP-1) , XQ(JP)
      GO TO 504
502   WRITE(6,1025) NCON
504   WRITE(6,1400)          (H(K,II),K=1,3),AMID,BMID,CMID
503   WRITE(6,1100) DC
            WRITE(6,1500)
C
C
      DO 555 I = 1, LPLOT
      XA= S(JJ,I,2)-S(JJ,I,1)
      XB= S(JJ,I,3) -S(JJ,I,1)
      XA = XA/X21
      XB = XB/X32
      XC = (XB-XA)/X31
530   WRITE(6,1600)Z(I),T(I),(S(JJ,I,K),K=1,3),XA,XB,XC
555   CONTINUE
      RETURN
C
C
717   FORMAT(16A4)
 950  FORMAT(1H 2A4,6HVARIES 32X 10HVALUES ARE 33X 14HMIDPOINTS ARE  )
1025  FORMAT(1H I8,' VARIES',32X'VALUES  ARE',33X'MIDPOINTS ARE')
 1050 FORMAT(1H0I8,' VARIES VALUES ARE',4X2F15.2,' MIDPOINT IS',F15.2)
1000  FORMAT(1H02A4,17HVARIES VALUES ARE 2F15.2,1X 11HMIDPOINT IS 1X ,
     1F15.2)
1100   FORMAT(1H 5HDELTA 2A4, 1X 5HTABLE)
1200  FORMAT(1H 7X'ALTITUDE',11X'TIME',4X'FIRST DELTA',3X'SECOND DELTA',
     14X'INFLU COEFF')
1300  FORMAT(1H F15.1,4F15.2)
1400  FORMAT (1H 30X6F15.2)
1500  FORMAT(1H 7X'ALTITUDE',11X'TIME',4X'FIRST DELTA',3X'SECOND DELTA',
     14X'THIRD DELTA FIRST INF COEF SECON INF COEF',5X' 2ND DERIV')
1600  FORMAT(1H F15.1,7F15.2)
      END
```

```
CINTERP
      SUBROUTINE INTERP(M,N,W,X,Y,Z)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(M), Y(M)
C
C
C        GIVEN THE TWO TABLES X AND Y(X) AND A PARTICULAR VALUE OF X
C      CALLED W.  SUBROUTINE INTERP FINDS Z EQUAL TO Y(W) BY LINEAR
C      INTERPOLATION.
C
C      M IS THE DIMENSION GIVEN X AND Y IN  THE CALLING PROGRAM.
C      X IS THE INDEPENDENT VARIABLE ARRAY.  IT CONTAINS N ELEMENTS.
C      Y IS THE DEPENDENT VARIABLE ARRAY.  IT CONTAINS N ELEMENTS.
C      W IS A GIVEN VALUE OF X FOR WHICH THE CORRESPONDING VALUE OF
C      Y IS TO BE FOUND.
C      Z IS THE VALUE OF Y FOUND TO CORRESPOND WITH W.
C
C      THIS VALUE IS FOUND BY LINEAR INTERPOLATION.
C
C       IF N IS LESS THAN 1 OR GREATER THAN M, AN ERROR MESSAGE IS
C      WRITTEN AND THE RUN IS TERMINATED.
C
C       THE ELEMENTS OF THE X ARRAY SHOULD BE EITHER MONOTONICALLY
C     INCREASING OR MONOTONICALLY DECREASING.
C
C      NO CHECK IS MADE FOR SUCH CONSISTENCY.
C
C       IF W IS OUTSIDE THE RANGE OF X, THE VALUE OF Z WILL BE FOUND
C      BY EXTRAPOLATION USING THE FIRST TWO OR THE LAST TWO POINTS
C      AS APPROPRIATE.
C
      IF(N.GT.0.AND.N.LE.M) GO TO 20
      WRITE(6,10) M, N
10    FORMAT(1H0 11HDIMENSION = I12, 1X 12HTABLE SIZE = I12)
      CALL EXIT
C
C
20    IF(X(1).LT.X(2)) GO TO 50
      DO 22 I = 2, N
      IF(W.GE.X(I)) GO TO 40
22    CONTINUE
C
30    I = N
40    Z = ((Y(I)-Y(I-1))/(X(I)-X(I-1)))*(W-X(I-1)) & Y(I-1)
C
      RETURN
C
C
50    DO 55 I = 2, N
      IF(W.LE.X(I)) GO TO 40
55    CONTINUE
      GO TO 30
C
      END
```

```
C INTGRL
      SUBROUTINE INTGRL(KA,KB,LX,KD,KE,Q,R,S,SIG,DC)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL * 4 DC, XQ
      COMMON OCCUR,NOCCUR
      COMMON /CPCCUR/ PCCUR(11770)
      COMMON /CICCUR/ IOCCUR(320)
      COMMON /TTMCOM/ TTM
      COMMON /INETST/ NDECOY
      DIMENSION OCCUR(4000), NOCCUR(30)
      DIMENSION A(27),B(40),C(40),D(160),E(160),F(40),G(40),H(40)
      DIMENSION IOP(90),IZ(15),Q(40),R(160),S(40),SIG(40),T(200)
      DIMENSION ZZ(200) , DC(2)
      DIMENSION TPLOT(160),TTM(200),XQ(40),XX(200),YY(200),Z(160)
      EQUIVALENCE (PCCUR(11571), A       )
      EQUIVALENCE (PCCUR(05841), H       )
      EQUIVALENCE (PCCUR(11758), HDIF    )
      EQUIVALENCE (PCCUR(05561), T       )
      EQUIVALENCE (PCCUR(00001), TPLOT )
      EQUIVALENCE (PCCUR(00161),  Z      )
      EQUIVALENCE (IOCCUR(00001), IOP    )
      EQUIVALENCE(IOCCUR(00310),JJ)
      EQUIVALENCE (IOCCUR(00313), KX     )
      EQUIVALENCE (IOCCUR(00302), LPLOT )
      EQUIVALENCE (IOCCUR(00304), NCP    )
      EQUIVALENCE (PCCUR(04961), B       )
      EQUIVALENCE (PCCUR(05001), C       )
      EQUIVALENCE (PCCUR(05041), D       )
      EQUIVALENCE (PCCUR(05201), E       )
      EQUIVALENCE (PCCUR(05361), F       )
      EQUIVALENCE (PCCUR(05401), G       )
      DATA IZ/133,134,135,136,137,138,139,140,141,142,143,144, 145,
     * 205,206,207,208,209,222/
      DATA XQ/160HW1        THETA1  RN1      RB1      LAMDA1  LA1      W2
     1 THETA2  RN2      RB2      LAMDA2  LA2      ZTURN   ZON     ZOFF
     2THO      TON      TOFF     ISP      ERROR   /
C
C
C         SUBROUTINE INTGRL PERFORMS ONE OF FOUR MAIN TASKS DEPENDING
C       UPON CERTAIN INPUT OPTIONS.  THESE ARE  (1) TO EVALUATE A
C       CORRIDOR INTEGRAL AND PRINT A TABULAR ARRAY OF DATA AT EACH
C       OF A NUMBER OF ALTITUDES.  (2)  TO EVALUATE A SIGMA SQUARE
C       INTEGRAL WHICH IS USED IN THE CALCULATIONS OF SUBROUTINE
C       EFFECT.  (3)  TO CALL THE PLOT ROUTINE TO OBTAIN A CORRIDOR
C       PLOT.  (4)  TO CALL THE PLOT ROUTINE TO OBTAIN AN INFLUENCE
C       COEFFICIENT PLOT.
C
C
      JJ = (LX&2)/3
      IF(IOP(KA&2*KB).EQ.0) GO TO 1120
      KQ = IOP(KA&4*KB)
      IF(KQ.EQ.1) WRITE(6,1000) DC
      JGO = 0
      IF(IOP(KA&3*KB).EQ.1) JGO = 30
      CALL LINEFIT(40,NCP,H,Q,B,C)
      CALL LINEFIT(40,NCP,H,S,F,G)
      CALL LINEFIT(160,LPLOT,Z,R,D,E)
      MX = 1
      ITEST = 0
```

```
      DA = 0.D0
      DB = 0.D0
      Q1 = B(1)*Z(1) & C(1)
      R1 = D(1)*Z(1) & E(1)
      S1 = F(1)*Z(1) & G(1)
      XX(1) = Q1
      YY(1) = R1
      ZZ(1) = S1
      IF(R1.GT.S1) NX = 2
      IF(R1.LT.Q1) NX = 0
      IF(KQ.EQ.0)GO TO 40
      IF(NX.NE.1) ITEST = 1
      IF(NX.NE.1) DP = Z(1)
      IF(NX-1) 10, 30, 20
 10   WRITE(6,1100) Z(1), TPLOT(1),Q1 , R1 , S1 , DA, Z(1)
      GO TO 40
 20   WRITE(6,1100) Z(1), TPLOT(1),Q1 , R1 , S1 , DA, Z(1)
      GO TO 40
 30   WRITE(6,1100) Z(1) , TPLOT(1), Q1, R1 , S1 , DA
C
 40   NS = LPLOT -1
      NR = NCP -1
C
C
      DO 1111 M = 2, KX
      DO 44 K = 2, NCP
      J = K -1
      IF(T(M).GE.H(K)) GO TO 50
 44   CONTINUE
      J = NR
 50   DO 55 K = 2, LPLOT
      I = K -1
      IF(T(M).GE.Z(K)) GO TO 60
 55   CONTINUE
      I = NS
 60   Q2 = B(J)*T(M) & C(J)
      R2 = D(I)*T(M) & E(I)
      S2 = F(J)*T(M) & G(J)
      XX(M) = Q2
      YY(M) = R2
      ZZ(M) = S2
      NY = 4
      IF(R2.GT.S2) NY = 7
      IF(R2.LT.Q2) NY = 1
      IGO = NX & NY
C
C
      GO TO(100,200,300,400,500,600,700,800,900),IGO
 100  T2 = 0.D0
      T1 = T(M-1)
      DA = DA & ADD(LX,T1,T(M),B(J),C(J),D(I),E(I),A)
      CALL WRITEM(JGO&10,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      GO TO 999
 200  T1 = T(M-1)
      T2 = -(C(J)-E(I))/(B(J)-D(I))
      DA = DA & ADD(LX,T2,T(M),B(J),C(J),D(I),E(I),A)
      CALL WRITEM(JGO&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      IF(ITEST.EQ.0) GO TO 950
      GO TO 999
 300  T2 = -(E(I)-G(J))/(D(I)-F(J))
```

```
      T1 = T(M-1)
      DA = DA & ADD(LX,T1,T2,D(I),F(I),F(J),G(J),A)
      CALL WRITEM(JGO&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      IF(ITEST.EQ.1) GO TO 350
      ITEST = 1
      DB = T2
350   T2 = -(C(J)-E(I))/(B(J)-D(I))
      DA = DA & ADD(LX,T2,T(M),B(J),C(J),D(I),E(I),A)
      CALL WRITEM(JGO&30,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      GO TO 999
400   T1 = T(M-1)
      T2 = -(C(J)-E(I))/(B(J)-D(I))
      DA = DA & ADD(LX,T1,T2,B(J),C(J),D(I),E(I),A)
      CALL WRITEM(JGO&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      IF(ITEST.EQ.0) GO TO 950
      GO TO 999
500   T1 = T(M-1)
      T2 = 0.D0
      CALL WRITEM(JGO&10,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      GO TO 999
600   T1 = T(M-1)
      T2 = -(E(I)-G(J))/(D(I)-F(J))
      DA = DA & ADD(LX,T1,T2  ,D(I),E(I),F(J),G(J),A)
      CALL WRITEM(JGO&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      IF(ITEST.EQ.0) GO TO 950
      GO TO 999
700   T2 = -(C(J)-E(I))/(B(J)-D(I))
      T1 = T(M-1)
      DA = DA & ADD(LX,T1,T2,B(J),C(J),D(I),E(I),A)
      CALL WRITEM(JGO&20,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      IF(ITEST.EQ.1) GO TO 750
      ITEST = 1
      DB = T2
750   T2 = -(E(I)-G(J))/(D(I)-F(J))
      DA = DA & ADD(LX,T2,T(M),D(I),E(I),F(J),G(J),A)
      CALL WRITEM(JGO&30,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      GO TO 999
800   T1 = T(M-1)
      T2 = -(E(I)-G(J))/(D(I)-F(J))
      DA = DA & ADD(LX,T2,T(M),D(I),E(I),F(J),G(J),A)
      CALL WRITEM(JGO&30,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      IF(ITEST.EQ.0) GO TO 950
      GO TO 999
900   T1 = T(M-1)
      T2 = 0.D0
      DA = DA & ADD(LX,T1,T(M),D(I),E(I),F(J),G(J),A)
      CALL WRITEM(JGO&10,T(M),TTM(M),Q2,R2,S2,DA,T2,D( I ),KQ)
      GO TO 999
950   ITEST = 1
      DB = T2
999    NX=NY/3
1111   CONTINUE
C
C
      OCCUR(KD) = DA
      OCCUR(KD&KB) = DB
      IF(KQ.EQ.0)  WRITE(6,1900) DC, DA, DB
C
1120  IF(IOP(KA&KB).EQ.0) GO TO 2300
      SUM = 0.D0
```

```
      RLO = (R(1)/SIG(1))**2
      DO 2222 I = 2, NCP
      CALL INTERP(160,LPLOT,H(I),Z,P,RHI)
      RHI = (RHI/SIG(I))**2
      SUM = SUM - (RLO & RHI)*(H(I) -H(I-1))*.500
      RLO = RHI
2222  CONTINUE
      OCCUR(KE) = SUM
      HDIF = H(1) -H(NCP)
      WRITE(6,1800) DC, SUM
C
2300  IF(IOP(KA).EQ.0) GO TO 4000
      CALL AVPLT(XO,IZ,XX,YY,ZZ,2,DC)
4000  IF(NDECOY.LT.2) RETURN
      IF(IOP(KA&5*KB).EQ.1) CALL INFCOF(DC,IZ,XO)
      IF(NDECOY.LT.3) RETURN
      IF(IOP(KA&6*KB).EQ.1) CALL AVPLT(XO,IZ,T,T,T,1,DC)
      RETURN
C
1000  FORMAT(1H0 6X 8HALTITUDE 10X 4HTIME 1X 13HLOWER CORRIDR 1X 5HRV-DE
     12A4, 1X 13HUPPER CORRIDR 1X 13HINTEGRL VALUE1X 13HLEAVE CORRIDR
     2 1X 13HENTER CORRIDR 5X 5HSLOPE)
1100  FORMAT(1H F14.1,6F14.2)
1200  FORMAT(1H 1P6E15.7,15X,E15.7)
1900  FORMAT(1H0 2A4, 1X 11HINTEGRAL = 1PE15.7,1X17HLEAVE CORRIDOR AT
     1 E15.7)
1900  FORMAT(1H0 13HINTEGRAL OF (2A4,10H/SIGMA)**2 1PE15.7)
      END
```

```
CJNXBES
          IMPLICIT REAL *8 (A-H, O-P )
          IMPLICIT REAL *8 (R-Z )
          IMPLICIT COMPLEX * 16 ( Q )
         SUBROUTINE JNXBES(ZR,ZI, NN, C, ZNR, ZNI )
            DIMENSION SR(2), SI(2) , TEMP(2)
C
C
C           SUBROUTINE JNXBES IS CALLED BY BESSEL TO CALCULATE THE BESSEL
C         FUNCTIONS OF MAXIMUM ORDER, N, AND OF N-1.
C
          QX= DCMPLX(ZR,ZI)
             Z=CDABS(QX)
          QXXI=(0.0D0,0.0D0)
          QXNU=(0.0D0,0.0D0)
C
                  IF( Z- 15.9999999999       )10,11,11
C
  11      QXI = (1.D0, 0.D0)
C FORMULA 9.2.5 HANDBOOK OF MATH FUNCTIONS-U.S. DEPT. COMMERCE
            QCC= DCMPLX(C, 0.D0)
           QX8= (8.D0,0.D0) * QX
         QPI =(3.141592653589793 , 0.D0 )
          QXNU=DFLOAT(NN)
          QCHI = QX -( (0.5D0,0.D0)* QXNU &(.25D0,0.D0) )* QPI
          QXMU = (4.D0,0.D0) * QXNU * QXNU
          QP = (1.D0,0.D0)
          QQ=(QXMU -(1.D0,0.D0)) / QX8
          QZ8 =QX8 ** 2
          QPT = QP
C
           DO  12  I=2,  4000 , 2
          QXXI=DFLOAT(I*(I-1))
          QPT = -QPT * ( QXMU - QXI * QXI) * (QXMU -( QXI &(2.D0,0.D0) )
      1**2 ) / QXXI  / QZ8
           QPNEW  = QP &  QPT
          RTPN =  CDABS(QPT/QPNEW)
           IF(  RTPN -C ) 13, 14, 14
   14     CONTINUE
           QP = QPNEW
   12     QXI = QXI  &(4.D0, 0.D0)
   13   CONTINUE
C
          QXI = (3.D0,0.D0)
          QQT = QQ
           DO  15  I=3,4000, 2
          QXXI=DFLOAT(I*(I-1))
          QQT = -QQT * (QXMU -QXI * QXI ) * (QXMU -(QXI &(2.D0,0.D0) )**2
      1 )/QXXI /QZ8
           QQNEW = QQ & QQT
            RQTON = CDABS( QQT /QQNEW )
            IF( RQTON - C ) 16, 17, 17
   17     CONTINUE
           QQ = QQNEW
   15     QXI = QXI & (4.D0,0.D0)
   16     CONTINUE
C
          QANS = CDSQRT( (2.D0,0.D0)/ QPI /QX ) * (QPNEW * CDCOS(QCHI)
      1-QQNEW * CDSIN(QCHI))
```

```
      CALL DRLIMI( QANS , TEMP )
      ZNR = TEMP(1)
      ZNI = TEMP(2)
      RETURN
C
 10   CONTINUE
      A=1.D0
      DO 1 I=1,NN
    1 A=DFLOAT(I) *A
      Z2R=ZR/2.D0
      Z2I=ZI/2.D0
      R=SQRT(Z2R**2&Z2I**2)
      THET=ATAN2(Z2I,Z2R)
      SUMR=1.D0
      SUMI=0.0D0
        ANG= DFLOAT(NN)           *THET
      RNNA=R**NN/A
      X2NR=COS(ANG)*RNNA
      X2NI=SIN(ANG)*RNNA
      DIV=1.D0
        SR(1) = 1.D65
        SR(2) = 1.D65
C
      DO 5 I=1,32000
      IF (MOD(I,2))2,3,2
    3 M=1
      GO TO 4
    2 M=2
    4 K=2*I
        AI=DFLOAT( K)* THET
      DIV=-(DIV*R/DFLOAT(I) )*R /DFLOAT(NN&I)
      SR(M)=DIV*COS(AI)
      SI(M)=DIV*SIN(AI)
      SUMR=SUMR&SR(M)
      SUMI=SUMI&SI(M)
      IF (ABS(SR(2)-SR(1))-C)6,5,5
    6 IF (ABS(SI(2)-SI(1))-C)7,5,5
    5 CONTINUE
C
      WRITE(6,1000)
 1000 FORMAT (52H  SERIES NOT CONVERGED IN 32000 ATTEMPTS FOR JNXBES.)
    7 ZNR=X2NR*SUMR-X2NI*SUMI
      ZNI=X2NR*SUMI&X2NI*SUMR
      RETURN
      END
```

```
CLINEFIT
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE LINEFIT(ND,NENTRY,X,Y,A,B)
      DIMENSION X(ND), Y(ND), A(ND), B(ND)
C
C
C         GIVEN THE NENTRY POINTS (X,Y), SUBROUTINE LINEFIT FINDS THE
C       EQUATION OF THE LINE PASSING THROUGH THE POINTS (X,Y) AND
C       ( X(I),Y(I) ) IN THE FORM Y=A(I)*X&B(I) FOR I=1,2,...,NENTRY-1.
C       THESE COEFFICIENTS CAN THEN BE USED BY THE CALLING PROGRAM
C       FOR MAKING A LINEAR FIT WHEN THE INTERVAL REQUIRED IS KNOWN
C       IN ADVANCE.
C
C
      N = NENTRY -1
      DO 77 J = 1, N
      A(J) = (Y(J&1)-Y(J))/(X(J&1)-X(J))
      B(J) = Y(J) -A(J)*X(J)
77    CONTINUE
1111  RETURN
      END
```

```
CLNTERP
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE LNTERP(TBATMZ,TABRHO,TABSND,UPBNDZ,DNBNDZ,Z,RHO,ASOUND)
C
         DIMENSION  TBATMZ(1),TABRHO(1),TABSND(1)
C
C          SUBROUTINE LNTERP DOES A LOGARITHMIC INTERPOLATION TO OBTAIN
C        FREE STREAM DENSITY AND A LINEAR INTERPOLATION TO OBTAIN FREE
C        STREAM SOUND SPEED FROM THE INPUT TABLES WHEN Z IS LESS THAN OR
C        EQUAL TO UPBNDZ AND GREATER THAN DNBNDZ.
C
C     TABLES MUST BE INPUT WITH LOWEST VALUE OF ALTITUDE FIRST
C     TABRHO IS IN LBM/FT**3   TABSND IS IN FT/SEC   TBATMZ IN FEET
C     IF Z IS LESS THAN UPBNDZ AND GREATER THAN DNBNDZ USE TABLES INPUT
C
      IF(Z-UPBNDZ)10,40,30
   10 IF(Z-DNBNDZ)30,30,40
   40 DO 45 I=1,50
      J=I
      IF(TBATMZ(I)-Z)45,46,46
   45 CONTINUE
      CALL EXIT
C
   46 EX=(Z-TBATMZ(J-1))/(TBATMZ(J)-TBATMZ(J-1))
      RHO=(TABRHO(J-1)*(TABRHO(J)/TABRHO(J-1))**EX)/32.174
      RATIO=(TABSND(J)-TABSND(J-1))/(TBATMZ(J)-TBATMZ(J-1))
      ASOUND=TABSND(J-1)&(Z-TBATMZ(J-1))*RATIO
C     RHO IS IN SLUG/FT**3   ASOUND IS IN FT/SEC
      GO TO 50
C
   30 CALL ARDFT2(Z,ASOUND,RHO)
   50 RETURN
      END
```

```
CMASSLO
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE MASSLO
C
      REAL*8 LAMDA
C
      DIMENSION OCCUR(4000),NOCCUR(30),TW(4,8),SDOT(4,8)
     1,QDOT(4,8)
C
      EQUIVALENCE
     1(OCCUR(032),LAMDA ),(OCCUR(068),SPD    ),(OCCUR(073),TWSTAG),
     2(OCCUR(091),Z      ),(OCCUR(092),ZTR    ),(OCCUR(115),C      ),
     3(OCCUR(2644),TW(1,1)  ),(OCCUR(2676),QDOT(1,1)),
     4(OCCUR(2740),SDOT(1,1))
      EQUIVALENCE
     1(NOCCUR(01),JJHOLD),(NOCCUR(02),KKHOLD)
C
      COMMON OCCUR,NOCCUR
C
C
C
C          SUBROUTINE MASSLO CONTROLS THE CALLING OF SUBROUTINE EVIL.
C        MASSLO CALLS IN EVIL FOR EACH BODY STATION FOR WHICH MASS LOSS
C        CALCULATIONS ARE REQUIRED.
C
C       J=1  STAG
C       J=2   LAM
C       J=3   TURB
C       J=4    SONIC
C
C          STAGNATION POINT CALCULATIONS.
C
      J=1
      K=1
      ASSIGN 2 TO NEXT
      QDOT(1,1)=QDOT(1,1)*C
      GO TO 100
    2 TWSTAG=TW(1,1)
      QDOT(1,1)=QDOT(1,1)/C
      IF(Z.LT.ZTR)GO TO 3
      IF(LAMDA.GT.1.0D-3)GO TO 4
C
C          LAMINAR FLOW, SHARP CONE AT MAXIMUM DIAMETER .
      J=2
      K=8
      ASSIGN 200 TO NEXT
      GO TO 100
C
C          LAMINAR FLOW, BLUNT CONE CALCULATIONS.
    4 J=2
      K=0
   20 K=K+1
      IF(K.EQ.9)GO  TO 200
      ASSIGN 20 TO NEXT
      GO TO 100
C
    3 IF(LAMDA.GT.1.0D-3)GO TO 5
C
C          TURBULENT FLOW, CALCULATIONS AT MAX. DIAMETER POINT OF A
C        SHARP CONE.
```

```
      J=2
      K=0
      ASSIGN 200 TO NEXT
      GO TO 100
C
C           TURBULENT FLOW, BLUNT CONE CALCULATIONS.
C           SONIC POINT TURBULENT CALCULATIONS.
    5 J=4
      K=1
      ASSIGN 6 TO NEXT
      GO TO 100
C           TANGENT POINT AND CONICAL FRUSTUM POINTS TURBULENT FLOW.
    6 J=3
      K=0
   30 K=K+1
      IF(K.EQ.9)GO TO 200
      ASSIGN 30 TO NEXT
      GO TO 100
  200 RETURN
C
C           CALLING IN SUBROUTINE EVIL TO PERFORM MASS LOSS RATE
C         CALCULATION.
  100 JJHOLD=J
      KKHOLD=K
      CALL EVIL(ZZ)
      TW(J,K)=ZZ
      SDOT(J,K)=SPD
C
      GO TO NEXT,(2,200,20,6,30)
      END
```

```
CMATM
         IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE MATMP (M,N,H,G,S)
      DIMENSION G(40),S(40)
      DIMENSION H(40,40)
C     MATRIX MULTIPLICATION
C
C
C          SUBROUTINE MATMP MULTIPLIES THE VECTOR G ON THE LEFT BY THE
C        MATRIX H TO PRODUCE A NEW VECTOR S.
C
C
  702 DO 703 I=1,M
  700 S(I)=0.0
      DO 703 J=1,N
  703 S(I)=H(J,I)*G(J)&S(I)
  704 RETURN
      END
```

```
CMATMPY
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE MATMPY(A,B,C)
      DIMENSION A(3),B(3,3),C(3)
C
C          SUBROUTINE MATMPY MULTIPLIES A THREE BY THREE MATRIX BY A
C        COLUMN MATRIX HAVING THREE COMPONENTS.
C
      DO 1 I=1,3
      A(I)=0.0D0
      DO 1 J=1,3
    1 A(I)=A(I)+B(I,J)*C(J)
      RETURN
      END
```

```
CMAXMIN
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE MAXMIN(NTABLE,NDIMEN,X,XMIN,XMAX)
      DIMENSION X(NDIMEN)
C
C         SUBROUTINE MAXMIN FINDS THE LARGEST AND SMALLEST VALUES IN
C       THE   ARRAY, WHICH CONTAINS NTABLE NUMBERS.  THE SMALLEST VALUE
C       IS PLACED IN LOCATION XMIN AND THE LARGEST IN LOCATION XMAX.
C
C
      IF(NDIMEN.GE.NTABLE) GO TO 20
      WRITE(6,10) NDIMEN, NTABLE
      CALL EXIT
20    XMIN = X(1)
      XMAX = X(1)
      IF(NTABLE.EQ.1) RETURN
C
      DO 77 J = 2, NTABLE
      IF(X(J).LT.XMIN) XMIN = X(J)
      IF(X(J).GT.XMAX) XMAX = X(J)
77    CONTINUE
C
1111  RETURN
10    FORMAT(1H112HDIMENSION IS I12, 1X13HTABLE SIZE IS I12/1H 4HEXIT)
      END
```

```
CMIMAX
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE MIMAX (AA,BB,NF,NMIMAX,ACCUR,NFUNC,XMIMAX,YMIMAX)
      IW=6
      DIMENSION E(100)
      COMMON /END/ ITERM
C
C
C        SUBROUTINE MIMAX UTILIZES A FIBCNACCI SEARCH TECHNIQUE TO
C     FIND THE MAXIMUM OR MINIMUM OF A ONE VARIABLE UNIMODAL
C     FUNCTION WITHIN A DEFINED REGION (AA,BB).
C
C
      A=AA
      B=BB
      IF (NMIMAX) 10,11,11
   10 JMIMAX=1
      GO TO 20
   11 JMIMAX=2
   20 IF (B-A) 21,22,23
   21 TEMP=A
      A=B
      B=TEMP
      GO TO 23
   22 L=1
C
   25 WRITE (IW,26)  L
   26 FORMAT(17H0  ERROR OF TYPE  I2)
      ITERM = 1
      GO TO 170
   23 R=B-A
      IF (NFUNC) 31,30,35
   30 IF (ACCUR) 31,31,40
   31 L=2
      GO TO 25
   35 NFUNC2=NFUNC&2
      R1=(1.0D0&SQRT (5.0D0))/2.0D0
      R2=(1.0D0-SQRT (5.0D0))/2.0D0
      ROACC=(R1**NFUNC2-R2**NFUNC2)/(R1-R2)
      GO TO 45
   40 ROACC=R/ACCUR
C
   45 IF (ROACC-2.0D0) 50,50,51
   50 XMIMAX=(A&B)/2.0D0
      YMIMAX=FMIMAX(XMIMAX,NF)
      GO TO 170
C
C
C
   51 IF (ROACC-3.0D0) 60,60,61
   60 XL=A&(B-A)/3.0D0
      XR=A&2.0D0*(B-A)/3.0D0
      YL=FMIMAX(XL,NF)
      IF(ITERM.NE.0) GO TO 160
      YR=FMIMAX(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      IF(YL-YR) 70,70,71
   70 XMIMAX=XR
      YMIMAX=YR
```

```
      GO TO 170
   71 XMIMAX=XL
      YMIMAX=YL
      GO TO 170
C
C
C
   61 K=3
      E(1) = 1.0D0
      E(2)=1.0D0
C
   74 E(K)=E(K-1)&E(K-2)
      IF (ROACC-E(K)) 80,80,75
   75 IF (K-40) 77,76,76
   76 L=3
      GO TO 25
   77 K=K&1
      GO TO 74
   80 N=K
      XL=A&E(N-2)*R/E(N)
      XR=A&E(N-1)*R/E(N)
      YL=FMIMAX(XL,NF)
      IF(ITERM.NE.0) GO TO 160
      YR=FMIMAX(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      J=1
C
   90 NJ=N-J
      NJONE=N-J-1
      NJTWO=N-J-2
      GO TO (100,101), JMIMAX
  100 IF (YR-YL) 130,130,110
  101 IF (YR-YL) 110,110,130
  110 IF (J-N&3) 120,160,160
  120 B=XR
      A=A
      R=B-A
      XR=XL
      XL=A&E(NJTWO)*R/E(NJ)
      YR=YL
      YL=FMIMAX(XL,NF)
      IF(ITERM.NE.0) GO TO 160
  125 J=J&1
      GO TO 90
C
  130 IF (J-N&3) 140,150,150
  140 B=B
      A=XL
      R=B-A
      XL=XR
      XR=A&E(NJONE)*R/E(NJ)
      YL=YR
      YR=FMIMAX(XR,NF)
      IF(ITERM.NE.0) GO TO 150
      GO TO 125
C
  150 XMIMAX=XR
      YMIMAX=YR
      GO TO 170
  160 XMIMAX=XL
```

```
      YMIMAX=YL                                                       MIMAX120
  170 RETURN                                                          MIMAX121
      END                                                             MIMAX122
```

```
CMISC-----------------------------------------------------------------------
      SUBROUTINE MISC(N,X,VAL)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LA1,LA2,LA1F,LAMDA1,LAMDA2,LAMD1F
      COMMON OCCUR, NOCCUR
      COMMON /IGDHL/ IGDL(20), IGDH(20)
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      COMMON /CPCCUR/ PCCUR(11770)
      DIMENSION ACOE(140)
      DIMENSION DIF(6),NOCCUR(30), OCCUR(4000),RATIO(8),X(20)
      DIMENSION GD(20)
      EQUIVALENCE (PCCUR(05881), ACOE  )
      EQUIVALENCE (OCCUR(03964), DELRCS)
      EQUIVALENCE(OCCUR(170),W1F),(OCCUR(150),THET1F),(OCCUR(169),RN1F),
     1(OCCUR(147),RB1F),(OCCUR(151),LAMD1F),(OCCUR(146),LA1F)
      EQUIVALENCE
     1(OCCUR(133),W1    ),(OCCUR(134),THETA1),(OCCUR(135),RN1   ),
     2(OCCUR(136),RB1   ),(OCCUR(137),LAMDA1),(OCCUR(138),LA1   ),
     3(OCCUR(139),W2    ),(OCCUR(140),THETA2),(OCCUR(141),RN2   ),
     4(OCCUR(142),RB2   ),(OCCUR(143),LAMDA2),(OCCUR(144),LA2   ),
     5(OCCUR(145),ZTURN)
      EQUIVALENCE (OCCUR(42),PI)
      EQUIVALENCE (OCCUR(3921), GD(1))
C
C-----------------------------------------------------------------------
C          AT PRESENT SUBROUTINE MISC CAN CALL SUBROUTINE POLCAL TO
C        CALCULATE THE FREE SPACE RADAR CROSS SECTION OF A VEHICLE.
C        HOWEVER, THE FIRST THREE ARGUMENTS OF POLCAL HAVE BEEN GENERAL-
C        IZED SO THAT ANY THREE ELEMENTS OF THE OCCUR ARRAY CAN BE USED.
C        IT CALCULATES THE ELEMENTS OF THE GD ARRAY OF GENERALIZED
C        DIFFERENCES BETWEEN UP TO TWENTY PAIRS OF ELEMENTS OF THE
C        OCCUR ARRAY.  MISC IS INTENDED, IN GENERAL, TO PERFORM ANY
C        TASKS WHICH DO NOT SEEM TO FIT READILY INTO ANY OF THE OTHER
C        SUBROUTINES.
C-----------------------------------------------------------------------
C
      IF(ICOM(7)*ICOM(8)*ICOM(9).NE.0) GO TO 30
      ANS = 0.D0
      GO TO 40
C
C-----------------------------------------------------------------------
 30   ID = ICOM(7)
      DV1 = OCCUR(ID)
      ID = ICOM(8)
      DV2 = OCCUR(ID)
      ID = ICOM(9)
      DV3 = OCCUR(ID)
      CALL POLCAL(DV1,DV2,DV3,ICOM(4),ICOM(5),ICOM(6),2,ACOE,ANS)
 40   DELRCS = XCOM(7)-ANS*XCOM(8)
C
C-----------------------------------------------------------------------
      DO 111 J = 1, 20
      IF(IGDH(J).EQ.0.OR.IGDL(J).EQ.0)  GO TO 112
      GD(J) = OCCUR(IGDH(J)) -OCCUR(IGDL(J))
111   CONTINUE
112   CONTINUE
C
C
C
```

```
      DO 11 I = 1, 8
      IF(I.GT.6) GO TO 11
      DIF(I) = 0.D0
11    RATIO(I) = 0.D0
      TR = .17453293D-1*THETA1
      R = RN1*DCOS(TR)
      H = RN1*(1.D0-DSIN(TR))
      V1=PI*(H*H*(RN1-H/3.D0)&(LA1-H)*(RB1*(RB1&R)&R*R)/3.D0)
       RATIO(7) = W1 / V1
      IF(ZTURN.LE.0.D0) GO TO 200
      TR = .17453293D-1*THETA2
      R = RN2*DCOS(TR)
      H = RN2*(1.D0-DSIN(TR))
      V2=PI*(H*H*(RN2-H/3.D0)&(LA2-H)*(RB2*(RB2&R)&R*R)/3.D0)
       RATIO(8) = W2 / V2
      DIF(1) = W2 -W1F
      DIF(2) = THETA2 -THET1F
      DIF(3) = RN2 -RN1F
      DIF(4) = RB2 -RB1F
      DIF(5) = LAMDA2 -LAMD1F
      DIF(6) = LA2 -LA1F
      RATIO(1) = W2/W1F
      RATIO(2) = THETA2/THET1F
      IF(RN1F.NE.0.D0) RATIO(3) = RN2/RN1F
      RATIO(4) = RB2/RB1F
      IF(LAMD1F.NE.0.D0) RATIO(5) = LAMDA2/LAMD1F
      RATIO(6) = LA2/LA1F
200   WRITE(6,1000) (RATIO(I),I=1,8),(DIF(I),I=1,6)
C
      DO 44 I = 1, 8
      IF(I.GT.6) GO TO 44
      OCCUR(I&3900) = DIF(I)
44    OCCUR(I&3906) = RATIO(I)
C
      RETURN
C
1000  FORMAT(1H0'MISC           W2/W1F          TH2/TH1F          RN2/RN1F
     1 RB2/RB1F          LM2/LM1F          LA2/LA1F          W1/V1           W2
     2/V2'/1H 4X 8F15.3/1H0'               W2-W1F          TH2-TH1F          R
     3N2-RN1F          RB2-RB1F          LM2-LM1F          LA2-LA1F'/1H 6F15.3)
      END
```

```
CNEUMAN                                                                  NEUMA
      IMPLICIT REAL*8(A-H,O-Z)                                           NEUMA
      SUBROUTINE NEUMAN(C,XR,XI,N,XL,ZZNR,                               NEUMA
     1ZZNI,XOR,XOI,X1R,X1I,IT)                                           NEUMA
      DIMENSION ZZNR(200,1),ZZNI(200,1)                                  NEUMA
C                                                                        NEUMA
C                                                                        NEUMA
C        SUBROUTINE NEUMAN CALCULATES FOR COMPLEX ARGUMENTS THE          NEUMA
C       NEUMANN FUNCTIONS OF INTEGRAL ORDER ZERO THROUGH TWENTY-FOUR.    NEUMA
C       FOR ARGUMENTS OF MAGNITUDE LESS THAN XL THE STANDARD POWER       NEUMA
C       SERIES IS USED TO CALCULATE THE NEUMANN FUNCTIONS OF ORDER ZERONEUMA
C       AND ONE.   NEUMAN CALLS IN NEUMN0 AND NEUMN1 TO CALCULATE THE   NEUMA
C       POWER SERIES FOR THE ZEROTH AND FIRST ORDER FUNCTIONS, RESPEC-  NEUMA
C       TIVELY.   FOR ARGUMENTS GREATER THAN XL AN ASYMPTOTIC SERIES ISNEUMA
C       USED FOR ORDERS ZERO AND ONE.   NEUMAN CALLS IN NEUMPO AND      NEUMA
C       NEUMQO TO PERFORM THE CALCULATIONS USED IN OBTAINING THE ZEROTHNEUMA
C       ORDER FUNCTIONS AND DIVMLT CALCULATES THE FIRST ORDER FUNCTIONSNEUMA
C       BY ASYMPTOTIC SERIES.   A FORMULA IS THEN USED TO GENERATE EACHNEUMA
C       INCREASING ORDER UNTIL THE NTH ORDER IS DETERMINED.              NEUMA
C                                                                        NEUMA
C                                                                        NEUMA
      D=DSQRT(XR*XR+XI*XI)                                               NEUMA
      IF (D)8,10,8                                                       NEUMA
   10 ZZNR(1,IT)=0.0D0                                                   NEUMA
      ZZNI(1,IT)=0.0D0                                                   NEUMA
      ZZNR(2,IT)=0.0D0                                                   NEUMA
      ZZNI(2,IT)=0.0D0                                                   NEUMA
      GO TO 11                                                           NEUMA
    8 CONTINUE                                                           NEUMA
C                                                                        NEUMA
      IF (D-XL)1,1,2                                                     NEUMA
C                                                                        NEUMA
C           NEUMANN FUNCTION CALCULATIONS FOR LARGE X                    NEUMA
    2 CALL NEUMPO(XR,XI,C,PNR,PNI)                                       NEUMA
      CALL NEUMQO(XR,XI,C,QNR,QNI)                                       NEUMA
      THET=ATANOR(XI,XR)                                                 NEUMA
      PI2RT=DSQRT(2.0D0/3.1415926D0)                                     NEUMA
      XRP=XR-0.7853982D0                                                 NEUMA
      SINRP=DSIN(XRP)                                                    NEUMA
      COSRP=DCOS(XRP)                                                    NEUMA
      E=DEXP(XI)                                                         NEUMA
      COSHH=(E+1.0D0/E)/2.0D0                                            NEUMA
      SINHH=(E-1.0D0/E)/2.0D0                                            NEUMA
      COSR=COSRP*COSHH                                                   NEUMA
      COSI=-SINRP*SINHH                                                  NEUMA
      SINR=SINRP*COSHH                                                   NEUMA
      SINI=COSRP*SINHH                                                   NEUMA
      OR=XR/D**2                                                         NEUMA
      OI=-XI/D**2                                                        NEUMA
C                                                                        NEUMA
      IF (XI)6,5,6                                                       NEUMA
    5 XOR=DSQRT(OR)                                                      NEUMA
      XOI=0.0D0                                                          NEUMA
      GO TO 7                                                            NEUMA
    6 THTO=ATANOR(OI,OR)                                                 NEUMA
      RO=DSQRT(DSQRT(OI*OI+OR*OR))                                       NEUMA
      XOR=-RO*DCOS(THTO/2.0D0)                                           NEUMA
      XOI=-RO*DSIN(THTO/2.0D0)                                           NEUMA
    7 ZNIRR=QNR*COSR-QNI*COSI+PNR*SINR-PNI*SINI                          NEUMA
      ZNINI=QNI*COSR+QNR*COSI+PNI*SINR+PNR*SINI                          NEUMA
```

```
      ZZNR(1,IT)=PI2RT*(X0R*ZNINR-X0I*ZNINI)
      ZZNI(1,IT)=PI2RT*(X0I*ZNINR+X0R*ZNINI)
      XUMR=ZZNR(1,IT)*X1R-ZZNI(1,IT)*X1I-2.0D0/3.1415926D0*OR
      XUMI=ZZNI(1,IT)*X1R+ZZNR(1,IT)*X1I-2.0D0/8.1415926D0*OI
      CALL DIVMLT(XUMR,XUMI,X0R,X0I,ZZNR(2,IT),ZZNI(2,IT))
      GO TO 3
C
C
C         NEUMANN FUNCTION CALCULATIONS FOR SMALL X
    1 CALL NEUMN0(XR,XI,X0R,X0I,ZZNR(1,IT),ZZNI(1,IT),C)
      CALL NEUMN1(XR,XI,X1R,X1I,ZZNR(2,IT),ZZNI(2,IT),C)
C
    3 B=XR**2+XI*XI
      XRB=XR/B
      XIB=-XI/B
   11 NN=N+1
      DO 4 I=3,NN
      XNN=DFLOAT(I-2)
      IM2=I-2
      IM=I-1
      ZZNR(I,IT)=2.0D0*XNN*(ZZNR(IM,IT)*XRB-ZZNI(IM,IT)*XIB)-ZZNR(IM2,
     1IT)
    4 ZZNI(I,IT)=2.0D0*XNN*(ZZNR(IM,IT)*XIB+ZZNI(IM,IT)*XRB)-ZZNI(IM2,
     1IT)
      RETURN
      END
```

```
CNEUMP
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE NEUMPO(XR,XI,C,PNR,PNI)
      DIMENSION SR(2),SI(2)
C
C           SUBROUTINE NEUMPO CALCULATES THE REAL AND IMAGINARY COMPO-
C        NENTS OF THE P-COMPONENT OF THE ZEROTH ORDER NEUMANN FUNCTION
C        N(P,Q).
C
      LTAP5=6
      D=XR**2&XI**2
      RR=1.D0/D
      RI=-2.D0*XR*XI/D**2
      SR(1)=1.0D0
      SI(1)=0.0D0
      SR(2)=1.D65
      PNR=0.0D0
      PNI=0.0D0
C
      DO 1 I=1,32000
      IF (MOD(I,2))3,2,3
    2 MM=1
      M=2
      GO TO 4
    3 MM=2
      M=1
C
    4 PNR=PNR&SR(M)
      PNI=PNI&SI(M)
      IF(DABS(SR(1)-SR(2))-C)5,5,6
5     IF(DABS(SI(1)-SI(2))-C)7,7,6
6     XB=DFLOAT(2*I)
      XT=2.D0*XB-1.D0
      XMULT=-1.D0/(XB-1.D0)*(XT-2.D0)**2/XB*XT*XT/64.D0
      SR(MM)=XMULT*(SR(M)*RR-SI(M)*RI)
    1 SI(MM)=XMULT*(SR(M)*RI&SI(M)*RR)
      WRITE (LTAP5,1000)
 1000 FORMAT (47H  SERIES NOT CONVERGED IN 32000 ATTEMPTS FOR PO)
C
    7 CONTINUE
      RETURN
      END
```

```
CNEUMQ
         IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE NEUMQO(XR,XI,C,QNR,QNI)
      DIMENSION SR(2),SI(2)
C
C           SUBROUTINE NEUMQO CALCULATES THE REAL AND IMAGINARY COMPO-
C        NENTS OF THE Q COMPONENT OF THE ZEROTH ORDER NEUMANN FUNCTION
C           N(P,Q).
C
      LTAP5=6
      D=XR**2&XI**2
      TNR=XR/D
      TNI=-XI/D
      RR=1.D0/D
      RI=-2.D0*XR*XI/D**2
      SR(2)=1.D65
      QNR=0.0D0
      QNI=0.0D0
      SR(1)=-.125D0*TNR
      SI(1)=-.125D0*TNI
C
      DO 1 I=1,32000
      IF (MOD(I,2))3,2,3
    2 MM=1
      M=2
      GO TO 4
    3 MM=2
      M=1
C
    4 QNR=QNR&SR(M)
      QNI=QNI&SI(M)
      IF(DABS(SR(1)-SR(2))-C)5,5,6
5     IF(DABS(SI(1)-SI(2))-C)7,7,6
6     XB=DFLOAT(2*I)
      XT=2.D0*XB&1.D0
      XMULT=-1.D0/XB*XT**2/(XB&1.D0)*(XT-2.D0)**2/64.D0
      SR(MM)=XMULT*(SR(M)*RR-SI(M)*RI)
    1 SI(MM)=XMULT*(SR(M)*RI&SI(M)*RR)
      WRITE (LTAP5,1000)
 1000 FORMAT (47H  SERIES NOT CONVERGED IN 32000 ATTEMPTS FOR QO)
C
    7 CONTINUE
      RETURN
      END
```

```
CNEUMO
      IMPLICIT REAL*8 (A-H,O-Z)
     SUBROUTINE NEUMNO(ZR,ZI,XR,XI,ZNR,ZNI,
    1C)
     DIMENSION SR(2),SI(2)
C
C        SUBROUTINE NEUMNO CALCULATES THE REAL AND IMAGINARY COMPO-
C      NENTS OF THE 0TH ORDER NEUMANN FUNCTION FROM THE STANDARD POWER
C      SERIES.
C
      LTAP5=6
         DATA  PI/3.141592653589793/
      R=DSQRT(ZR**2+ZI**2)/2.D0
      RLG=DLOG(R)
      THET=ATANOR(ZI,ZR)
      ZNR=(.57721566D0+RLG)*XR-THET*XI
      ZNI=(.57721566D0+RLG)*XI+THET*XR
      FACT=1.D0
      T=1.D0
      SR(1)=1.D65
      SI(1)=SR(1)
C
      DO 1 I=1,32000
      IF (MOD(I,2))2,3,2
    3 M=1
      GO TO 4
    2 M=2
    4 SUM=(-1.D0)**M*T
      XK=I
      XN=XK*THET*2.D0
      TEMP=R**XK
      SR(M)=SUM*(TEMP/FACT)**2*DCOS(XN)
      SI(M)=SUM*(TEMP/FACT)**2*DSIN(XN)
      ZNR=ZNR+SR(M)
      ZNI=ZNI+SI(M)
      IF(DABS(SR(2)-SR(1))-C)5,6,6
    5 IF(DABS(SI(2)-SI(1))-C)7,6,6
    6 XXI=DFLOAT(I+1)
      FACT=FACT*XXI
    1 T=T+1.D0/XXI
C
    7 ZNR=ZNR*2.D0/PI
      ZNI=ZNI*2.D0/PI
      RETURN
      END
```

```
CNEUMN1
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE NEUMN1(ZR,ZI,XR,XI,ZNR,ZNI,
     1C)
      DIMENSION SR(2),SI(2)
C
C        SUBROUTINE NEUMN1 CALCULATES THE REAL AND IMAGINARY COMPO-
C      NENTS OF THE 1ST ORDER NEUMANN FUNCTION FROM THE STANDARD POWER
C      SERIES.
C
         DATA  PI/3.141592653589793/
      R=DSQRT(ZR**2+ZI**2)/2.DO
      RLG=DLOG(R)
      THET=ATANOR(ZI,ZR)
      ZNR=2.0D0*(.5772156600D0+RLG)*XR-2.0D0*THET*XI
      ZNI=2.0D0*(.5772156600D0+RLG)*XI+2.0D0*THET*XR
      B=ZR**2+ZI**2
      XRB=ZR/B
      XIB=ZI/(-B)
      ZNR=ZNR-XRB*2.DO
      ZNI=ZNI-XIB*2.DO
      FACT2=2.DO
      FACT=1.D0
      SR(1)=1.D65
      SI(1)=SR(1)
      SERS=1.0D0
C
      DO 1 I=1,32000
      IF (MOD(I,2))2,3,2
    3 M=1
      GO TO 4
    2 M=2
    4 P1=DFLOAT(I)
      SUM=-(-1.DO)**M/FACT/P1*SERS
      XK=DFLOAT(2*I-1)
      XN=XK*THET
      SR(M)=SUM*R**XK*DCOS(XN)/FACT
      SI(M)=SUM*R**XK*DSIN(XN)/FACT
      ZNR=ZNR+SR(M)
      ZNI=ZNI+SI(M)
      IF (DABS(SR(2)-SR(1))-C)5,6,6
    5 IF (DABS(SI(2)-SI(1))-C)7,6,6
    6 FACT=FACT*P1
    1 SERS=SERS+1.DO/P1+1.DO/(P1+1.DO)
C
    7 ZNR=ZNR/PI
      ZNI=ZNI/PI
      RETURN
      END
```

```
CNOSEBL
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE NOSEBL
C
      REAL*8 LAMDA
C
      DIMENSION SDOT(4,8),OCCUR(4000),NOCCUR(30)
C
      EQUIVALENCE
     1(OCCUR(008),COST  ),(OCCUR(032),LAMDA ),(OCCUR(059),RNDOT ),
     2(OCCUR(060),RBDOT ),(OCCUR(064),SINT  ),
     3(OCCUR(090),XBAR  ),(OCCUR(091),Z     ),(OCCUR(092),ZTR   ),
     4(OCCUR(237),XUP   ),(OCCUR(2740),SDOT(1,1))
C
      COMMON OCCUR,NOCCUR
C
C
C
C          SUBROUTINE NOSEBL OBTAINS DERIVATIVES OF NOSE RADIUS AND BASE
C        RADIUS WITH TIME IN ORDER TO CALCULATE SHAPE CHANGE FOR
C        CONSTANT CONE HALF ANGLE IN CONTINUUM FLOW.  IN RAREFIED FLOW,
C        THE DERIVATIVES ARE SET EQUAL TO ZERO.
C
C          RNDOT IS CALCULATED FROM STAG. POINT RECESSION RATE.
    1 IF(XBAR.GT.XUP)GO TO 2
      RNDOT=SDOT(1,1)*SINT/(1.D0-SINT)
C
C          BLUNTNESS RATIO LESS THAN OR = 1.0D-3 IS TREATED AS A SHARP
C        CONE AND SHARP CONE SIDEWALL RECESSION RATES ARE USED.
C        BLUNTNESS RATIO GREATER THAN 1.0D-3 USES BLUNT CONE SIDEWALL
C        RECESSION RATES.
      IF(LAMDA.GT.1.0D-3)GO TO 3
C
C          SHARP CONE SIDEWALL RECESSION RATE FACTORS.
C          TURBULENT
      IF(Z.LT.ZTR)TEM=0.3706D0*SDOT(3,8)
C          LAMINAR
      IF(Z.GE.ZTR)TEM=0.7071D0*SDOT(2,8)
      GO TO 4
C
C          BLUNT CONE SIDEWALL RECESSION RATE FACTORS
C          TURBULENT
    3 IF(Z.LT.ZTR)TEM=SDOT(3,7)
C          LAMINAR
      IF(Z.GE.ZTR)TEM=SDOT(2,7)
C
C          RATE OF CHANGE OF THE BASE
    4 RBDOT=-TEM/COST
      RETURN
C
C          DERIVATIVES ARE ZERO IN NON-CONTINUUM FLOW
    2 RNDOT=0.D0
      RBDOT=0.D0
      RETURN
      END
```

```
CPLT
      SUBROUTINE PLT
C
C
C        SUBROUTINE PLT IS A DUMMY SUBROUTINE INCLUDED SO THAT CALLS
C       TO THE AEROSPACE PLOTTER CAN BE SIMULATED.
      RETURN
      END
```

```
CPOLCAL
      SUBROUTINE POLCAL(X,Y,Z,NAX,NAY,NAZ,ITP,ACOE,ANS)
      IMPLICIT  REAL*8 (A-H, O-Z)
      DIMENSION ACOE(140)
C
C
C          SUBROUTINE POLCAL EVALUATES THE POLYNOMIAL DEFINED BY THE
C         TRIPLE SUMMATION ( I FROM 1 TO NAX, J FROM 1 TO NAY, K FROM 1
C         TO NAZ ) OF THE EXPRESSION  -  ( A(NAX*NAY*(K-1)&NAX*(J-1)&I)
C         *X**(I-1) *Y**(J-1) *Z**(K-1)   ).
C
C
      M=0
      ANS=0.0D0
      IF(ITP-1) 20,20,10
   10 NAZU=NAZ
      GO TO 30
   20 NAZU=1
   30 DO 130 K=1,NAZU
      DO 130 J=1,NAY
      DO 130 I=1,NAX
      M=M&1
      IF(I-1) 40,40,50
   40 A=1.0D0
      GO TO 60
   50 A=X**(I-1)
   60 IF(J-1) 70,70,80
   70 B=1.0D0
      GO TO 90
   80 B=Y**(J-1)
   90 IF(K-1) 100,100,110
  100 C=1.0D0
      GO TO 120
  110 C=Z**(K-1)
C
  120 ANS=ANS & ACOE(M) * A * B * C
C
  130 CONTINUE
C
      RETURN
      END
```

```
CPRELI
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE PRELIM(LP)
C
      REAL*8 LAMDA,LA,MUINF,MINF,ME,M,MX,MY,MZ
      REAL*8 MINF2,MINF3,MLE,MUW
C
      DIMENSION  A(514),TXCGD(50),TABZ(50),TW(4,8),TBMAT(3),
     1           OCCUR(4000),NOCCUR(30),THDELZ(25),THTHO(25),THDELT(25)
      DIMENSION  TABRHO(50),TABSND(50),TBATMZ(50)
C
      EQUIVALENCE
     1(OCCUR(001),AREF   ),(OCCUR(008),COST   ),(OCCUR(009),CODRAG),
     2(OCCUR(010),CAPL   ),(OCCUR(011),COSLAM),(OCCUR(012),CNALP ),
     3(OCCUR(013),CMALP  ),(OCCUR(017),CPF    ),(OCCUR(018),CPW   ),
     4(OCCUR(020),CMO    ),(OCCUR(021),D      ),(OCCUR(027),G     )
      EQUIVALENCE
     1(OCCUR(028),GAMMA  ),(OCCUR(029),HSRT)  ),(OCCUR(031),HS    ),
     2(OCCUR(032),LAMDA  ),(OCCUR(033),LA     ),(OCCUR(034),MUINF ),
     3(OCCUR(035),MINF   ),(OCCUR(036),ME     ),(OCCUR(037),M     ),
     4(OCCUR(042),PT     ),(OCCUR(046),PINFPS),(OCCUR(047),PS    ),
     5(OCCUR(048),PE     ),(OCCUR(049),PINF  ),(OCCUR(051),QD    ),
     6(OCCUR(052),RN     ),(OCCUR(053),RB     ),(OCCUR(054),RESINT)
      EQUIVALENCE
     1(OCCUR(055),RHOINF),(OCCUR(056),RHOINL),(OCCUR(057),R     ),
     2(OCCUR(061),RHOE   ),(OCCUR(062),REYL   ),(OCCUR(064),SINT  ),
     3(OCCUR(066),SOCOST),(OCCUR(067),SINTM ),(OCCUR(069),THETAL),
     4(OCCUR(070),TANT   ),(OCCUR(072),TINF   ),(OCCUR(076),THETA ),
     5(OCCUR(079),TE     ),(OCCUR(080),TIMER )
      EQUIVALENCE
     1(OCCUR(082),V      ),(OCCUR(083),VE     ),(OCCUR(084),W     ),
     2(OCCUR(085),W.     ),(OCCUR(088),XCPO   ),(OCCUR(089),XBARST),
     3(OCCUR(090),XBAR   ),(OCCUR(091),Z      ),(OCCUR(092),ZTR   ),
     4(OCCUR(094),COPLAM),(OCCUR(097),DELW   ),(OCCUR(126),XBAR1 ),
     5(OCCUR(196),CMOIN ),(OCCUR(198),S      ),(OCCUR(201),TH    )
      EQUIVALENCE
     1(OCCUR(204),WTH    ),(OCCUR(205),ZON    ),(OCCUR(206),ZOFF  ),
     2(OCCUR(207),THO    ),(OCCUR(208),TON    ),(OCCUR(209),TOFF  ),
     3(OCCUR(210),MX     ),(OCCUR(211),MY     ),(OCCUR(212),MZ    ),
     4(OCCUR(213),THINF ),(OCCUR(214),AE     ),(OCCUR(215),CTHZET),
     5(OCCUR(216),CPSZET),(OCCUR(217),STHZET),(OCCUR(218),SPSZET),
     6(OCCUR(219),DELY   ),(OCCUR(220),DELZ   ),(OCCUR(225),HWBAR)
      EQUIVALENCE
     1 (OCCUR(226),DELW2 ),(OCCUR(227),DELW3 ),(OCCUR(228),WTOTAL),
     2(OCCUR(243),TRZTR ),(OCCUR(244),REYINF),(OCCUR(245),PTOTAL),
     3(OCCUR(247),UPANDZ),(OCCUR(248),DNANDZ)
      EQUIVALENCE
     1(OCCUR(0301),     A(1)),(OCCUR(0844) ,TXCGD(1)),
     2(OCCUR(0994),  TABZ(1)),(OCCUR(2644),TW(1,1) ),
     3(OCCUR(3568), THTHO(1)),(OCCUR(3593),THDELZ(1)),
     4(OCCUR(3618), THDELT(1)),(OCCUR(3643), TBMAT(1))
      EQUIVALENCE
     1(OCCUR(3721), TBATMZ(1)),(OCCUR(3771) , TABRHO(1)),
     2(OCCUR(3821), TABSND(1))
      EQUIVALENCE
     1(NOCCUR(04),MAXTAB),(NOCCUR(07),LOPT  ),(NOCCUR(08),IATMOS),
     2(NOCCUR(09),IKCMO ),(NOCCUR(14),NPRINT),(NOCCUR(22),NTHRST),
     3(NOCCUR(23),ITHRST)
C
```

```
      COMMON OCCUR,NOCCUR
C
C
C
C          SUBROUTINE PRELIM DOES PRELIMINARY CALCULATIONS OF GEOMETRIC,
C      FLOW FIELD, AND THRUSTING PARAMETERS.
C
C
C          GEOMETRIC PARAMETERS
C       CAPL=SHARP CONE SLANT LENGTH, LAMDA=BLUNTNESS RATIO, COSLAM=
C       COSINE OF THETA*LAMDA, LA=AXIAL LENGTH FROM STAGNATION POINT,
C       AREF=AREA OF THE BASE, D=BASE DIAMETER, S=SURFACE DISTANCE
C       FROM THE NOSE OF BLUNT CONE
C
      TEMTEM=32.2D0*2.27D-8
      CAPL=RB*RESINT
      LAMDA=RN/RB
      COSLAM=COST*LAMDA
      LA=CAPL*(COST-LAMDA*(1.D0-SINT))
      AREF=PI*RB*RB
      D=2.D0*RB
      S=RN*(0.5D0*PI-THETA)&RB*RESINT-RN/TANT
C
C           CALCULATIONS FOR AERO. COEFF. CN ALPHA,CMQ,CM ALPHA
C           XCGD AND XCPD ARE THE DISTANCES OF CENTER OF GRAVITY AND
C        CENTER OF PRESSURE FROM THE NOSE NON-DIMENSIONALIZED BY THE
C         BASE DIAMETER.
      CNALP=1.92*SQCOST*(1.D0-0.5D0*COSLAM**2)
      LL=-1
      XCG=D*TABLE(Z,TABZ,TXCGD,MAXTAB,LL)
      CMALP=-(2.D0*(1.D0-COSLAM**3)-3.D0*SQCOST*COSLAM*(1.D0-COSLAM**2))
     1/(3.D0*TANT)
      CMALP=CMALP-(RN-XCG)*SQCOST*(1.D0-0.5*COSLAM*COSLAM)/RB
      CMALP=CMALP&LAMDA*SINT*SQCOST*(1.D0-COSLAM*COSLAM)
      CMALP=0.96*CMALP
      IF(CMALP.GT.0.D0)ILP=6
      IF(CMALP.GT.0.D0)WRITE(6,1000)
 1000 FORMAT(1H0,10X,20HCMALP GREATER THAN 0)
      IF(ILP.EQ.6)RETURN
      XCPD=-CMALP/CNALP
      XCPD=XCPD&TABLE(Z,TABZ,TXCGD,MAXTAB,LL)
      TEM=COSLAM*SQCOST
      IF(IKCMQ.GT.0)GO TO 43
C
C          CALCULATED CMQ BEING USED
      CMQ=-TEM*(TEM*(1.D0-COSLAM**2)-4.D0*(1.D0-COSLAM**3)/3.D0)/(SINT**
     12)
      CMQ=CMQ-0.5D0*(1.D0-COSLAM**4)/(SINT*SINT)
      CMQ=CMQ-0.5D0*SQCOST*(COSLAM*(RN-XCG)/RB)**2
      TEM=XCG-RN*(1.D0-SINT)
      CMQ=CMQ&2.D0*TEM*(2.D0*(1.D0-COSLAM**3)-3.D0*SQCOST*COSLAM*(1.D0-
     1COSLAM*COSLAM))/(3.D0*RB*TANT)
      CMQ=CMQ-TEM*TEM*SQCOST*(1.D0-COSLAM**2)/(RB*RB)
      CMQ=0.96D0*CMQ
      GO TO 44
C
C          INPUT CMQ BEING USED
   43 CMQ=CMQIN
   44 CONTINUE
C
```

```
C          IF LOPT=4 THE FREE STREAM FLOW PARAMETERS ARE DETERMINED
C        FROM THE WIND TUNNEL INPUTS.
      IF(LOPT.EQ.4)GO TO 79
C
      IF(IATMOS.GT.0)GO TO 47
C
C          USING BUILT IN 1962 STANDARD ATMOSPHERE
      CALL ARFOT2(Z,ASOUND,RHO)
      GO TO 48
C
C          USING INPUT ATMOSPHERE FOR SPECIFIED RANGE OF ALTITUDE AND
C        THE 1962 STANDARD ATMOSPHERE OUTSIDE THIS RANGE.
   47 CONTINUE
      CALL INTERP(TBATMZ,TABRHO,TABSND,LPBNDZ,DNRNDZ,Z,RHO,ASOUND)
   48 CONTINUE
C
C          CALCULATION OF FREE STREAM CONDITIONS
C        REYLA=FREE STREAM REYNOLDS NUMBER BASED ON LA.  TINF IS IN
C        DEGREES RANKINE,  MUINF IS IN LBM/(FT-SEC),  RHOINF IS IN
C        SLUGS/FT**3,  RHOIN1 IS IN LBM/FT**3,  PINF IS IN LB/FT**2.
C        ALL LENGTHS ARE IN FEET
      AINF=ASOUND
      RHOINF=RHO
      RHOIN1=RHOINF*G
      CDDRAG=CODLAM/RHOIN1
      TINF=AINF*AINF*4.16D-4
      MUINF= TEMTEM*TINF**1.5D0/(TINF&198.6D0)
      REYLA=RHOIN1*V*LA/MUINF
      MINF=V/AINF
      MINF2=MINF*MINF
      MINF3=MINF2*MINF
      PINF=RHOIN1*R*TINF
      TINE=TINER
      GO TO 89
C
C          CALCULATIONS OF FREE STREAM FLOW PROPERTIES FROM WIND TUNNEL
C        CONDITIONS INPUT.
C        REYINF IS THE REYNOLDS NUMBER PER INCH.  REYFT IS THE
C        REYNOLDS NUMBER PER FOOT.  PTOTAL IS IN LB/FT**2.
   79 CONTINUE
      MINF2=MINF*MINF
      MINF3=MINF2*MINF
      REYFT=REYINF*12.0
      PINF=PTOTAL/((1.0D0&(GAMMA-1.0D0)*MINF2/2.0D0)**(GAMMA/(GAMMA-1.0D0
     1D0)))
      TTK1=PINF*MINF/(2.0356D-02*R*TEMTEM)
      TTK2=198.6D0*TTK1
      T2TK1=TTK1*TTK1
      TTFACT=T2TK1&4.0D0*REYFT*TTK2
      IF(TTFACT.LT.0.0D0)GO TO 59
      TINF=(TTK1&DSQRT(TTFACT))/(2.0D0*REYFT)
      GO TO 57
   59 WRITE(6,1059)
 1059 FORMAT(10X,28HNEGATIVE SQUARE ROOT IN TINF)
      LP = 6
      RETURN
   57 CONTINUE
      RHOIN1=PINF/(R*TINF)
      RHOINF=RHOIN1/G
      AINF=DSQRT(TINF)/2.0356D-2
```

```
      MUINF=32.200*2.270-8*TINF**1.500/(TINF&198.600)
      V=MINF*AINF
C
   89 IF(LOPT.GE.3)GO TO 16
      IF(NTHRST.EQ.0) GO TO 16
C
C          THRUSTING OPTION CALCULATIONS
      GO TO (11,12),NTHRST
C
C          THRUST IS READ IN AS A FUNCTION OF ALTITUDE
   11 IF((Z-ZON).GT.1.D-4)GO TO 16
      IF(Z.LE.ZOFF)GO TO 13
      IF((Z-ZON).GE.0.0D0)KKTHR=0
      IF(KKTHR.EQ.0.AND.NPRINT.NE.0)WRITE(6,2222)Z
 2222 FORMAT(1H0,25X,'***THRUST TURNED ON AT Z=',E15.6,'***'/15X,'THIS M
     1ESSAGE IS VALID ONLY WHEN IT IS IMMEDIATELY BEFORE PRINTOUT'/)
      JJTHR=0
      KKTHR=KKTHR&1
      DELTAZ=ZON-Z
      LL=-1
      THINF=THO*TABLE(DELTAZ,THDELZ,THTHO,ITHRST,LL)
      GO TO 14
C
C          THRUST IS READ IN AS A FUNCTION OF TIME
   12 IF((TIME-TON).LT.1.D-4)GO TO 16
      IF(TIME.GE.TOFF)GO TO 13
      IF((TON-TIME).GE.0.0D0)KKTHR=0
      IF(KKTHR.EQ.0.AND.NPRINT.NE.0)WRITE(6,2223)TIME
 2223 FORMAT(1H0,25X,'***THRUST TURNED ON AT TIME=',F12.4,'***'/15X,'THI
     1S MESSAGE IS VALID ONLY WHEN IT IS IMMEDIATELY BEFORE PRINTOUT'/)
      JJTHR=0
      KKTHR=KKTHR&1
      DELTAT=TIME-TON
      LL=-1
      THINF=THO*TABLE(DELTAT,THDELT,THTHO,ITHRST,LL)
   14 TH=THINF-AE*PINF
      TBMAT(1)=TH*CTHZET*CPSZET
      TBMAT(2)=TH*CTHZET*SPSZET
      TBMAT(3)=-TH*STHZET
      RX=XCG-LA
      MX=DELY*TBMAT(3)-DELZ*TBMAT(2)
      MY=DELZ*TBMAT(1)-RX*TBMAT(3)
      MZ=RX*TBMAT(2)-DELY*TBMAT(1)
      GO TO 10
C
C          THRUST PARAMETERS ARE ZEROED OUT WHEN NOT BEING USED
   13 IF(JJTHR.NE.0)GO TO 16
      WRITE(6,2224)TIME,Z
 2224 FORMAT(1H0,25X,'***THRUST SHUT OFF AT TIME=',F12.4,' AND Z=',E15.6
     1,'***'/15X,'THIS MESSAGE IS VALID ONLY WHEN IT IS IMMEDIATELY BEFO
     2RE PRINTOUT'/)
      JJTHR=1
   16 MX = 0.0D0
      MY=0.0D0
      MZ=0.0D0
      TBMAT(1)=0.0D0
      TBMAT(2)=0.0D0
      TBMAT(3)=0.0D0
      TH=0.0D0
      THINF=0.0D0
```

```
      KKTHR=0
   10 CONTINUE
C
C          CALCULATIONS FOR SHARP CONE EDGE PROPERTIES, STAGNATION
C        ENTHALPY AND PRESSURE
      HSPTO=(0.24D0*TINF&20.D-6*V*V)/33.86D0
      HS=32.96D0*HSPTO
      PINFPS=(2.D0/((GAMMA&1.D0)*MINF2       ))**(GAMMA/(GAMMA-1.D0))
      PINFPS=PINFPS*(((2.D0*GAMMA*MINF2      -GAMMA&1.D0)/(GAMMA&1.D0))**(1
     X.D0/(
     1GAMMA-1.D0))
      PS=PINF/PINFPS
      QP=0.5D0*RHOINF*V*V
      REYL =RHOINF*V*CAPL/MUINF
      SINTM=MINF*SINT
      VE=V*SQRT(1.D0-1.40D0*SINTM**1.90D/(MINF2))
      PE=PINF*(1.D0&2.8D0*SINTM*SINTM*(2.5D0&60.D0*SINTM)/(1.D0&16.D0*SIN
     XTM))
      IF(SINTM.GE.5.70D)GO TO 1
      TETINF=1.D0&SINTM*(0.C966D0&0.22670D*SINTM)
      GO TO 2
    1 TETINF=0.D0
      DO 3 K=1,3
      KK=K-1
      ISUB1=5D&5*KK
      PTEM=(PINF/2116.2D0)**KK
      DO 3 I=1,5
      II=I-1
      ISUB=ISUB1&II
    3 TETINF=TETINF&A(ISUB)*(SINTM**II)*PTEM
    2 TE=TETINF*TINF
      RHOE=PE/(R*TE)
      ME=VE*MINF/(V*SQRT(TETINF))
      MUE= TEM1C**TE**1.5DC/(TE&198.6D0)
      REYES=RHOE*VE*S/MUE
C
      IF(TRZTR.EQ.0.D0)GO TO 46
C          TRANSITION ALTITUDE IS INPUT
      ZTR=TRZTR
      GO TO 45
C
C          TRANSITION ALTITUDE IS CALCULATED
   46 CONTINUE
      ZTR=0.D0
      DO 8 J=1,2
      JJ=J-1
      ISUB1=30D&JJ
      TEMJ=CAPL**JJ
      DO 8 I=1,2
      II=I-1
      ISUB2=ISUB1&2*II
      TEMI=THETAD**II
      IF(THETAD.GT.15.D0)TEMI=(15.D0)**II
      DO 8 K=1,2
      KK=K-1
      ISUB=ISUB2&6*KK
      TEMC=A(ISUB)*TEMJ*TEMI
      TEML=LAMDA
      IF(LAMDA.GT.0.3D0)TEML=0.3D0
      IF(KK.NE.0)TEMC=TEMC*TEML**KK
```

```
    8 ZTR=ZTR&TEMO
   45 CONTINUE
C
      IF(Z.GE.ZTR)J=2
      IF(Z.LT.ZTR)J=3
C
C         CALCULATION OF CONSTANT PRESSURE SPECIFIC HEAT AT THE EDGE OF
C           SHARP CONE BOUNDARY LAYER
      IF(TE.GT.5000.D0)CPE=A(111)&A(112)*TE
      IF(TW(J,8).GT.5000.D0)CPW=A(111)&A(112)*TW(J,8)
      IF(TE.LT.700.D0)CPE=.2398D0
      IF(TW(J,8).LT.700.D0)CPW=0.2398D0
      IF(TE.GT.5000.D0)GO TO 4
      IF(TE.LT.700.D0)GO TO 4
      CPE=0.D0
      DO 5 I=1,6
      II=I-1
    5 CPE=CPE&A(II&105)*TE**II
C
C         CALCULATION OF CONSTANT PRESSURE SPECIFIC HEAT AT WALL
    4 IF(TW(J,8).GT.5000.D0)GO TO 6
      IF(TW(J,8).LT.700.D0)GO TO 6
      CPW=0.D0
      DO 7 I=1,6
      II=I-1
    7 CPW=CPW&A(II&105)*TW(J,8)**II
    6 CONTINUE
C
      HWBAR=CPW*TW(J,8)/33.86D0
      MUW=TEMTEM*TW(J,8)**1.5D0/(TW(J,8)&198.6D0)
C
C         CALCULATION OF XBAR, VISCOUS INTERACTION PARAMETER, USED IN
C           DETERMINING START OF CONTINUUM FLOW REGIME
      CE=MUW*TE/(MUE*TW(J,8))
      XBAR=ME*ME*ME*SQRT(CE/REYES)
C
C         CALCULATION OF RAREFACTION PARAMETER USED IN DETERMINING END
C           OF FREE MOLECULE AND THE START OF STRONG INTERACTION REGIME
      CINF=MUW*TINF/(MUINF*TW(J,8))
      XBARST=MINF*SQRT(CINF/(REYL*S/CAPL))/(SINT*SINT)
      XBAR1=XBARST*SINT*SINT
C
      IF(LOPT.EQ.3)GO TO 91
C
C         DELW2=WEIGHT LOSS DUE TO ABLATION, DELW3=WEIGHT LOSS DUE TO
C           THRUSTING, DELW=TOTAL WEIGHT LOSS, WTOTAL=TOTAL WEIGHT OF
C           VEHICLE
      DELW2=WO-W
      DELW3=WO-WTH
      DELW=DELW2&DELW3
      WTOTAL=WO-DELW
      IF(WTOTAL.GT.0.D0)GO TO 91
      WRITE(6,1091)WTOTAL
 1091 FORMAT(1H0,10X,'*****WARNING***** WTOTAL =',E11.4,' IS NOT ACCEPTA
     1BLE' )
      LP=6
C
   91 CONTINUE
      M=WTOTAL/G
      RETURN
```

NOT REPRODUCIBLE

```
CRANDOM
      FUNCTION  RANDOM(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
C          SUBROUTINE RANDOM IS A DUMMY SUBROUTINE.
C
C
      RANDOM = X
      RETURN
      END
```

```
CRCSEC
C
      SUBROUTINE RCSEC(B1,ZNET,WAKEL,SIGP,NOPT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON OCCUR,NOCCUR
      COMMON /CRCSEC/ F,TAU,SIGMOS,PHI,XBZ,ZME,HH,DDW,C1,C2,C3,C4,C5
      COMMON /DRCSEC/ BZERO,B2,B3,BTWEN,B24,DX,ZNUS,CNE,DSB,X2BOD,X3B,
     1IND2,NSTWL
      COMMON /CSIG/ XTT,WS1,XODP,DWST,SPH2,XLMS,XMSP,XSSP,CB1,ZCR2,ZCR4,
     1XB20,DWBSTL,SPH4,B0,CODW,WS4,CIGMOS,INDO
      DIMENSION OCCUR(4000),NOCCUR(30)
      EQUIVALENCE (OCCUR(4000),LP)
C
C
C          SUBROUTINE RCSEC COMPUTES THE RADAR CROSS SECTION AND PULSE
C       SHAPES FROM THE GIVEN OBSERVING RADAR AND THE TRANSITION POINT
C       ELECTRON DENSITY OF THE VEHICLE AND ALSO COMPUTES WAKE LENGTH.
C
C
C
C        IND2=0 - NO OUTPUT GENERATED BY THIS ROUTINE
C        IND2=1 - INTERMEDIATE STEPS PRINTED OUT
C        ALTITUDE INDEPENDENT QUANTITIES
C          SCALING CONSTANTS - BZERO,B2,B3,BTWEN,B24
C          SCALING CONSTANTS - XB0,B2,B3,XB20,B24
C          FREQUENCY - F
C          PULSE LENGTH - TAU
C          SEA LEVEL COLLISION FREQ. - ZNUS
C          PRESET CONSTANTS - C1,C2,C3,C4,C5
C          NOISE LEVEL TO WHICH WAKE LENGTH IS MEASURED - SIGMOS
C          STEP IN AXIAL COORDINATE - DX
C          MAXIMUM NUMBER OF STEPS USED TO COMPUTE WAKE LENGTH - NSTWL
C        ALTITUDE DEPENDENT QUANTITIES
C          ALTITUDE - HH
C          MACH NUMBER(UPSTREAM) - ZME
C          LOOK ANGLE - PHI
C          WAKE DIAMETER = BASE DIAM. - DDW
C          SCALE HEIGHT - XBZ
C          TRANSITION POINT ELECTRON DENSITY - ZNET
C        2 BODY OVERDENSE LENGTH - X2BOD
C        STATION WHERE LINEAR PRODUCTION TERMS FIRST DOMINATE OVER
C          NON-LINEAR PRODUCTION TERMS - X3B
C        TRANSITION ELECTRON DENSITY WHEN NONLINEAR PRODUCTION TERMS
C          ARE CONSIDERED IN TURBULENT WAKE - CNE
C        ADDITIONAL RCS DUE TO CONSIDERATION ON NONLINEAR PRODUCTION
C          TERMS IN TURBULENT WAKE - DSB
      CB1 = B1
      CODW = DDW
      CIGMOS = SIGMOS
      XB20 = BTWEN
      INDO = IND2
  200 F2 = F* F
      ZK = 6.2831853080C69 * F2
      WC1 =  F2/1.0D8
      ZNS2= ZNUS * ZNUS
C
C        RO - ATMOSPHERIC DENSITY AS A FUNCTION OF ALTITUDE
      RO = DEXP(-HH/XBZ)
C
```

```
C        ZNCR - ELECTRON DENSITY AT WHICH THE PLASMA BECOMES OVERDENSE
      ZNCR = WC1*((1.0D0 + RO * RO * ZNS2/F2)**C5)
C
C        XTT - TRANSITION PT. ONSET OF OVERDENSE REGION
      XTT = C1/RO * (1.0D0 -C2/(ZNE ** C3)) + C4
      ZNNCR= ZNET/ZNCR
      WS4 = ZNNCR ** XB20
      WCR2 = CNE * CNE/(ZNCR * ZNCR)
      IF(PHI)605,604,605
  604 PHI = 0.01D0
  605 IF(PHI -90.0D0)790,790,791
  791 PHI = 180.0D0 - PHI
C
C        CONVERT PHI  TO RADIANS AND FIND ITS SINE AND COSINE
  790 PH = PHI * 1.745329250-2
        SPH = DSIN(PH)
        CPH = DCOS(PH)
      SPH2 = SPH * SPH
      SPH4  = SPH2 * SPH2
C
C        TEST TO PREVENT OVERFLOW (COS 90 = 0)
      IF(PHI-90.0D0)606,607,606
  607 WS1 = 5000.0D0
      GO TO 608
  606 WS1 = 150.0D0 * TAU/CPH
  608 ZCR2 = ZNNCR * ZNNCR
      ZCR4 = ZCR2* ZCR2
      WS7 = B1 * XB20
C
C        XOD - ONSET OF CONSTANT MULTIPLE SCATTERING REGION
      XOD = X2BOD
      IF(X2BOD-X38)53,51,51
   51 XOD= DLOG(ZCR2)/B1
   53 IF(XOD)1,2,2
    1  CLOD = 0.0D0
C
C        COMPUTE XODP WHICH WILL REPLACE XOD IN ALL EQUATIONS -
C          XODP = XOD OR 0.0 IF XOD IS NEGATIVE
      XODP = 0.0D0
      GO TO 3
    2 CLOD = DMIN1(WS1,XOD)
      XODP = XOD
C
C        XSS - PT. OF ONSET OF SINGLE SCATTERING REGION
    3 XSS = DLOG(ZCR2/SPH4)/B1
      IF(INO2-1)30000,10000,30000
10000 WRITE (6,20000) RO,XTT,ZNCR,ZNNCR,XOD
20000 FORMAT(18X,7H    RO=,1PE15.7,6H   XT=,1PE15.7,6H NECR=,1PE15.7,10H
     1 NET/NECR=,1PE15.7,6H  XOD=,1PE15.7)
30000 IF(XSS)4,5,5
    4 CLSS = 0.0D0
C
C        COMPUTE XSSP WHICH WILL REPLACE XSS IN ALL EQUATIONS -
C          XSSP = XSS OR 0.0 IF XSS IS NEGATIVE
      XSSP = 0.0D0
      GO TO 6
    5 CLSS = DMIN1(WS1,XSS)
      XSSP = XSS
    6 CONTINUE
      IF(INO2-1)30001,10001,30001
```

```
10001 WRITE (6,20001) CLOD,XODP,XSS,CLSS,XSSP                           RCSEC120
20001 FORMAT(18X,7H  CLOD=,1PE15.7,6H XODP=,1PE15.7,6H  XSS=,1PE15.7,10HRCSEC121
     1      CLSS=,1PE15.7,6H XSSP=,1PE15.7)                             RCSEC122
30001  DDW2 = DDW * DDW                                                 RCSEC123
       DDW2 = DDW * DDW                                                 RCSEC124
       DDW4 = DDW2 * DDW2                                               RCSEC125
      XXXX = (ZK * B24 * DDW)**3                                        RCSEC126
C                                                                       RCSEC127
C        DWST - SURFACE SCATTERING CORRECTION TERM FOR BASE DIAMETER    RCSEC128
      DWST =(XXXX/(2.0D0 * SPH2 & XXXX)) * DDW                          RCSEC129
      DEN = (1.0D0 &B2 *F2/1.0D8 * DDW2)**B3                            RCSEC130
      B0 =(BZERO* (F2/1.0D8)**2 * DDW4)/DEN                             RCSEC131
      DWB1= DDW * SPH2/B1                                               RCSEC132
      DWBST1 = DWST * SPH2/B1                                           RCSEC133
       IF(XB20)800,801,800                                              RCSEC134
C                                                                       RCSEC135
C        XMS - PT. OF ONSET OF VARIABLE MULTIPLE SCATTERING REGION      RCSEC136
  801 XMS = XODP                                                        RCSEC137
      GO TO 802                                                         RCSEC138
  800 CONTINUE                                                          RCSEC139
      XMS = 2.0D0 * DLOG(B0 * WS4 * SPH**(2.0D0*(1.0D0- XB20))*2.0D0)/  RCSEC140
     1 WS7                                                              RCSEC141
  802  IF(XMS-XODP)760,761,761                                          RCSEC142
  760  XMSP = XODP                                                      RCSEC143
      GO TO 762                                                         RCSEC144
  761 XMSP = XMS                                                        RCSEC145
  762 XLMOD = DMAX1(XMSP,XODP)                                          RCSEC146
       IF(IND2-1)30002,10002,30002                                      RCSEC147
10002 WRITE (6,20002) XXXX,DWST,DEN,B0,XMS                              RCSEC148
20002 FORMAT(18X,7H  XXXX=,1PE15.7,6H DWST=,1PE15.7,6H  DEN=,1PE15.7,10HRCSEC149
     1       B0=,1PE15.7,6H  XMS=,1PE15.7)                              RCSEC150
30002  XLMSD = DMIN1(XLMOD,WS1)                                         RCSEC151
      XLMS = 0.0D0                                                      RCSEC152
      IF(XMSP-XODP)742,742,743                                          RCSEC153
  743 XLMS = (XMSP-XODP)/2.0D0                                          RCSEC154
      XLMS = DMIN1(0.5D0 * (WS1-XODP),XLMS)                             RCSEC155
      XLMS = DMAX1(0.0D0,XLMS)                                          RCSEC156
  742  CONTINUE                                                         RCSEC157
      WS6 = X3B/WS1                                                     RCSEC158
      E1 = DEXP(-B1 * CLOD)                                             RCSEC159
      E2 = DEXP(-B1 * CLSS)                                             RCSEC160
      E3 = DEXP(-B1 * WS1)                                              RCSEC161
      SPH16 = SPH2/16.0D0                                               RCSEC162
      TRM1 = CLOD * DWST * SPH2-& DWST/B1*ZCR2*((E1-E3)*SPH16           RCSEC163
     1& B0 *(E2-E3)) & XLMS * DWST * SPH2                               RCSEC164
       IF(IND2-1)30003,10003,30003                                      RCSEC165
10003 WRITE (6,20003) XLMSD,XLMS,TRM1,E1,E2                             RCSEC166
20003 FORMAT(18X,7H XLMSD=,1PE15.7,6H XLMS=,1PE15.7,6H TRM1=,1PE15.7,10HRCSEC167
     1        E1=,1PE15.7,6H   E2=,1PE15.7)                             RCSEC168
30003  CONTINUE                                                         RCSEC169
      E1 = DEXP(-4.0D0 * B1 * CL3D) - DEXP(-4.0D0*WS1 * B1)             RCSEC170
C                                                                       RCSEC171
C        THE FOLLOWING TEST IS USED TO PREVENT UNDERFLOW                RCSEC172
       IF(DABS(E1) - 1.0D-6)9005,9005,9006                              RCSEC173
 9005  E1 = 0.0D0                                                       RCSEC174
 9006 CONTINUE                                                          RCSEC175
       IF(XB20)780,781,780                                              RCSEC176
  780 E2 = DEXP(-0.5D0 * WS7 * XLMSD) - DEXP(-WS7 * CLSS/2.0D0)         RCSEC177
C                                                                       RCSEC178
C        SIGP - PEAK WAKE RADAR CROSS SECTION                           RCSEC179
```

```
      SIGP = TRM1 & DWST * SPH2 * ZCR4*0.25D0/B1*E1*ZCR4 &2.0D0*B0*SPH2
     1 **(2.0D0 - XB20)/WS7 * ZNNCR**XB20 * E2 * DWST
      GO TO 782
  781 SIGP = TRM1 & 0.25D0*DWST*SPH2/B1*ZCR4*E1*ZCR4&B0*SPH4
     1 * (CLSS - XLMSD)       * DDW
  782 CONTINUE
      IF(IND2-1)30004,10004,30004
10004 WRITE (6,20004) SIGP
20004 FORMAT(18X,7H  SIGP=,1PE15.7 //)
C
C       THE FOLLOWING SET OF STATEMENTS COMPUTE THE WAKE LENGTH(WAKEL)
30004 WAKEL = 0.0D0
C
C       NOPT = 0, COMPUTE SIGP ONLY
C       NOPT = 1, COMPUTE SIGP AND WAKEL
C
      IF(NOPT)65,900,65
   65 J = 0
      N1 = NSTWL &1
      IT = 1
      NF = 1
C
  740 XJ = DFLOAT(J)
      XX = XJ * DX  &XTT
       SIG = FUN1(XX,NF)
      IF(J-N1)750,750,751
C
  751 WRITE (6,20005)
20005 FORMAT(2X,69HWARNING-NUMBER OF STEPS REQUIRED TO COMPUTE WAKE LENG
     1TH EXCEEDS NSTWL /)
       LP = 6
     RETURN
C
C
  750 GO TO (731,732,733),IT
  731 XL1 = XX
      IT = 2
  734 J = J&1
      XN = XX
      SIGN = SIG
      GO TO 740
C
  732 IT = 3
      GO TO 734
C
C
C
  733 IF(SIG-SIGN)736,736,735
  735  XL1 = XN
       GO TO 734
  736 XH1 = XX
C
C
     CALL FIB1(XL1,XH1,1,1,1.0D-4,20,XPEAK,SPEAK)
C
C
      IF(SPEAK-SIGMDS)737,737,738
  737 WAKEL = 0.0D0
     RETURN
C
```

```
C                                                                       RCSEC240
  738  XN = XPEAK                                                       RCSEC241
       J = 1                                                            RCSEC242
  741  XJ = DFLOAT(J)                                                   RCSEC243
       XX = XPEAK & DX * XJ                                             RCSEC244
       SIG = FUN1(XX,NF)                                                RCSEC245
       IF(SIG - SIGMUS)739,739,7422                                     RCSEC246
 7422  J = J&1                                                          RCSEC247
       IF(J-N1)752,752,751                                              RCSEC248
  752  XN = XX                                                          RCSEC249
       GO TO 741                                                        RCSEC250
C                                                                       RCSEC251
  739   CALL FIB1(XN,XX,2,-1,1.0D-4,20,XNL,DUM)                         RCSEC252
C                                                                       RCSEC253
       WAKEL = XNL - XPEAK                                              RCSEC254
C                                                                       RCSEC255
  900  RETURN                                                           RCSEC256
       END                                                              RCSEC257
```

```
CREADIT
        IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE READIT(ILL111,IMPLOT)
C
C
C
C         SUBROUTINE READIT READS IN THE INPUT FOR ONE CASE AND TESTS
C           CERTAIN VALUES IN THE IOP ARRAY.
C
C
      REAL*8 LAMDA1,LAMDA2,LA1,LA2
      REAL*8  ISP,MW,NSL1,NSL2,NST1,NST2,NGL1,NGL2,NGT1,NGT2
C
      COMMON OCCUR,NOCCUR
      COMMON /CICCUR/ IOCCUR(320)
      DIMENSION IOP(90)
      DIMENSION ALPTAB(75),OCCUR(4000),NOCCUR(30),NPLOT(5),
     1          CHIGH(16),CDOWN(16),THTHO(25),THDELZ(25),THDELT(25),
     2          TXCGD1(50),TXCGD2(50),TABI1(50),TABI2(50),TABIX1(50),
     3           TABZ1(50),TABZ2(50),HTAB(75),WHTAB(75),COTAB(75),
     4           TABIX2(50),WCDTAB(75)
      DIMENSION  WTZ(75),WTMINF(75),WTKINF(75),WTPTOT(75),TRAJT(75),
     1           TRAJZ(75),TRAJV(75),TRAJW(75),TRAJRN(75),TRJALP(75)
      DIMENSION TABRHO(50),TABSND(50),TBATMZ(50)
      DIMENSION  A(514),B(21)
C
      EQUIVALENCE (OCCUR(00301), A     )
      EQUIVALENCE (OCCUR(00214), AE    )
      EQUIVALENCE (OCCUR(03646), ALPTAB)
      EQUIVALENCE (OCCUR(00122), ALST  )
      EQUIVALENCE (OCCUR(00188), AWREF )
      EQUIVALENCE (OCCUR(00823), B     )
      EQUIVALENCE (OCCUR(00152), BETA11)
      EQUIVALENCE (OCCUR(00171), BETA12)
      EQUIVALENCE (OCCUR(00153), BETA21)
      EQUIVALENCE (OCCUR(00172), BETA22)
      EQUIVALENCE (OCCUR(00154), BETA31)
      EQUIVALENCE (OCCUR(00173), BETA32)
      EQUIVALENCE (OCCUR(00155), BETA41)
      EQUIVALENCE (OCCUR(00174), BETA42)
      EQUIVALENCE (OCCUR(00115), C     )
      EQUIVALENCE (OCCUR(00019), CAPG  )
      EQUIVALENCE (OCCUR(00128), CASE  )
      EQUIVALENCE (OCCUR(03549), CDOWN )
      EQUIVALENCE (OCCUR(03363), COTAB )
      EQUIVALENCE (OCCUR(03533), CHIGH )
      EQUIVALENCE (OCCUR(00124), CHUIN1)
      EQUIVALENCE (OCCUR(00125), CHUIN2)
      EQUIVALENCE (OCCUR(00161), CPG1  )
      EQUIVALENCE (OCCUR(00180), CPG2  )
      EQUIVALENCE (OCCUR(00160), CP21  )
      EQUIVALENCE (OCCUR(00179), CP22  )
      EQUIVALENCE (OCCUR(00127), DATE  )
      EQUIVALENCE (OCCUR(00166), DELHC1)
      EQUIVALENCE (OCCUR(00185), DELHC2)
      EQUIVALENCE (OCCUR(00187), DELIN )
      EQUIVALENCE (OCCUR(00159), DELRH1)
      EQUIVALENCE (OCCUR(00178), DELRH2)
      EQUIVALENCE (OCCUR(00129), DELY  )
```

```
      EQUIVALENCE (OCCUR(00220), DELZ  )                                READI 60
      EQUIVALENCE (OCCUR(00248), DNBNDZ)                                READI 61
      EQUIVALENCE (OCCUR(00129),  EMO  )                                READI 62
      EQUIVALENCE (OCCUR(167), EPSIL1)                                  READI 63
      EQUIVALENCE (OCCUR(00186), EPSIL2)                                READI 64
      EQUIVALENCE (OCCUR(00157), F1    )                                READI 65
      EQUIVALENCE (OCCUR(00176), F2    )                                READI 66
      EQUIVALENCE (OCCUR(00027), G     )                                READI 67
      EQUIVALENCE (OCCUR(00105), GAMF0 )                                READI 68
      EQUIVALENCE (OCCUR(00028), GAMMA )                                READI 69
      EQUIVALENCE (OCCUR(00156),  HREF1)                                READI 70
      EQUIVALENCE (OCCUR(00175),  HREF2)                                READI 71
      EQUIVALENCE (OCCUR(03233), HTAB  )                                READI 72
      EQUIVALENCE (OCCUR(00222), ISP   )                                READI 73
      EQUIVALENCE (OCCUR(00137), LAMDA1)                                READI 74
      EQUIVALENCE (OCCUR(00143), LAMDA2)                                READI 75
      EQUIVALENCE (OCCUR(00138), LA1   )                                READI 76
      EQUIVALENCE (OCCUR(00144), LA2   )                                READI 77
      EQUIVALENCE (OCCUR(00117), MW    )                                READI 78
      EQUIVALENCE (OCCUR(00164), NGL1  )                                READI 79
      EQUIVALENCE (OCCUR(00183), NGL2  )                                READI 80
      EQUIVALENCE (OCCUR(00165), NGT1  )                                READI 81
      EQUIVALENCE (OCCUR(00184), NGT2  )                                READI 82
      EQUIVALENCE (OCCUR(00162), NSL1  )                                READI 83
      EQUIVALENCE (OCCUR(00181), NSL2  )                                READI 84
      EQUIVALENCE (OCCUR(00163), NST1  )                                READI 85
      EQUIVALENCE (OCCUR(00182), NST2  )                                READI 86
      EQUIVALENCE (OCCUR(00112), PHI0  )                                READI 87
      EQUIVALENCE (OCCUR(00223), PSIZET)                                READI 88
      EQUIVALENCE (OCCUR(00114), PSI0  )                                READI 89
      EQUIVALENCE (OCCUR(00109), P0    )                                READI 90
      EQUIVALENCE (OCCUR(00110), Q0    )                                READI 91
      EQUIVALENCE (OCCUR(00057), R     )                                READI 92
      EQUIVALENCE (OCCUR(00136), RB1   )                                READI 93
      EQUIVALENCE (OCCUR(00142), RB2   )                                READI 94
      EQUIVALENCE (OCCUR(00063), RE    )                                READI 95
      EQUIVALENCE (OCCUR(00158), RHO21 )                                READI 96
      EQUIVALENCE (OCCUR(00177), RHO22 )                                READI 97
      EQUIVALENCE (OCCUR(00135), RN1   )                                READI 98
      EQUIVALENCE (OCCUR(00141), RN2   )                                READI 99
      EQUIVALENCE (OCCUR(00116), SIG   )                                READI100
      EQUIVALENCE (OCCUR(00111), SMR0  )                                READI101
      EQUIVALENCE (OCCUR(03033), TABIX1)                                READI102
      EQUIVALENCE (OCCUR(03083), TABIX2)                                READI103
      EQUIVALENCE (OCCUR(02933), TABI1 )                                READI104
      EQUIVALENCE (OCCUR(02983), TABI2 )                                READI105
      EQUIVALENCE (OCCUR(03771), TABRHO)                                READI106
      EQUIVALENCE (OCCUR(03821), TABSND)                                READI107
      EQUIVALENCE (OCCUR(03133), TABZ1 )                                READI108
      EQUIVALENCE (OCCUR(03183), TABZ2 )                                READI109
      EQUIVALENCE (OCCUR(03721), TBATMZ)                                READI110
      EQUIVALENCE (OCCUR(00077), TCRIT )                                READI111
      EQUIVALENCE (OCCUR(00078), TECUN )                                READI112
      EQUIVALENCE (OCCUR(03618), THDELT)                                READI113
      EQUIVALENCE (OCCUR(03593), THDELZ)                                READI114
      EQUIVALENCE (OCCUR(00113), THEAL0)                                READI115
      EQUIVALENCE (OCCUR(00134), THETA1)                                READI116
      EQUIVALENCE (OCCUR(00140), THETA2)                                READI117
      EQUIVALENCE (OCCUR(00224), THEZET)                                READI118
      EQUIVALENCE (OCCUR(03568), THTHO )                                READI119
```

```
      EQUIVALENCE (OCCUR(00207), THO   )                                READI120
      EQUIVALENCE (OCCUR(00132), TINIT )                                READI121
      EQUIVALENCE (OCCUR(00209), TOFF  )                                READI122
      EQUIVALENCE (OCCUR(00208), TON   )                                READI123
      EQUIVALENCE (OCCUR(01644), TRAJRN)                                READI124
      EQUIVALENCE (OCCUR(01344), TRAJT )                                READI125
      EQUIVALENCE (OCCUR(01494), TRAJY )                                READI126
      EQUIVALENCE (OCCUR(01569), TRAJW )                                READI127
      EQUIVALENCE (OCCUR(01419), TRAJZ )                                READI128
      EQUIVALENCE (OCCUR(01719), TRJALP)                                READI129
      EQUIVALENCE (OCCUR(00243), TRZTR )                                READI130
      EQUIVALENCE (OCCUR(00123), TST   )                                READI131
      EQUIVALENCE (OCCUR(00148), TWST  )                                READI132
      EQUIVALENCE (OCCUR(00149), TW1   )                                READI133
      EQUIVALENCE (OCCUR(00168), TW2   )                                READI134
      EQUIVALENCE (OCCUR(02833), TXCGD1)                                READI135
      EQUIVALENCE (OCCUR(02883), TXCGD2)                                READI136
      EQUIVALENCE (OCCUR(00102), TO    )                                READI137
      EQUIVALENCE (OCCUR(00247), UPBNDZ)                                READI138
      EQUIVALENCE (OCCUR(00106), VO    )                                READI139
      EQUIVALENCE (OCCUR(03458), WCDTAB)                                READI140
      EQUIVALENCE (OCCUR(03308), WHTAB )                                READI141
      EQUIVALENCE (OCCUR(01119), WTMINF)                                READI142
      EQUIVALENCE (OCCUR(01269), WTPTOT)                                READI143
      EQUIVALENCE (OCCUR(01194), WTRINF)                                READI144
      EQUIVALENCE (OCCUR(01044), WTZ   )                                READI145
      EQUIVALENCE (OCCUR(00133), W1    )                                READI146
      EQUIVALENCE (OCCUR(00139), W2    )                                READI147
      EQUIVALENCE (OCCUR(00238), XLOW  )                                READI148
      EQUIVALENCE (OCCUR(00107), XRO   )                                READI149
      EQUIVALENCE (OCCUR(00237), XUP   )                                READI150
      EQUIVALENCE (OCCUR(00240), X1LOW )                                READI151
      EQUIVALENCE (OCCUR(00239), X1UP  )                                READI152
      EQUIVALENCE (OCCUR(00120), ZBAR  )                                READI153
      EQUIVALENCE (OCCUR(00093), ZETA  )                                READI154
      EQUIVALENCE (OCCUR(00206), ZOFF  )                                READI155
      EQUIVALENCE (OCCUR(00205), ZON   )                                READI156
      EQUIVALENCE (OCCUR(00118), ZPR1  )                                READI157
      EQUIVALENCE (OCCUR(00119), ZPR2  )                                READI158
      EQUIVALENCE (OCCUR(00121), ZST   )                                READI159
      EQUIVALENCE (OCCUR(00145), ZTURN)                                 READI160
      EQUIVALENCE (OCCUR(00109), ZO    )                                READI161
      EQUIVALENCE (NOCCUR(08), IATMOS)                                  READI162
      EQUIVALENCE (NOCCUR(09), IKCMU )                                  READI163
      EQUIVALENCE (NOCCUR(30), INALPH)                                  READI164
      EQUIVALENCE (NOCCUR(29), ITAPE )                                  READI165
      EQUIVALENCE (NOCCUR(23), ITHRST)                                  READI166
      EQUIVALENCE (NOCCUR(07), LJPT  )                                  READI167
      EQUIVALENCE (NOCCUR(20), MATLN1)                                  READI168
      EQUIVALENCE (NOCCUR(21), MATLN2)                                  READI169
      EQUIVALENCE (NOCCUR(06), MAXVAL)                                  READI170
      EQUIVALENCE (NOCCUR(10), MHEAT)                                   READI171
      EQUIVALENCE (NOCCUR(16), MXTAB1)                                  READI172
      EQUIVALENCE (NOCCUR(17), MXTAB2)                                  READI173
      EQUIVALENCE (NOCCUR(18), MAXCD )                                  READI174
      EQUIVALENCE (NOCCUR(19), MAXWCD)                                  READI175
      EQUIVALENCE (NOCCUR(15), NGEOM )                                  READI176
      EQUIVALENCE (NOCCUR(03), NJPT  )                                  READI177
      EQUIVALENCE (NOCCUR(005), NOSEDP)                                 READI178
      EQUIVALENCE (NOCCUR(24), NPLOT )                                  READI179
```

```
      EQUIVALENCE (NOCCUR(14), NPRINT)
      EQUIVALENCE (NOCCUR(22), NTHRST)
      EQUIVALENCE (IOCCUR(00001), IOP    )
C
C
      DATA INCOL1/8888/
C
C
C
      GO TO (1,2),ILL111
1     CONTINUE
      CALL ZREADX
      CALL SETUP(8HOCCUR    ,8,OCCUR,4000)
C
C           CASE IDENTIFICATION INPUTS
      CALL SETUP(8HCASE     ,8,CASE  )
      CALL SETUP(8HDATE     ,8,DATE  )
      CALL SETUP(8HMEMO     ,8,EMO   )
C
C           PRINTING AND STOP CONTROLS
      CALL SETUP(8HNPRINT   ,4,NPRINT)
      CALL SETUP(8HZPR1     ,8,ZPR1  )
      CALL SETUP(8HZBAR     ,8,ZBAR  )
      CALL SETUP(8HZPR2     ,8,ZPR2  )
      CALL SETUP(8HZST      ,8,ZST   )
      CALL SETUP(8HTST      ,8,TST   )
      CALL SETUP(8HALST     ,8,ALST  )
C
C           INTEGRATION ACCURACY CONTROLS
      CALL SETUP(8HDELIN    ,8,DELIN )
      CALL SETUP(8HCHIGH    ,8,CHIGH ,16)
      CALL SETUP(8HCDOWN    ,8,CDOWN ,16)
C
C           INPUT TRANSITION ALTITUDE  TRZTR
      CALL SETUP(8HTRZTR    ,8,TRZTR )
C
C           INPUT LIMITS ON THE FAIRING REGIONS
      CALL SETUP(8HX1UP     ,8,X1UP  )
      CALL SETUP(8HX1LOW    ,8,X1LOW )
      CALL SETUP(8HXUP      ,8,XUP   )
      CALL SETUP(8HXLOW     ,8,XLOW  )
C
C           TRAJECTORY OPTION AND INITIAL VALUES OF TRAJECTORY PARAMETERS
      CALL SETUP(8HLOPT     ,4,LOPT  )
      CALL SETUP(8HZO       ,8,ZO    )
      CALL SETUP(8HGAMFO    ,8,GAMFO )
      CALL SETUP(8HVO       ,8,VO    )
      CALL SETUP(8HTO       ,8,TO    )
      CALL SETUP(8HXRO      ,8,XRO   )
      CALL SETUP(8HPO       ,8,PO    )
      CALL SETUP(8HQO       ,8,QO    )
      CALL SETUP(8HSMRO     ,8,SMRO  )
      CALL SETUP(8HPHIO     ,8,PHIO  )
      CALL SETUP(8HPSIO     ,8,PSIO  )
      CALL SETUP(8HTHEALO   ,8,THEALO)
C
C           INPUT TEST QUANTITIES FOR ANGLE OF ATTACK CYCLE TIME
      CALL SETUP(8HTCRIT    ,8,TCRIT )
      CALL SETUP(8HTECON    ,8,TECON )
C
```

```
C          NUMBER OF TABULAR INPUTS FOR EITHER THE INPUT WIND TUNNEL
C        OPTION OR THE INPUT TRAJECTORY OPTION.
      CALL SETUP(8HMAXVAL  ,4,MAXVAL)
C
C          INPUTS FOR WIND TUNNEL OPTION
      CALL SETUP(8HWTZ     ,8,WTZ,75)
      CALL SETUP(8HWTMINF  ,8,WTMINF,75)
      CALL SETUP(8HWTRINF  ,8,WTRINF,75)
      CALL SETUP(8HWTPTOT  ,8,WTPTOT,75)
C
C          INPUTS FOR TABULAR INPUT TRAJECTORY
      CALL SETUP(8HTRAJT   ,8,TRAJT,75)
      CALL SETUP(8HTRAJZ   ,8,TRAJZ,75)
      CALL SETUP(8HTRAJV   ,8,TRAJV,75)
      CALL SETUP(8HTRAJW   ,8,TRAJW,75)
      CALL SETUP(8HTRAJRN  ,8,TRAJRN,75)
      CALL SETUP(8HTRJALP  ,8,TRJALP,75)
C
C          TABULAR INPUT ANGLE OF ATTACK FOR USE WITH LOPT=1 TRAJECTORY
      CALL SETUP(8HINALPH  ,4,INALPH)
      CALL SETUP(8HALPTAB  ,8,ALPTAB,75)
C
C          MASS LOSS OPTION INPUTS
      CALL SETUP(8HMOPT    ,4,MOPT  )
      CALL SETUP(8HMHEAT   ,4,MHEAT )
      CALL SETUP(8HNOSEOP  ,4,NOSEOP)
C
C          TWST IS A WALL TEMPERATURE USED IN RAREFIED FLOW CALCULATIONS
      CALL SETUP(8HTWST    ,8,TWST  )
C
C          TINIT IS INTERNAL TEMPERATURE OF VEHICLE USED TO OBTAIN
C        TEMPERATURE GRADIENT IN MASS LOSS CALCULATIONS USING THE
C        ITERATIVE METHOD.
      CALL SETUP(8HTINIT   ,8,TINIT )
C
C          OPTION INDICATING THE SET OF GEOMETRIC VARIABLES BEING INPUT
      CALL SETUP(8HNGEOM   ,4,NGEOM )
C          OPTION TO INPUT CMQ
      CALL SETUP(8HIKCMQ   ,4,IKCMQ )
C          GEOMETRY INPUTS FOR THE FIRST CONFIGURATION
      CALL SETUP(8HW1      ,8,W1    )
      CALL SETUP(8HTHETA1  ,8,THETA1)
      CALL SETUP(8HRN1     ,8,RN1   )
      CALL SETUP(8HRB1     ,8,RB1   )
      CALL SETUP(8HLAMDA1  ,8,LAMDA1)
      CALL SETUP(8HLA1     ,8,LA1   )
      CALL SETUP(8HMXTAB1  ,4,MXTAB1)
      CALL SETUP(8HCMQIN1  ,8,CMQIN1)
      CALL SETUP(8HTABZ1   ,8,TABZ1 ,50)
      CALL SETUP(8HTXCG01  ,8,TXCG01,50)
      CALL SETUP(8HTABI1   ,8,TABI1 ,50)
      CALL SETUP(8HTABIX1  ,8,TABIX1,50)
C
C          FIRST CONFIGURATION HEATSHIELD PROPERTIES
      CALL SETUP(8HMATLN1  ,4,MATLN1)
      CALL SETUP(8HTW1     ,8,TW1   )
      CALL SETUP(8HBETA11  ,8,BETA11)
      CALL SETUP(8HBETA21  ,8,BETA21)
      CALL SETUP(8HBETA31  ,8,BETA31)
      CALL SETUP(8HBETA41  ,8,BETA41)
```

```
      CALL SETUP(8HHREF1   ,8,HREF1 )
      CALL SETUP(8HF1      ,8,F1    )
      CALL SETUP(8HRHO21   ,8,RHO21 )
      CALL SETUP(8HDELRH1  ,8,DELRH1)
      CALL SETUP(8HCP21    ,8,CP21  )
      CALL SETUP(8HCPG1    ,8,CPG1  )
      CALL SETUP(8HNSL1    ,8,NSL1  )
      CALL SETUP(8HNST1    ,8,NST1  )
      CALL SETUP(8HNGL1    ,8,NGL1  )
      CALL SETUP(8HNGT1    ,8,NGT1  )
      CALL SETUP(8HDELHC1  ,8,DELHC1)
      CALL SETUP(8HEPSIL1  ,8,EPSIL1)
C
C     ZTURN IS ALTITUDE FOR DISCONTINUOUS CONFIGURATION CHANGE
      CALL SETUP(8HZTURN   ,8,ZTURN )
C
C          GEOMETRY INPUTS FOR THE SECOND CONFIGURATION
      CALL SETUP(8HW2      ,8,W2    )
      CALL SETUP(8HTHETA2  ,8,THETA2)
      CALL SETUP(8HRN2     ,8,RN2   )
      CALL SETUP(8HRB2     ,8,RB2   )
      CALL SETUP(8HLAMDA2  ,8,LAMDA2)
      CALL SETUP(8HLA2     ,8,LA2   )
      CALL SETUP(8HMXTAB2  ,4,MXTAB2)
      CALL SETUP(8HCMQIN2  ,8,CMQIN2)
      CALL SETUP(8HTABZ2   ,8,TABZ2 ,50)
      CALL SETUP(8HTXCGD2  ,8,TXCGD2,50)
      CALL SETUP(8HTABI2   ,8,TABI2 ,50)
      CALL SETUP(8HTABIX2  ,8,TABIX2,50)
C
C          SECOND CONFIGURATION HEATSHIELD PROPERTIES
      CALL SETUP(8HMATLN2  ,4,MATLN2)
      CALL SETUP(8HTW2     ,8,TW2   )
      CALL SETUP(8HBETA12  ,8,BETA12)
      CALL SETUP(8HBETA22  ,8,BETA22)
      CALL SETUP(8HBETA32  ,8,BETA32)
      CALL SETUP(8HBETA42  ,8,BETA42)
      CALL SETUP(8HHREF2   ,8,HREF2 )
      CALL SETUP(8HF2      ,8,F2    )
      CALL SETUP(8HRHO22   ,8,RHO22 )
      CALL SETUP(8HDELRH2  ,8,DELRH2)
      CALL SETUP(8HCP22    ,8,CP22  )
      CALL SETUP(8HCPG2    ,8,CPG2  )
      CALL SETUP(8HNSL2    ,8,NSL2  )
      CALL SETUP(8HNST2    ,8,NST2  )
      CALL SETUP(8HNGL2    ,8,NGL2  )
      CALL SETUP(8HNGT2    ,8,NGT2  )
      CALL SETUP(8HDELHC2  ,8,DELHC2)
      CALL SETUP(8HEPSIL2  ,8,EPSIL2)
C
C          DRAG COEFFICIENT INPUTS
      CALL SETUP(8HMAXCD   ,4,MAXCD )
      CALL SETUP(8HHTAB    ,8,HTAB  ,75)
      CALL SETUP(8HCDTAB   ,8,CDTAB ,75)
C
C          INCREMENT IN DRAG COEFF. INPUTS
      CALL SETUP(8HMAXWCD  ,4,MAXWCD)
      CALL SETUP(8HWHTAB   ,8,WHTAB ,75)
      CALL SETUP(8HWCDTAB  ,8,WCDTAB,75)
      CALL SETUP(8HAWREF   ,8,AWREF )
```

```
C
C          THRUSTING OPTION INPUTS
      CALL SETUP(8HNTHRUST ,4,NTHRST)
      CALL SETUP(8HTON     ,8,TON   )
      CALL SETUP(8HTOFF    ,8,TOFF  )
      CALL SETUP(8HZON     ,8,ZON   )
      CALL SETUP(8HZOFF    ,8,ZOFF  )
      CALL SETUP(8HITHRST  ,4,ITHRST)
      CALL SETUP(8HTHTHO   ,8,THTHO ,25)
      CALL SETUP(8HTHDELT  ,8,THDELT,25)
      CALL SETUP(8HTHDELZ  ,8,THDELZ,25)
      CALL SETUP(8HTHO     ,8,THO   )
      CALL SETUP(8HISP     ,8,ISP   )
      CALL SETUP(8HAE      ,8,AE    )
      CALL SETUP(8HPSIZET  ,8,PSIZET)
      CALL SETUP(8HTHEZET  ,8,THEZET)
      CALL SETUP(8HDELY    ,8,DELY  )
      CALL SETUP(8HDELZ    ,8,DELZ  )
C
C          INPUTS FOR TABULAR INPUT ATMOSPHERE
      CALL SETUP(8HIATMOS  ,4,IATMOS)
      CALL SETUP(8HUPBNDZ  ,8,UPBNDZ)
      CALL SETUP(8HDNBNDZ  ,8,DNBNDZ)
      CALL SETUP(8HTBATMZ  ,8,TBATMZ,50)
      CALL SETUP(8HTABRHO  ,8,TABRHO,50)
      CALL SETUP(8HTABSND  ,8,TABSND,50)
C
C          PLOTTING AND TAPE INPUTS
      CALL SETUP(8HNPLOT   ,4,NPLOT ,5)
      CALL SETUP(8HITAPE   ,4,ITAPE )
C
C          PHYSICAL CONSTANTS
      CALL SETUP(8HC       ,8,C     )
      CALL SETUP(8HR       ,8,R     )
      CALL SETUP(8HGAMMA   ,8,GAMMA )
      CALL SETUP(8HRE      ,8,RE    )
      CALL SETUP(8HCAPG    ,8,CAPG  )
      CALL SETUP(8HG       ,8,G     )
      CALL SETUP(8HZETA    ,8,ZETA  )
      CALL SETUP(8HSIG     ,8,SIG   )
      CALL SETUP(8HMW      ,8,MW    )
C
C          CURVE FIT CONSTANTS
      CALL SETUP(8HA       ,8,A,516)
      CALL SETUP(8HB       ,8,B,21)
C
    2 CALL READIN(INCOL1,$99)
      IOP(73) = 0
      DO 66 I = 1, 3
      J = I & 63
      IOP(J) = 0
       DO 55 K = I , 21 , 3
      IF(IOP(K).EQ.1) IOP(J) = 1
   55  CONTINUE
      IF(IOP(J).EQ.1) IOP(73) = 1
   66  CONTINUE
       IOP(74) = 0
      ITEM = 0
      DO 88 I = 22, 27
       J = I & 45
```

```
      IOP(J) = 0
       IF(IOP(J&10).EQ.1) ITEM = 1
       DO 77 K = I , 63 , 6
       IF(IOP(K).EQ.1)  IOP(J) = 1
77     CONTINUE
       IF(ITEM.EQ.1)  IOP(74) = 1
      IF(IOP(J).EQ.1) IOP(74) = 1
88    CONTINUE
      CALL HEDING(6)
      GO TO 98
99    IF(IMPLOT.EQ.1) CALL AVPLT(H,H,H,H,H,5,1)
      IF(ITAPE.NE.0)END FILE 8
      IF(ITAPE.NE.0)REWIND 8
      CALL EXIT
   98 RETURN
      END
```

```
CREADY
      SUBROUTINE READY
        IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FO,E, P,TD,RS,SL, Z,Q,A,EL,
     2DELTA,FAC,C(40,10)
      COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC
      COMMON /NALTFG/FUSM,NALT
      COMMON /END/ ITERM
C     READY
C
C
C          SUBROUTINE READY PERFORMS THE CALCULATIONS OF THE DAVIDON
C        MINIMIZATION METHOD WHICH ESTABLISH A DIRECTION ALONG WHICH TO
C        SEARCH FOR A RELATIVE MINIMUM AND BOX OFF AN INTERVAL IN
C        THIS DIRECTION WITHIN WHICH A RELATIVE MINIMUM IS LOCATED.
C
C
      L=L
      GO TO (200,201),L
  200 IT=1
  201 CALL MATMP (N,N,H,G,S)
C
  202 DO 203 I=1,N
  203 S(I)=-S(I)
C
  204 M=1
  205 CALL MATMP (M,N,S,G,GS)
  206 IF (GS&E)207,227,227
C
207   TP1 = -FUSM*(F/GS)
      WRITE(6,1000) GS, E, TP1, IT
      EL =  DMIN1(2.0D0, TP1)
  209 SL=-GS
C
  210 DO 211 I=1,N
  211 XP(I)=X(I)&EL*S(I)
C
  212 M1=2
  213 CALL FCN(N,GP,FP,XP,M1)
      IF(ITERM.EQ.0) GO TO 214
      DO 100I = 1,N
100   X(I) = XP(I)
      GO TO 222
C
  214 M=1
  215 CALL MATMP (M,N,S,GP,GSP)
  216 IF (GSP)217,229,229
  217 IF (FP-F)218,229,229
  218 WRITE(6,1)
C
      IF(NALT) 231, 231, 10
10    EL = 2.D0*EL
      GO TO 210
  231 FB=FP
  232 DO 234 I=1,N
  233 GB(I)=GP(I)
  234 T(I)=XP(I)
  220 IF (EL-2.0)221,223,223
```

```
  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 ='I7)  READY 74
      END                                                          READY 75
```

```
CREDUCE
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE REDUCE(CUP,KRED,IRED,LRED,WRF)
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      COMMON OCCUR,NOCCUR
      COMMON /IDNUS/ ID1(50), ID2(50)
      COMMON/IOPT/IPROC,IN,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON /END/ ITERM
      DIMENSION CUP(20)
      DIMENSION OCCUR(4000),NOCCUR(30)
C
C
C            SUBROUTINE REDUCE CAN MULTIPLY ANY SPECIFIED ELEMENT OF THE
C          OCCUR ARRAY BY THE REDUCTION FACTOR WRF AND SET THE FIRST
C          ELEMENT OF THE CUP ARRAY EQUAL TO ANY SPECIFIED ELEMENT OF THE
C          OCCUR ARRAY MULTIPLIED BY THE FACTOR WRF.
C
C
C
      IZ = ID2(1)
      IRED = IRED & 1
      IF(IRED.EQ.LRED) GO TO 100
      IF(ITERM) 10, 50, 60
C
C
10    DO 20 I = 1, IN
      IF(ID1(I).EQ.IZ) GO TO 40
20    CONTINUE
C
      IF(ICOM(3).EQ.1) GO TO 40
C
30    OCCUR(IZ) = WRF*OCCUR(IZ)
      KRED = 0
      GO TO 200
C
40    CUP(1) = WRF*OCCUR(IZ)
      KRED = 0
      GO TO 200
C
50    KRED = 1
C      50 MAY BE USED FOR LATER IMPLEMENTATION
C
C
      GO TO 200
C
60    KRED = 1
      GO TO 200
C
C
100    KRED = -1
C
200   RETURN
      END
```

```
CRITOUT
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE RITOUT
C
      DIMENSION ALMAX(200),ALMIN(200),TAMAX(200),TAMIN(200),ZMAX(200),
     1ZMIN(200),FMAX(200),FMIN(200),OCCUR(4000),NOCCUR(30)
C
      EQUIVALENCE
     1(OCCUR(1044),ALMAX(1)),(OCCUR(1244),ALMIN(1)),
     2(OCCUR(1444),TAMAX(1)),(OCCUR(1644),TAMIN(1)),
     3(OCCUR(1844),ZMAX(1)), (OCCUR(2044), ZMIN(1)),
     4(OCCUR(2244),FMAX(1)), (OCCUR(2444), FMIN(1))
      EQUIVALENCE
     1(NOCCUR(11), IKMAX),(NOCCUR(12), IKMIN),(NOCCUR(14),NPRINT)
C
      COMMON OCCUR,NOCCUR
C
C
C         SUBROUTINE RITOUT PRINTS OUT A SUMMARY OF THE MAXIMUMS AND
C         MINIMUMS IN ANGLE OF ATTACK AS DETERMINED BY THE TESTING IN
C         SUBROUTINE VIXEN.
C
      IF(NPRINT.EQ.0)RETURN
   51 WRITE(6,1051)
 1051 FORMAT(1H0 50X,23HMAXIMUMS IN ALPHA PRIME/40X,4HTIME,9X,1HZ,12X,
     13HSMF,6X,11HALPHA PRIME)
C
      DO 52 J=1,IKMAX
      ALMAX(J)=57.29578D0*ALMAX(J)
      WRITE(6,1052)TAMAX(J),ZMAX(J),FMAX(J),ALMAX(J)
 1052 FORMAT(1H 38X,F7.2,2E14.5,4X,F8.3)
   52 CONTINUE
      WRITE(6,1053)
 1053 FORMAT(1H0 50X,23HMINIMUMS IN ALPHA PRIME/40X,4HTIME,9X,1HZ,12X,
     13HSMF,6X,11HALPHA PRIME)
C
      DO 53 J=1,IKMIN
      ALMIN(J)=57.29578D0*ALMIN(J)
      WRITE(6,1052)TAMIN(J),ZMIN(J),FMIN(J),ALMIN(J)
   53 CONTINUE
      RETURN
      END
```

```
CROSBRK
      SUBROUTINE ROSBRK
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON OCCUR, NOCCUR
      COMMON/BLKO/HH(1600),P(40),O(40),E(40),DP(40),A(40),
     1 B(40),GS,GSP,GTP,GSS,GTT,GSB,FZZ,FP,FB,FO,Z,PZZ,TD,RS,SL,ZZZ,
     1DZZ,AXX,EL,
     2DELXX,FAK,CXX(40,10)
      COMMON/DOPT/EO(20)
      COMMON/END/ITERM
      COMMON/IOPT/IPROC,NMAX,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      DIMENSION C(20,20)
      DIMENSION OCCUR(4000), NOCCUR(30)
      EQUIVALENCE (HH(401),C(1))
      EQUIVALENCE (XCOM(1),ALPHA)
      EQUIVALENCE (XCOM(2),BETA)
      EQUIVALENCE (XCOM(3),GAMMA)
      EQUIVALENCE (XCOM(4),DEL)
      EQUIVALENCE (XCOM(5),RATU)
      EQUIVALENCE (XCOM(6),TOL)
C
C
C          SUBROUTINE ROSBRK ATTEMPTS TO MINIMIZE THE VALUE OF A FUNC-
C        TION U( P1,P2,....,P(NMAX) ) WHERE THE P'S ARE ADJUSTABLE
C        DESIGN PARAMETERS BY THE ROSENBROCK ROTATING COORDINATE MINI-
C        MIZATION TECHNIQUE.
C
C     NMAX IS THE NUMBER OF PARAMETERS TO BE ADJUSTED. ITS MAXIMUM
C     VALUE IS 20.
C
C
      NTRIA = 0
      NSTAG = 0
      INDEX = 0
C
  123 CALL FEV(NMAX,P,U)
      IF(ITERM.NE.0) GO TO 600
C
      IF(NTRIA.LE.LIMIT) GO TO 610
      WRITE(6,611)
  611 FORMAT(1HO 34HNTRIA GREATER THAN LIMIT IN ROSBRK)
      ITERM = 1
      GO TO 120
  610 INDEX = INDEX&1
C
      IF(NTRIA.NE.0) GO TO 103
C
      ULAST = U
      USTAG = U
      UPREV = U
      NSUCC = 0
      NTRIA = 1
      DO 104 I=1,NMAX
  104 E(I) = EO(I)
C
C  C  MATRIX IS SET TO A UNIT MATRIX
C
      DO 105 I=1,NMAX
```

```
      DO 105 J=1,NMAX
      IF(I-J) 106,107,106
  106 C(I,J) = 0.D0
      GO TO 105
  107 C(I,J) = 1.D0
  105 CONTINUE
      N=1
      D(1) = 0.D0
      GO TO 108
C
C  SECOND AND SUBSEQUENT TRIALS START HERE
C
C
  103 NTRIA = NTRIA & 1
      IF(U-ULAST*(1.D0&TOL)) 109,109,110
C
C  IF TRIAL WAS A SUCCESS,START HERE
C
  109 ULAST = U
      NSUCC = NSUCC & 1
      D(N) = D(N) & E(N)
      IF(NSUCC.GE.20) GO TO 50
      E(N) = ALPHA*E(N)
      GO TO 108
C
C     IF TRIAL WAS A FAILURE,START HERE
  110 CONTINUE
      WRITE(6,306)NTRIA,N,NSTAG,NSUCC,U,NMAX,(P(I),I=1,NMAX)
 306  FORMAT(1H024H NTRIA     N NSTAG NSUCC19X1HU/ 1H 4I6,F20.10/1H
     1                13HP(I) I = 1 TO I3/(1H 6F20.10))
      DO 111 I=1,NMAX
  111 P(I) = P(I) - DP(I)
      U = ULAST
      IF(NSUCC) 112,112,113
  112 E(N) = -BETA*E(N)
      GO TO 125
 50   WRITE(6,922) NSUCC
 922  FORMAT(1H0 15HWARNING-NSUCC= I6,1X 22HGOING TO NEXT VARIABLE)
C
C     SEARCH IN ONE DIRECTION
C
  113 E(N) = GAMMA*D(N)
C IF N = NMAX, THIS STAGE IS COMPLETE.
      IF(N.EQ.NMAX)GO TO 115
      N=N&1
      D(N) = 0.D0
      NSUCC = 0
      GO TO 125
C
C        STOPPING CRITERIA IS APPLIED AFTER 2ND AND SUBSEQUENT STAGES
C
  115 IF(NSTAG) 116,116,117
  117 ULIM = UPREV - DEL*UPREV
      IF((UPREV-USTAG).EQ.0.C0) GO TO 121
      URAT = (USTAG-ULAST)/(UPREV-USTAG)
      IF(ULAST - ULIM)121,121,118
  118 IF(USTAG - ULIM)121,121,119
  119 IF(URAT - RATO)120,121,121
C
  120 WRITE(6,301)
```

```
301    FORMAT(1H0 26HROSBRK SEARCH IS COMPLETE.)
       WRITE(6,302) ULAST,(P(I),I=1,NMAX)
 302  FORMAT(1H037HFINAL PERFORMANCE FUNCTION.  ULAST = E20.10/1H
     120HFINAL PARAMETERS ARE /(1H 6E20.10))
       WRITE(6,303)NTRIA,NSTAG
 303  FORMAT(1H018HNUMBER OF TRIALS = 15/1H 18HNUMBER OF STAGES = I5)
       GO TO 600
C
C SEARCH IS COMPLETE
C
C
C SEARCH WILL CONTINUE IN A NEW COORDINATE SYSTEM.
  121 UPREV = USTAG
  116 USTAG = ULAST
      NSTAG = NSTAG&1
C
C DEFINE NEW COORDINATE SYSTEM.
      CALL GRAM
      WRITE(6,304)
304   FORMAT(1H0 20HC MATRIX FROM ROSBRK/  )
      DO 200 I=1,NMAX
  200 WRITE(6,305) (C(I,J),J=1,NMAX)
  305 FORMAT(12(1X,1PE9.2))
  129 N=1
      D(1)= 0.D0
      NSUCC=0
  108 CONTINUE
      WRITE(6,306)NTRIA,N,NSTAG,NSUCC,U,NMAX,(P(I),I=1,NMAX)
C
C DEFINE COORDINATES FOR NEXT TRIAL.
  125 DO 122 I=1,NMAX
      DP(I)=C(I,N)*E(N)
  122 P(I)=P(I)&DP(I)
C
      WRITE(6,500)(DP(I),I=1,NMAX)
      WRITE(6,501)NMAX,(P(I),I=1,NMAX)
      WRITE(6,503) N,E(N)
500   FORMAT(1H014HIN ROSBRK DP =/(2H 1P6E20.7))
501   FORMAT(1H013HP(I) I = 1 TO 13/(1H 1P6E20.7))
503   FORMAT(1H0 13HIN ROSBRK N = I3,2X6HE(N) =E20.10)
      GO TO 123
600   RETURN
      END
```

```
CROTATE
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE ROTATE(DERIV,LP)
C
      REAL*8 MX,MY,MZ
      REAL*8 NUMER1,NUMER2
C
      DIMENSION TABIX(50),TABI(50),TABZ(50),OCCUR(4000),NOCCUR(30),
     1          DERIV(16),XJR(25),XJI(25)
      DIMENSION  XNR(25), XNI(25)
C
      EQUIVALENCE
     1(OCCUR(001),AREF   ),(OCCUR(002),ALPHA  ),(OCCUR(003),ALPRIM),
     2(OCCUR(012),CNALP  ),(OCCUR(013),CMALP  ),(OCCUR(015),CD     ),
     3(OCCUR(020),CMQ    ),(OCCUR(021),D      ),(OCCUR(027),G      ),
     4(OCCUR(042),PI     ),(OCCUR(043),P      ),(OCCUR(044),PHI    ),
     5(OCCUR(045),PSI    ),(OCCUR(050),Q      ),(OCCUR(051),QD     ),
     6(OCCUR(055),RHOINF),(OCCUR(056),RHOIN1),
     7(OCCUR(065),SMR    ),(OCCUR(071),THEALP),(OCCUR(075),T      ),
     8(OCCUR(077),TCRIT  ),(OCCUR(078),TECON  ),(OCCUR(081),SMF    ),
     9(OCCUR(082),V      ),(OCCUR(084),W      ),(OCCUR(090),XBAR   ),
     X(OCCUR(091),Z      )
      EQUIVALENCE
     1(OCCUR(108),ZO     ),(OCCUR(113),THEALO),(OCCUR(122),ALST   ),
     2(OCCUR(130),ALWIG2),(OCCUR(131),ALBARP),(OCCUR(202),CM     ),
     3(OCCUR(203),CN     ),(OCCUR(210),MX     ),(OCCUR(211),MY     ),
     4(OCCUR(212),MZ     ),(OCCUR(221),SINGO ),(OCCUR(229),ALALI  ),
     5(OCCUR(237),XUP    ),
     6(OCCUR(894),TABI(1)),(OCCUR(944),TABIX(1)),(OCCUR(994),TABZ(1))
C
      EQUIVALENCE
     1(NOCCUR(04),MAXTAB),(NOCCUR(07),LOPT   )
C
      COMMON OCCUR,NOCCUR
C
C
C
C
C        SUBROUTINE ROTATE CALCULATES THE ANGLE OF ATTACK BY ONE OF
C     TWO METHODS DEPENDING ON THE VALUE OF THE INPUT LOPT.
C     LOPT = 0 IS AN UNCOUPLED THREE DEGREE OF FREEDOM CALCULATION.
C     LOPT = 2 IS A SIMPLIFIED ANGLE OF ATTACK SOLUTION REQUIRING NO
C     INTEGRATION.
C        THE FOLLOWING DERIVATIVES ARE CALCULATED IN THIS SUBROUTINE -
C        DERIV(8)=PSI DOT
C        DERIV(9)=THEALP DOT
C        DERIV(10)=PHI DOT
C        DERIV(11)=Q DOT
C        DERIV(12)=SMR DOT
C        DERIV(13)=P DOT
C
C
C        TESTING LOPT
      IF(LOPT.EQ.0)GO TO 8
C        FOR A NON-ZERO VALUE OF LOPT THE DERIVATIVES ARE ZEROED.
      DO 9 J=8,13
    9 DERIV(J)=0.D0
      IF(LOPT.EQ.2)GO TO 15
      RETURN
```

```
C
C        FOR LOPT=O USING A THREE DEGREE OF FREEDOM IN ROTATION METHOD
C           WHICH CALCULATES THE DERIVATIVES WITH RESPECT TO TIME OF THE
C           EULER ANGLES - PSI, THETA SUB ALPHA, AND PHI - AND OF THE
C           ANGULAR VELOCITIES - P,Q,SMR.
    8 CONTINUE
C             TRIGONOMETRIC FUNCTIONS OF THE EULER ANGLES.
    3 SINP=SIN(PHI)
      COSP=COS(PHI)
      SINI=SIN(PSI)
      COSI=COS(PSI)
      SINL=SIN(THEALP)
      COSL=COS(THEALP)
C             CALCULATION OF THE DERIVATIVES OF THE EULER ANGLES, WHERE
C          DERIV(8)=PSI DOT, DERIV(9)=THETA SUB ALPHA DOT, DERIV(10)=
C            PHI DOT.
      DERIV(8)=(SMR*COSP&Q*SINP)/COSL
      DERIV(9)=Q*COSP-SMR*SINP
      DERIV(10)=P&DERIV(8)*SINL
      COSLP=COSL*COSI
C             DETERMINATION OF THE ANGLE OF ATTACK FROM THE ARCCOSINE OF
C          THE PRODUCT -  COS(THETA SUB ALPHA)*COS(PSI).
      IF(COSLP.LT.0.DO)GO TO 100
      ALPRIM=ACOSK(COSLP)
      CN=CNALP*ALPRIM
      GO TO 4
  100 WRITE(6,101)
  101 FORMAT(1H0 10X,76HTHE ARGUMENT OF ALPRIM IN SUBROUTINE ROTATE IS N
     1EGATIVE. GOING TO NEXT CASE.)
      LP=6
      RETURN
C
C             MOMENTS OF INERTIA ARE DETERMINED FROM THE INPUT TABLES.
    4 CONTINUE
      LL=-1
        UI=TABLE(Z,TABZ, TABI,MAXTAB,LL)
       UIX=TABLE(Z,TABZ,TABIX,MAXTAB,LL)
      IF(UI.EQ.0.0D0)GO TO 13
      IF(UIX.EQ.0.0D0)GO TO 14
    6 CM=CMALP*ALPRIM
C
C             CALCULATION OF DERIVATIVES OF THE ANGULAR VELOCITIES, WHERE
C          DERIV(11)=Q DOT, DERIV(12)=SMR DOT, AND DERIV(13)=P DOT.
      TEM1=QD*AREF*D/UI
      TEM2=0.DO
      IF(ALPRIM.NE.0.DO)TEM2=CM/SIN(ALPRIM)
      TEM3=SINL*COSI
      TEM4=0.5DO*D*CMQ/V
      TEM5=(UI-UIX)*P/UI
      DERIV(11)=TEM1*(TEM2*(TEM3*COSP&SINI*SINP)&TEM4*Q)&TEM5*SMR&MY/JI
      DERIV(12)=TEM1*(TEM2*(SINI*COSP-TEM3*SINP)&TEM4*SMR)-TEM5*Q&MZ/JI
      DERIV(13)=MX/UIX
C
      IF(XBAR.LT.XUP)GO TO 10
   11 ALPHA=ALPRIM
      RETURN
C
C             IN NON-CONTINUUM FLOW REGIME, PERIOD OF THE CYCLE IS TESTED
C          TO DETERMINE APPROPRIATE ANGLE OF ATTACK FOR USE IN DRAGCO.
   10 IF(T.EQ.0.DO)GO TO 11
```

```
      IF(T.GE.TCRIT)GO TO 11
      IF(T.GE.TECON)GO TO 12
      IF(P.EQ.0.D0)ALPHA=2.D0*ALWIG2/PI
      IF(P.NE.0.D0)ALPHA=0.5D0*(ALWIG2&ALBARP)
      GO TO 12
C
C
C
C          SIMPLIFIED ANGLE OF ATTACK SOLUTION WHICH REQUIRES NO
C         CALCULATION OF DERIVATIVES OR SUBSEQUENT INTEGRATION.
   15 LL=-1
      IF(Z.LE.100.0D0)GO TO 90
      UI=TABLE(Z,TABZ,TABI,MAXTAB,LL)
      IF(UI.EQ.0.0D0)GO TO 13
      BETAZ=-DLOG(RHOIN1/0.076474D0)
      IF(BETAZ.LE.0.0001D0)BETAZ=0.0001D0
      IF(Z.EQ.ZO)CD=0.8
      AK1=AREF*G*Z*(2.0D0*CD-CNALP&CMQ*D*D*W/(UI*G))/
     1(4.0D0*BETAZ*W*DABS(SINGO))
      AK2=AREF*G*Z*Z*((CNALP[illegible])*BETAZ*DABS(SINGO)/Z-CMALP*D*W/(UI*G))/
     1(2.0D0*W*(BETAZ*SINGO)**2)
   17 CONTINUE
      TBAR=-2.0D0*PI*Z/(BETAZ*V*SINGO*DSQRT(AK2*RHOINF))
      IF(TBAR.LE.TECON)GO TO 16
      IF(TBAR.LT.1.0D-10)GO TO 16
C
C          OSCILLATORY SOLUTION FOR CYCLE TIMES, TBAR, GREATER THAN
C         TECON.
      SMF=1.0D0/TBAR
      TEM=2.0D0*DSQRT((AK1&AK2)*RHOINF)
      CALL BESSEL(TEM,0.0D0,1.0D-5,10,XJR,XJI,1)
      CALL NEUMAN(1.0D-5,TEM,0.0D0,10,1.0D1,XNR,XNI,XJR(1),XJI(1),XJR(2)
     1,XJI(2),1)
      IF(Z.NE.ZO)GO TO 2
      DTEMDZ=-0.5D0*TEM*BETAZ/Z
      DENAC2=DTEMDZ*(-XNR(2)*XJR(1)&XJR(2)*XNR(1))
      NUMER1=(XJR(1)*AK1*THEALO*2.3769D-3*BETAZ)/(Z*DEXP(AK1*RHOINF&
     1BETAZ))
      NUMER2=THEALO*DTEMDZ*(-XJR(2))*DEXP(-AK1*RHOINF)
      AC2=(NUMER1-NUMER2)/DENAC2
      AC1=(THEALO*DEXP(-AK1*RHOINF)-AC2*XNR(1))/XJR(1)
      ALALI=DEXP(AK1*RHOINF)/DSQRT(PI*DSQRT(AK2*RHOINF))
    2 CONTINUE
      ALPRIM=DEXP(AK1*RHOINF)*(AC1*XJR(1)&AC2*XNR(1))
      CN=CNALP*ALPRIM
      CM=CMALP*ALPRIM
      ALPHA=ALPRIM
      RETURN
C
C          ENVELOPE ANGLE OF ATTACK CALCULATION FOR TBAR LESS THAN OR
C         EQUAL TO TECON.
   16 ALALO=DEXP(AK1*RHOINF)/DSQRT(PI*DSQRT(AK2*RHOINF))
      IF(Z.EQ.ZO)ALALI=ALALO
      ALPRIM=0.63661D0*THEALC*ALALO
      ALPRIM=ALPRIM/ALALI
      SMF=0.0D0
C
C          IF ANGLE OF ATTACK ENVELOPE VALUE IS LESS THAN OR EQUAL TO
C         ALST, INPUT ALPHA STOP VALUE, THE TRAJECTORY IS SWITCHED OVER
C         TO A PARTICLE TRAJECTORY, LUPT=1, AND ANGLE OF ATTACK IS SET
```

```
C        TO ZERO.
      IF(ALPRIM.GT.ALST)GO TO 92
   90 ALPRIM=0.0D0
      LOPT=1
   92 CONTINUE
      CN=CNALP*ALPRIM
      CM=CMALP*ALPRIM
      ALPHA=ALPRIM
   12 RETURN
   13 WRITE(6,1013)Z
      LP=6
      RETURN
   14 WRITE(6,1014)Z
      LP=6
      RETURN
 1013 FORMAT(1H0 10X,37HRESULT OF TABI LOOKUP IS 0.0D0 FOR Z=E12.5)
 1014 FORMAT(1H0 10X,38HRESULT OF TABIX LOCKUP IS 0.0D0 FOR Z=E12.5)
      END
```

```
CSAVEDV
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE SAVEDV(K,DVAR,OCCUR)
      DIMENSION DVAR(50),OCCUR(4000)
C
C
C           SUBROUTINE SAVEDV STORES THE VALUES OF OCCUR(133-145),
C             OCCUR(205-209), AND OCCUR(222) IN LOCATIONS DVAR(1-19)
C             RESPECTIVELY AND LATER RESTORES THESE VALUES TO THE OCCUR
C             ARRAY.
C
C
      K = K
      GO TO (100,200), K
  100 DO 111 I = 1, 13
  111 DVAR(I) = OCCUR(I&132)
      DO 112 I = 14, 18
  112 DVAR(I) = OCCUR(I&191)
      DVAR(19) = OCCUR(222)
      RETURN
  200 DO 222 I = 1, 13
  222 OCCUR(I&132) = DVAR(I)
      DO 223 I = 14, 18
  223 OCCUR(I&191) = DVAR(I)
      OCCUR(222) = DVAR(19)
      RETURN
      END
```

```
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
      COMMON/MULT/SMULT(25)
      DIMENSION PENLTY(25),                    A(25),OCCUR(4000),
     1NOCCUR(30)
      EQUIVALENCE (OCCUR(42),PI),(OCCUR(102),TO),(OCCUR(108),ZO),
     1(OCCUR(121),ZST),(OCCUR(123),TST),(NOCCUR(15),NGEOM),
     2(NOCCUR(22),NTHRST)
      EQUIVALENCE
     1(OCCUR(170),W1F),(OCCUR(150),THET1F),(OCCUR(169),RN1F),
     2(OCCUR(147),RB1F),(OCCUR(151),LAMD1F),(OCCUR(146),LA1F)
      REAL*8DLL(25)/4*0.D0,2*3.D0,2*4.D0,-10.D0,14*0.D0/
      EQUIVALENCE
     1(OCCUR(133),W1    ),(OCCUR(134),THETA1),(OCCUR(135),RN1   ),
     2(OCCUR(136),RB1   ),(OCCUR(137),LAMDA1),(OCCUR(138),LA1   ),
     3(OCCUR(139),W2    ),(OCCUR(140),THETA2),(OCCUR(141),RN2   ),
     4(OCCUR(142),RB2   ),(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.D0,2*.6D0,2*168.D0,40.D0,40.D0,17*0.D0/
      REAL*4M(25)/135,141,137,143,138,144,134,140,145,139,222,205,208,
     1206,209,10*1/
C
C
C         SUBROUTINE SCREEN TESTS THE VALUES OF UP TO TWENTY-FIVE
C         ELEMENTS OF THE OCCUR ARRAY TO DETERMINE WHETHER THESE VALUES
C         LIE WITHIN PRESCRIBED LIMITS.  IF ANY VALUES LIE OUTSIDE THE
C         PRESCRIBED LIMITS, A QUANTITY E IS CALCULATED AS A MEASURE OF
C         ERROR.
C
C
      DO 2 I = 1,25
      PENLTY(I)=0.D0
    2   A(I)=0.D0
        A(1)=1.D0
        A(3)=1.D0
        A(5)=1.D0
        A(7)=1.D0
        A(9)=1.D0
       DLL(12)=ZOFF
      DLL(13)=TO
      DLL(14)=ZST
      DLL(15)=TON
      DUL(9)=ZO
      DUL(10)=W1
      DUL(11)=10000.D0
      DUL(12)=ZO
      DUL(13)=TOFF
      DUL(14)=ZON
      DUL(15)=TST
      IF(ZTURN.LT.0.D0)GO TO 10
      DO 8 I= 2,10,2
    8 A(I)=1.D0
```

```
 10   IF(NTHRST.EQ.0)GO TO 15
      A(11)=1.D0
      IF(NTHRST.EQ.2)GO TO 12
        A(12)=1.D0
        A(14)=1.D0
      GO TO 15
 12   A(13) = 1.0D0
        A(15)=1.D0
C
C
C
15    IF(NGEOM-2) 18, 20, 40
18    LAMDA1 = RN1/RB1
      IF(ZTURN.GE.0.D0) LAMDA2 = RN2/RB2
      GO TO 25
  20  RN1 = LAMDA1*RB1
 25   IF(ZTURN.LT.0.D0)GO TO 30
      RN2=LAMDA2*RB2
      TR=0.01745329D0*THETA2
      ST2=DSIN(TR)
      CT2=DCOS(TR)
      LA2=(RN2*ST2-RN2&RB2*CT2)/ST2
  30  TR=0.01745329D0*THETA1
      ST1=DSIN(TR)
      CT1=DCOS(TR)
      LA1=(RN1*ST1-RN1&RB1*CT1)/ST1
      GO TO 200
 40   LAMDA1 = RN1/RB1
      TEM=LA1-RN1
      TEST=TEM*TEM-RN1*RN1&RB1*RB1
      IF(TEST.LT.0.0D0)GO TO 50
      THETA1=180.D0*ASINR((-(LA1-RN1)*RN1&RB1*DSQRT(TEST))/(TEST&RN1*RN
     1))/PI
      GO TO 60
  50  THETA1=0.D0
  60  IF(ZTURN.LT.0.D0)GO TO 200
      LAMDA2 = RN2/RB2
      TEM=LA2-RN2
      TEST=TEM*TEM-RN2*RN2&RB2*RB2
      IF(TEST.LT.0.D0)GO TO 70
      THETA2=180.D0*ASINR((-(LA2-RN2)*RN2&RB2*DSQRT(TEST))/(TEST&RN2*RN2
     1))/PI
      GO TO 200
  70  THETA2=0.D0
C
C
C
 200  E=0.D0
      DO 500 I=1,25
      IZ=M(I)
      IF(OCCUR(IZ).GT.DLL(I))GO TO 250
      PENLTY(I)=A(I)*SMULT(I)*DABS(OCCUR(IZ)-DLL(I))**2
      GO TO 500
 250  IF(OCCUR(IZ).LT.DUL(I))GO TO 500
      PENLTY(I)=A(I)*SMULT(I)*DABS(OCCUR(IZ)-DUL(I))**2
 500  E=E&PENLTY(I)
C
C
      IF(E.NE.0.D0)GO TO 600
      ISUC=0
```

```
      GO TO 700                                                          SCREE120
C                                                                        SCREE121
  600 ISUC=-1                                                            SCREE122
      WRITE(6,450)PI,TO,ZO,ZST,TST,NGEOM,NTHRST,W1F,THET1F,RN1F,RB1F,    SCREE123
     1LAMD1F,LA1F,W1,THETA1,RN1,RB1,LAMDA1,LA1,W2,THETA2,RN2,RB2,LAMDA2,SCREE124
     2LA2,ZTURN,ISP,ZON,ZOFF,THO,TON,TOFF,(PENLTY(I),I=1,25)             SCREE125
      VAL=E&D                                                            SCREE126
C                                                                        SCREE127
  700 CONTINUE                                                           SCREE128
C                                                                        SCREE129
      RETURN                                                             SCREE130
C                                                                        SCREE131
  450 FORMAT(1H1 13X'PI',13X'TO',13X'ZO',12X'ZST',12X'TST',10X'NGEOM',   SCREE132
     1 9X'NTHRST',12X'W1F'/1H 1P5E15.7,2I15,E15.7/                       SCREE133
     21H0 9X'THET1F',11X'RN1F',11X'RB1F', 9X'LAMD1F',11X'LA1F',13X'W1',  SCREE134
     3 9X'THETA1',12X'RN1'/1H 1P8E15.7/                                  SCREE135
     41H0 12X'RB1', 9X'LAMDA1',12X'LA1',13X'W2', 9X'THETA2',12X'RN2',    SCREE136
     512X'RB2', 9X'LAMDA2'/1H 1P8E15.7/                                  SCREE137
     61H0 12X'LA2',10X'ZTURN',12X'ISP',12X'ZON',11X'ZOFF',12X'THO',      SCREE138
     712X'TON',11X'TOFF'/1H 1P8E15.7/1H0'PENLTY(I),I=1,25'/              SCREE139
     1(1H 1P8E15.7))                                                     SCREE140
      END                                                                SCREE141
```

```
CSR2490
      SUBROUTINE SR2490
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON OCCUR,NOCCUR
      COMMON /CPCCUR/ PCCUR(11770)
      COMMON /CICCUR/ IOCCUR(320)
      REAL*8 ISP,MW
        DIMENSION A(514), B(21), AA(27), CDOWN(16), CHIGH(16), IOP(90)
        DIMENSION NCOMDV(50), NOCCUR(30 ), CCCUR(4000)
        DIMENSION  NPV(160)
      DIMENSION  TABI1(50),TABI2(50),TABIX1(50),TABIX2(50)
      EQUIVALENCE (OCCUR(301),A(1))
      EQUIVALENCE (OCCUR(122),ALST)
      EQUIVALENCE (OCCUR(823),B(1))
      EQUIVALENCE (OCCUR(115),C)
      EQUIVALENCE (OCCUR(19),CAPG)
      EQUIVALENCE (OCCUR(3549),CDOWN(1))
      EQUIVALENCE (OCCUR(3533),CHIGH(1))
      EQUIVALENCE (OCCUR(187),DELIN)
      EQUIVALENCE (OCCUR(189),FACTR1)
      EQUIVALENCE (OCCUR(27),G)
      EQUIVALENCE (OCCUR(28),GAMMA)
      EQUIVALENCE (OCCUR(222),ISP)
      EQUIVALENCE (OCCUR(4000),LP)
      EQUIVALENCE (OCCUR(117),MW)
      EQUIVALENCE (OCCUR(42),PI)
      EQUIVALENCE (OCCUR(57),R)
      EQUIVALENCE (OCCUR(63),RE)
      EQUIVALENCE (OCCUR(116),SIG)
      EQUIVALENCE (OCCUR(3033),TABIX1(1))
      EQUIVALENCE (OCCUR(3083),TABIX2(1))
      EQUIVALENCE (OCCUR(2933),TABI1(1))
      EQUIVALENCE (OCCUR(2983),TABI2(1))
      EQUIVALENCE (OCCUR(78),TECON)
      EQUIVALENCE (OCCUR(132),TINIT)
      EQUIVALENCE (OCCUR(123),TST)
      EQUIVALENCE (OCCUR(148),TWST)
      EQUIVALENCE (OCCUR(149),TW1)
      EQUIVALENCE (OCCUR(168),TW2)
      EQUIVALENCE (OCCUR(236),XLOW )
      EQUIVALENCE (OCCUR(237),XUP  )
      EQUIVALENCE (OCCUR(240),X1LOW)
      EQUIVALENCE (OCCUR(239),X1UP )
      EQUIVALENCE (OCCUR(120),ZBAR)
      EQUIVALENCE (OCCUR(93),ZETA)
      EQUIVALENCE (OCCUR(118), ZPR1)
      EQUIVALENCE (OCCUR(145),ZTURN)
      EQUIVALENCE (NOCCUR(7),LUPT)
      EQUIVALENCE (NOCCUR(20),MATLN1)
      EQUIVALENCE (NOCCUR(21),MATLN2)
      EQUIVALENCE (NOCCUR(16),MXTAB1)
      EQUIVALENCE (NOCCUR(17),MXTAB2)
      EQUIVALENCE (NOCCUR(15),NGEOM)
      EQUIVALENCE (NOCCUR(14),NPRINT)
      EQUIVALENCE (NOCCUR(22),NTHRST)
      EQUIVALENCE (PCCUR(11571), AA    )
      EQUIVALENCE (IOCCUR(309), IMPLOT)
      EQUIVALENCE (IOCCUR(00001), IOP   )
      EQUIVALENCE (IOCCUR(00301), IREF  )
```

```
      EQUIVALENCE (IOCCUR(00302), LPLOT )
      EQUIVALENCE (IOCCUR(00304), NCP   )
      EQUIVALENCE (IOCCUR(00091), NCOMDV)
      EQUIVALENCE (IOCCUR(00305), NDECOY)
      EQUIVALENCE(IOCCUR(003C6), NDVCH )
      EQUIVALENCE(IOCCUR(000307), NPA   )
      EQUIVALENCE (IOCCUR(00141), NPV   )
C
C
C         SUBROUTINE SR2490 PRESETS THE VALUES OF MANY OF THE INPUT
C       VARIABLES.
C
C
      DO 11 I = 1, 4000
      OCCUR(I) = 0.D0
11    CONTINUE
C
       DO 88 I = 1, 11770
       PCCUR(I) = 0.D00
88     CONTINUE
C
      DO 44 I = 1, 30
      NOCCUR(I) = 0
44    CONTINUE
C
       DO 66 I = 1,320
       IOCCUR(I) = 0
66     CONTINUE
C
C
C
      ALST=0.2D0
      C=1.D0
      CAPG=32.2185D0
      DELIN=-2000.0D0
      FACTR1=0.002375D0**0.8D0
      G=32.174D0
      GAMMA=1.4D0
      IMPLOT = 0
      IREF = 1
      ISP=1.0D0
      LOPT=1
      LP = 1
      LPLOT = 1
      MATLN1=1
      MATLN2=1
      MW=28.9D0
      MXTAB1=1
      MXTAB2=1
      NCP = 1
      NDECOY = 1
      NDVCH = 1
      NGEOM=1
      NPA = 1
      NPRINT=1
      PI = 3.14159265358979300
      R=53.5D0
      RE=2.090229D07
      SIG=3.5D0
      TABIX1(1)=1.0D0
```

```
      TABIX2(1)=1.0D0
      TABI1(1)=1.0D0
      TABI2(1)=1.0D0
      TECON=2.0D0
      TINIT=500.0D0
      TST=100.0D0
      TWST=580.0D0
      TW1=1200.0D0
      TW2=1200.0D0
      XLOW=4.0D0
      XUP=6.0D0
      XILOW=0.2D0
      XIUP=0.4D0
      ZBAR=-10000.0D0
      ZETA=0.9D0
      ZPRI=10000.0D0
      ZTURN = -1.D0
C
      DO 22 I = 1, 160
      NPV(I) = 1
22    CONTINUE
C
      DO 33 I = 1, 50
      NCOMDV(I) = 133
33    CONTINUE
C
       DO 99  I = 1, 90
       IOP(I) = 1
99     CONTINUE
C
      DO 4 J=1,16
      CHIGH(J)=1.0D-4
    4 CDOWN(J)=1.0D-5
C
       DO 77 I = 3, 27, 3
      AA(I) = 1.D0
77    CONTINUE
C
      CALL ZPRS(A,B)
C
      RETURN
C
      END
```

```
CSTUFF
      SUBROUTINE STUFF
        IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BLKO/ H(40,40),X(40),G(40),S(40),XP(40),GP(40),T(40),GB(40)
     1,GS, GSP, GTP,GSS, GTT, GSB,F,FP,FB,FO,E, P,TO,RS,SL, Z,D,A,EL,
     2DELTA,FAC,C(40,10)
      COMMON/BLKI/M,N,L,LS,M1,MS,NS,IT,K,NC
C
C
C            SUBROUTINE STUFF, PART OF THE DAVIDON MINIMIZATION METHOD,
C         TESTS HOW WELL THE FUNCTION HAS BEEN MINIMIZED AND HOW WELL
C         THE MATRIX H APPROXIMATES THE INVERSE OF THE MATRIX OF SECOND
C         PARTIAL DERIVATIVES AT THE MINIMUM.
C
C
C     STUFF
  600 K=K-1
  601 IF (K)617,602,602
  602 MS=MS&1
  620 WRITE (6,1)MS,GS
  603 DO 604 I=1,N
C
C
  604 T(I)=RANDOM(DUMMY)-0.5
C
  605 CALL MATMP (N,N,H,T,S)
  606 M1=1
  607 CALL MATMP (M1,N,S,T,TP1)
  608 TP1 = DSQRT(TP1)
  609 EL=P/TP1
  610 DO 611 I=1,N
  611 X(I)=X(I)&EL*S(I)
  612 M1=2
  614 L=1
  616 RETURN
  617 L=2
  618 MS=0
  619 GO TO 616
    1 FORMAT (13HORANDOM STEP I4,5H  GS=1PE14.5)
      END
```

```
CTABLE
      IMPLICIT REAL*8 (A-H,O-Z)
      FUNCTION TABLE(X,XTAB,YTAB,KMAX,L)
      DIMENSION XTAB(2),YTAB(2)
C
C         FUNCTION TABLE PERFORMS A ONE DIMENSIONAL TABLE LOOK UP
C      USING LINEAR INTERPOLATION.
C
      IF(KMAX.EQ.1)GO TO 6
      IF(L.GT.0)GO TO 2
      IGO=1
      IF(XTAB(1).GT.XTAB(2))IGO=2
C
      DO 1 K=2,KMAX
      GO TO(3,4),IGO
    3 IF(X.GT.XTAB(K))GO TO 1
      GO TO 5
    4 IF(X.LT.XTAB(K))GO TO 1
    5 L=K
      GO TO 2
    1 CONTINUE
C
      L=KMAX
    2 ANS=YTAB(L-1)+(X-XTAB(L-1))*(YTAB(L)-YTAB(L-1))/
     1(XTAB(L)-XTAB(L-1))
      TABLE=ANS
      RETURN
C
    6 TABLE=YTAB(1)
      RETURN
      END
```

```
CTEQUAT
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE TEQUAT(DERIV)
C
      REAL*8 ISP,M
C
      DIMENSION DERIV(16),CMAT(3,3),TTMAT(3),TBMAT(3),FBMAT(3),FTMAT(3)
      DIMENSION OCCUR(4000),NOCCUR(30)
C
      EQUIVALENCE
     1(OCCUR(001),AREF    ),(OCCUR(016),CD      ),(OCCUR(019),CAPG   ),
     2(OCCUR(026),GAME    ),(OCCUR(037),M       ),(OCCUR(051),QD     ),
     3(OCCUR(063),RE      ),(OCCUR(082),V       ),(OCCUR(091),Z)
      EQUIVALENCE
     1(OCCUR(003),ALPRIM),(OCCUR(044),PHI     ),(OCCUR(045),PSI     ),
     2(OCCUR(071),THEALP),(OCCUR(200),PSIALP),(OCCUR(203),CN      ),
     3(OCCUR(213),THINE ),(OCCUR(222),ISP     ),
     4(OCCUR(3565),TTMAT(1)),(OCCUR(3643),TBMAT(1))
C
      COMMON OCCUR,NOCCUR
C
C
C
C           SUBROUTINE TEQUAT CALCULATES THE DERIVATIVES OF THE PARTICLE
C        TRAJECTORY AND THRUSTING PARAMETERS.
C           THE FOLLOWING DERIVATIVES ARE CALCULATED IN THIS SUBROUTINE -
C           DERIV(1)=V DOT
C           DERIV(2)=GAME DOT
C           DERIV(3)=D(TIME)/D(ALTITUDE)
C           DERIV(4)=XR DOT
C           DERIV(14)=YR DOT
C           DERIV(15)=PSIALP DOT
C           DERIV(16)=WTH DOT
C
C
C           TRIGONOMETRIC FUNCTIONS OF THE EULER ANGLES.
      CTHEL=COS(THEALP)
      STHEL=SIN(THEALP)
      CPSI=COS(PSI)
      SPSI=SIN(PSI)
      CPHI=COS(PHI)
      SPHI=SIN(PHI)
C
C           COMPONENTS OF THE CONVERSION MATRIX WHICH TRANSFORMS FROM A
C        BODY ORIENTED TO A TRAJECTORY ORIENTED COORDINATE SYSTEM.
      CMAT(1,1)=CTHEL*CPSI
      CMAT(2,1)=CTHEL*SPSI
      CMAT(3,1)=-STHEL
      CMAT(1,2)=CPSI*STHEL*SPHI-SPSI*CPHI
      CMAT(2,2)=SPSI*STHEL*SPHI+CPSI*CPHI
      CMAT(3,2)=CTHEL*SPHI
      CMAT(1,3)=CPSI*STHEL*CPHI+SPSI*SPHI
      CMAT(2,3)=SPSI*STHEL*CPHI-CPSI*SPHI
      CMAT(3,3)=CTHEL*CPHI
C
C           RESOLVING THE THRUST VECTOR INTO TRAJECTORY COORDINATES.
      CALL MATMPY(TTMAT,CMAT,TBMAT)
      XRDOT=V*CMAT(1,1)
      YRDOT=V*CMAT(1,2)
```

```
      ZBDOT=V*CMAT(1,3)
C
C           NORMAL FORCE CALCULATION.
      FN=CN*QD*AREF
C           AXIAL FORCE COEFFICIENT.
      CX=(-CD&CN*SIN(ALPRIM))/COS(ALPRIM)
C
C           CALCULATION OF BODY FORCES.
      FBMAT(1)=QD*AREF*CX
      ROOT=SQRT(YBDOT**2&ZBDOT**2)
      IF(ROOT.LT.1.0D-10)GO TO 1
      FBMAT(2)=-FN*YBDOT/ROOT
      FBMAT(3)=-FN*ZBDOT/ROOT
      GO TO 2
    1 FBMAT(2)=0.0D0
      FBMAT(3)=0.0D0
C           RESOLVING THE BODY FORCES INTO TRAJECTORY COORDINATE SYSTEM.
    2 CALL MATMPY(FTMAT,CMAT,FBMAT)
C
C           CALCULATION OF DERIVATIVES.
      TEM=RE/(RE&Z)
      TEM2=CAPG*TEM*TEM
      SING=SIN(GAME)
      COSG=COS(GAME)
      DERIV(1)=-QD*CD*AREF/M-TEM2*SING&TTMAT(1)/M
      DERIV(2)=COSG*(V*V*TEM/RE-TEM2)/V-(FTMAT(3)&TTMAT(3))/(M*V)
      DERIV(3)=V*SING
      DERIV(4)=COS(PSIALP)*V*COSG*TEM
      DERIV(14)=SIN(PSIALP)*V*COSG*TEM
      DERIV(15)=-(FTMAT(2)&TTMAT(2))/(M*V*COSG)
      DERIV(16)=-THINT/ISP
      DERIV(3)=1.0D0/DERIV(3)
      RETURN
      END
```

```
CTIMERS                                                                 TIMER   1
      IMPLICIT REAL*8(A-H,O-Z)                                          TIMER   2
      SUBROUTINE TIMERS(N,TABLE)                                        TIMER   3
      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
      RETURN                                                            TIMER  10
      END                                                               TIMER  11
```

```
CTOMALO
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE TOMALO

      REAL*8 LAMDA,LA,MDOT,MC,MT,MN

      DIMENSION OCCUR(4000),NOCCUR(30),MDOT(4,8),XLA(8)
      DIMENSION MC(6)
C
      EQUIVALENCE
     1(OCCUR(008),COST  ),(OCCUR(032),LAMDA ),(OCCUR(033),LA    ),
     2(OCCUR(042),PI    ),(OCCUR(052),RN    ),(OCCUR(064),SINT  ),
     3(OCCUR(070),TANT  ),(OCCUR(086),WDOT  ),(OCCUR(091),Z     ),
     4(OCCUR(092),ZTR   ),(OCCUR(0815),XLA(1)),(OCCUR(2708),MDOT(1,1))
C
      COMMON OCCUR,NOCCUR
C
C
C
C
C         SUBROUTINE TOMALO INTEGRATES THE MASS LOSS RATE ALONG THE
C         BODY TO OBTAIN THE RATE OF CHANGE IN WEIGHT DUE TO ABLATION.
C
C         INTEGRATION ALONG SHARP CONE FOR LAMINAR FLOW.
      IF(LAMDA.GT.1.0D-3)GO TO 2
      IF(Z.GE.ZTR)MT=2.[illegible]618D0*SQRT(2.0D0)*TANT*MDOT(2,8)*LA**2/COST
C
C         INTEGRATION ALONG SHARP CONE FOR TURBULENT FLOW.
      IF(Z.LT.ZTR)MT=3.04D0*((2.0D0)**0.2D0)*TANT*MDOT(3,8)*LA**2/COST
      GO TO 3
C
    2 TEM=PI*LA*LA/(COST*COST)
      TEM1=RN*(1.D0-SINT)/LA
      TEM2=0.5D0*SINT
      TEM3=(.766D0-SINT)/.234D0
C
C         BLUNT CONE LAMINAR MASS LOSS INTEGRATION.
      IF(Z.LT.ZTR)GO TO 4
C         INTEGRATION AROUND THE NOSECAP.
      MN=PI*RN*(1.D0-SINT)*(MDOT(1,1)&MDOT(2,1))*RN
C         INTEGRATION ALONG CONICAL FRUSTUM USING TRAPEZOIDAL RULE.
      DO 5 I=1,6
      MC(I)=TEM*(XLA(I&1)-XLA(I))*(TEM1&TEM2*(XLA(I&1)&XLA(I)))
    5 MC(I)=MC(I)*(MDOT(2,I)&MDOT(2,I&1))
      GO TO 7
C
C
C         BLUNT CONE TURBULENT MASS LOSS INTEGRATION.
C         INTEGRATION AROUND THE NOSECAP.
    4 MN=0.785D0*RN*RN*(MDOT(1,1)&MDOT(4,1)&(MDOT(4,1)&MDOT(3,1))
     1*TEM3)
C         INTEGRATION ALONG CONICAL FRUSTUM USING TRAPEZOIDAL RULE.
      DO 6 I=1,6
      MC(I)=TEM*(XLA(I&1)-XLA(I))*(TEM1&TEM2*(XLA(I&1)&XLA(I)))
    6 MC(I)=MC(I)*(MDOT(3,I)&MDOT(3,I&1))
C
C
C         SUMMATION OF CONTRIBUTIONS FROM NOSECAP AND CONICAL FRUSTUM.
    7 MT=MN
```

```
      DO 8 I=1,4                                                      TOMAL 60
    8 MT=MTEMC(I)                                                     TOMAL 61
    9 WDOT= MT                                                        TOMAL 62
      RETURN                                                          TOMAL 63
      END                                                             TOMAL 64
```

```
CVIXEN
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE VIXEN
C
      REAL*4        TIXM,TIXT,TIXZ,T1,T2,T3,T4,
     1T5,T6,T7,T8,T9,T10,T11,T12,T13,T14,T15,T16,T17,T18,T19,T20,T21,T22
      REAL * 4  T23, T24, T25, T26, T27, T28
      REAL*8 LAMDA,LA,MINF,MDOT,LAMSTO
      COMMON /CPCCUR/ PCCUR(1177C)
      COMMON /CICCUR/ IOCCUR(320)
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      COMMON OCCUR,NOCCUR
C
      DIMENSION  A(514),COEINF(2,2,2),QDOT(4,8),MDOT(4,8),PERSU(8),
     1           OCCUR(4000),NOCCUR(30),CDOWN(16),TTMAT(3),TITLE(2),
     2           ALMAX(200),ALMIN(200),TAMAX(200),TAMIN(200),
     3            ZMAX(200), ZMIN(200), EMAX(200), EMIN(200)
      DIMENSION DVALUE(16),DERIV(16),LPBND(16),DNBND(16),CHIGH(16)
      DIMENSION QINT(8),QLAST(8)
      DIMENSION NPLOT(5)
      DIMENSION  WTZ(75),WTMINF(75),WTRINF(75),WTPTOT(75),TRAJT(75),
     1           TRAJZ(75),TRAJV(75),TRAJW(75),TRAJRN(75),TRJALP(75),
     2           ALPTAB(75)
      DIMENSION          TIXM(10),TIXT(10),TIXZ(10),T1(10),T2(10),
     1T3(10),T4(10),T5(10),T6(10),T7(10),T8(10),T9(10),T10(10),T11(10),
     1T12(10),T13(10),T14(10),T15(10),T16(10),T17(10),T18(10),T19(10),
     1T20(10),T21(10),T22(10),
     3 ZPLOT(160), TPLOT(160),PLMINF(160),CDPLOT(160),CDEPLT(160),
     4CDBPLT(160),CDIPLT(160),WTOTPL(160),BETAPL(160), VPLOT(160),
     5VCGPLT(160), QPLOT(160),PSPLOT(160),QSRTPL(160),CD11PL(160),
     6D41PL(160),PE1PLT(160),PE7PLT(160),PE8PLT(160),CD7PLT(160),
     7 QB8PLT(160),ALPLOT(160),OCPLOT(160)
      DIMENSION WR1P(160), WR2P(160), WR3P(160)
      DIMENSION  WL1P(160), WL2P(160),WL3P(160)
      DIMENSION T23(10), T24(10), T25(10), T26(10), T27(10), T28(10)
      DIMENSION IOP(50)
C
      EQUIVALENCE
     1(OCCUR(001),APEE   ),(OCCUR(002),ALPHA ),(OCCUR(003),ALPRIM),
     2(OCCUR(010),CAPL   ),(OCCUR(016),CD    ),(OCCUR(020),CMQ   ),
     3(OCCUR(026),GAMF   ),(OCCUR(027),G     )
      EQUIVALENCE
     1(OCCUR(029),HSRTC  ),(OCCUR(032),LAMDA ),(OCCUR(033),LA       ),
     2(OCCUR(035),MINF   ),(OCCUR(042),PI    ),(OCCUR(043),P        ),
     3(OCCUR(044),PHI    ),(OCCUR(045),PSI   ),(OCCUR(047),PS       ),
     4(OCCUR(050),Q      ),(OCCUR(051),QD    ),(OCCUR(052),RN       )
      EQUIVALENCE
     1(OCCUR(062),REYL   ),(OCCUR(065),SMR   ),
     2(OCCUR(069),THETAD),(OCCUR(071),THEALP),(OCCUR(075),T        ),
     3(OCCUR(076),THETA ),(OCCUR(080),TIMER ),(OCCUR(081),SMF      )
      EQUIVALENCE
     1(OCCUR(082),V      ),(OCCUR(087),XR    ),
     2(OCCUR(090),XBAR   ),(OCCUR(091),Z     ),(OCCUR(092),ZTK   ),
     3(OCCUR(095),GRATE  ),(OCCUR(096),ALPENV),(OCCUR(097),DELW  ),
     4(OCCUR(098),CDP    ),(OCCUR(099),CDB   ),(OCCUR(100),CDI   ),
     5(OCCUR(101),CDPC   ),(OCCUR(102),TU    ),(OCCUR(105),GAMFU ),
     6(OCCUR(106),VU     ),(OCCUR(107),XRO   ),(OCCUR(108),ZC    )
      EQUIVALENCE
     1(OCCUR(109),PU     ),(OCCUR(110),QU    ),(OCCUR(111),SMRU  ),
```

NOT REPRODUCIBLE

```
     2(OCCUR(112),PHIU   ),(OCCUR(113),THEALO),(OCCUR(114),PSIU   ),       VIXFN  60
     3(OCCUR(118),ZPR1   ),(OCCUR(119),ZPR2   ),(OCCUR(120),ZBAR   ),       VIXFN  61
     4(OCCUR(121),ZST    ),(OCCUR(122),ALST   ),(OCCUR(123),TST    ),       VIXFN  62
     5(OCCUR(126),XBAR1 ),                                                  VIXFN  63
     6(OCCUR(127),DATE   ),(OCCUR(128),CASE   ),(OCCUR(129),EMC    ),       VIXFN  64
     7(OCCUR(130),ALWIG2),(OCCUR(131),ALBARP),(OCCUR(133),W1     ),         VIXFN  65
     8(OCCUR(107),DELIN )                                                   VIXFN  66
      EQUIVALENCE                                                           VIXFN  67
     1(OCCUR(199),YK     ),(OCCUR(200),PSIALP),(OCCUR(201),TH     ),        VIXFN  68
     2(OCCUR(202),CM     ),(OCCUR(203),CN     ),(OCCUR(226),DELW2 )         VIXFN  69
      EQUIVALENCE                                                           VIXFN  70
     1(OCCUR(227),DELW3 ),(OCCUR(228),WTOTAL),(OCCUR(230),BETAP ),          VIXFN  71
     2(OCCUR(231),BETA   ),(OCCUR(232),V2GDOT),(OCCUR(233),DOW    ),        VIXFN  72
     3(OCCUR(234),DELCDP),(OCCUR(235),DECDFP),(OCCUR(236),DELFTC),          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(0301),      A(1)),(OCCUR(1044), ALMAX(1)),                     VIXFN  77
     2(OCCUR(1244), ALMIN(1)),(OCCUR(1444), TAMAX(1)),                      VIXFN  78
     3(OCCUR(1644), TAMIN(1)),(OCCUR(1344),  ZMAX(1)),                      VIXFN  79
     4(OCCUR(2044),  ZMIN(1)),(OCCUR(2244),  EMAX(1)),                      VIXFN  80
     5(OCCUR(2444),  EMIN(1))                                               VIXFN  81
      EQUIVALENCE                                                           VIXFN  82
     1(OCCUR(1044),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),TRAJZ(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),QDOT(1,1)),(OCCUR(2709),MDOT(1,1)),                      VIXFN  89
     2(OCCUR(2793),COFINF(1,1,1)),(OCCUR(2901),PERSB(1)),                   VIXFN  90
     3(OCCUR(3233),HTAB(1))                                                 VIXFN  91
      EQUIVALENCE                                                           VIXFN  92
     1(OCCUR(3533), CHIGH(1)),(OCCUR(3549), CDOWN(1)),                      VIXFN  93
     2(OCCUR(3565), TTMAT(1)),(OCCUR(3646),ALPTAB(1))                       VIXFN  94
      EQUIVALENCE                                                           VIXFN  95
     1(NOCCUR(01),JJHOLD),(NOCCUR(03),MOPT  ),                              VIXFN  96
     2(NOCCUR(06),MAXVAL),(NOCCUR(07),LOPT  ),(NOCCUR(10),MHEAT ),          VIXFN  97
     X(NOCCUR(11),IKMAX ),(NOCCUR(12),IKMIN ),                              VIXFN  98
     3(NOCCUR(14),NPRINT),(NOCCUR(24),NPLOT(1)),(NOCCUR(29),ITAPE),         VIXFN  99
     4(NOCCUR(30),INALPH)                                                   VIXFN 100
      EQUIVALENCE (NOCCUR(0000), IATMOS)                                    VIXFN 101
      EQUIVALENCE(OCCUR(4000),LP)                                           VIXFN 102
      EQUIVALENCE (IOCCUR(00001), IOP    )                                  VIXFN 103
       EQUIVALENCE (IOCCUR(00314), IOBL   )                                 VIXFN 104
      EQUIVALENCE (IOCCUR(00302), LPLOT )                                   VIXFN 105
      EQUIVALENCE (PCCUR(00641), BETAPL)                                    VIXFN 106
      EQUIVALENCE (PCCUR(00001), TPLOT )                                    VIXFN 107
      EQUIVALENCE (PCCUR(00481), V2GPLT)                                    VIXFN 108
      EQUIVALENCE (PCCUR(00321), VPLOT )                                    VIXFN 109
      EQUIVALENCE (PCCUR(01281), WL1P   )                                   VIXFN 110
      EQUIVALENCE (PCCUR(01441), WL2P   )                                   VIXFN 111
      EQUIVALENCE (PCCUR(01601), WL3P   )                                   VIXFN 112
      EQUIVALENCE (PCCUR(00801), WR1P   )                                   VIXFN 113
      EQUIVALENCE (PCCUR(00961), WR2P   )                                   VIXFN 114
      EQUIVALENCE (PCCUR(01121), WR3P   )                                   VIXFN 115
      EQUIVALENCE (PCCUR(00161), ZPLOT )                                    VIXFN 116
C                                                                           VIXFN 117
C                                                                           VIXFN 118
C                                                                           VIXFN 119
```

```
C
C         SUBROUTINE VIXEN PERFORMS THE FOLLOWING FUNCTIONS  -
C       SETS INITIAL VALUES OF CERTAIN QUANTITIES.
C       CALLS IN SUBROUTINE CHNTBL, WHICH ASSIGNS APPROPRIATE GEOMETRIC
C       AND MATERIAL PROPERTIES.
C       ASSIGNS THE ACCURACY LIMITS FOR INTEGRATION.
C       CALLS IN THE INTEGRATION SUBROUTINE, ADMARK, FOR THESE OPTIONS
C       WHERE A TRAJECTORY IS BEING CALCULATED,  OR SETS PROGRAM
C       VARIABLES EQUAL TO TABULAR INPUT VALUES OF THE WIND TUNNEL OR
C       INPUT TRAJECTORY OPTIONS, AS WELL AS CALLING IN THE SUBROUTINES
C       WHICH PERFORM CALCULATIONS NECESSARY TO OBTAIN DRAG VALUES
C       WHEN NO INTEGRATION IS DONE.
C       CALCULATES THE MAXIMUMS AND MINIMUMS IN ANGLE OF ATTACK WHEN
C       THE THREE DEGREE OF ROTATIONAL FREEDOM OPTION IS USED.
C       CONTROLS THE PRINTOUT.
C       CALLS IN THE TAPE AND THE PLOTTING SUBROUTINES.
C
C
C
C         TITLES FOR THE PLOTTING OPTIONS
      DATA TITLE(1)/4HTURB/
      DATA TITLE(2)/4H LAM/
      DATA TIXT/40H                   TIME IN SECONDS      /
      DATA TIXZ/40H                 ALTITUDE IN FEET       /
      DATA TIXM/40H                         MINF           /
      DATA  T1/40H                       CD TOTAL          /
      DATA  T2/40H                       CDB & CDP         /
      DATA T3/40H       CDFINF(BLUNT WITH BLOWING)         /
      DATA  T4/40H                         CDI             /
      DATA  T5/40H                   WTOTAL/WI             /
      DATA  T6/40H                        BETA             /
      DATA  T7/40H                VELOCITY(FT/SEC)         /
      DATA  T8/40H                        MINF             /
      DATA  T9/40H                       VDOT/G            /
      DATA T10/40H                  ALTITUDE(FT)           /
      DATA T11/40H          DYNAMIC PRESSURE(LB/FT2)/
      DATA T12/40H                    PS(ATMOS)            /
      DATA T13/40H                      HSRTC              /
      DATA T14/40H            QDOT(STAG)(BTU/FT2-SEC) /
      DATA T15/40H           QDOT(SONIC)(BTU/FT2-SEC) /
      DATA T16/40H               QDOT(7)(BTU/FT2-SEC) /
      DATA T17/40H               QDOT(8)(BTU/FT2-SEC) /
      DATA T18/40H                 PEPSB(TANG. PT)     /
      DATA T19/40H                     PEPSB(7)        /
      DATA T20/40H                     PEPSB(8)        /
      DATA T21/40H            ALPHA PRIME CAV(DEGREES)/
      DATA T22/40H                    CCCUR(NPLCT5)     /
      DATA T23/40H          WAKE  L1                   /
      DATA T24/40H          WAKE  L2                   /
      DATA T25/40H          WAKE  L3                   /
      DATA T26/40H          WAKE  R1                   /
      DATA T27/40H          WAKE  R2                   /
      DATA T28/40H          WAKE  R3                   /
C
C
C         SETTING INITIAL VALUES
      NK=0
    1 V=V0
      GAME=GAMEO
      TIME=T0
```

NOT REPRODUCIBLE

```
      Z=ZO
      XR=XRO
      P=PO
      Q=QO
      SMR=SMRO
      PSI=PSIO
      THEALP=THEALO
      PHI=PHIO
      YR=0.0D0
      PSIALP=0.0D0
      ZUST=Z
      CVALUE(1)=V
      CVALUE(2)=GAMF
      CVALUE(3)=TIME
      CVALUE(4)=XR
      CVALUE(8)=PSI
      CVALUE(9)=THEALP
      CVALUE(10)=PHI
      CVALUE(11)=Q
      CVALUE(12)=SMR
      CVALUE(13)=P
      CVALUE(14)=YR
      CVALUE(15)=PSIALP
      BETA=0.D0
      BETAP=0.D0
      DELBET=0.D0
      T=0.D0
      SMF=0.0D0
      ALPRIM=0.0D0
      ALPHA=0.0D0
      ALPENV=0.0D0
      IALP=1
      IKMAX=0
      IKMIN=0
      PRINTZ=0.D0
      ZSTO=0.D0
      DELPR1=ZPR1
      FREQ=-DELPR1
      DEL=-ZPR1/5.0D0
      LPLOT=0
      NNTAPE=0
C
C          SELECTION OF DELTA OF INTEGRATION
      IF(DABS(DELIN).LT.DABS(DEL))DEL=DELIN
C
C          SUBROUTINE CHNTOL ASSIGNS THE APPROPRIATE GEOMETRIC PARA-
C       METERS AND MATERIAL PROPERTIES AT ENTRY AND WHEN CONFIGURATION
C       CHANGES AT ALTITUDE ZTURN.
      LCHNGE=1
   61 CALL CHNTRL(DVALUE,ZTURNX,LCHNGE)
      L =1
      IF(JJHOLD.NE.(-1))GO TO 50
      LP=6
      WRITE(6,1005)
      GO TO 100
   50 IF(ZUST.GT.ZTURNX)GO TO 61
      LCHNGE=2
      WRITE(6,1085)Z
 1085 FORMAT(1H0,20X,'CONFIGURATION IS CHANGING DISCONTINUOUSLY AT Z=',
     1E15.6 )
```

```
      GO TO 60
   61 CONTINUE
C
C           ASSIGNING OF BOUNDS ON THE ABSOLUTE DIFFERENCE BETWEEN THE
C         EXTRAPOLATED AND INTERPOLATED VALUES IN THE INTEGRATION
      DO 3 J=1,16
      TEM=ABS(DVALUE(J))
      IF(TEM.LE.1.D0)GO TO 4
      UPBND(J)=CHIGH(J)*TEM
      DNBND(J)=CDOWN(J)*TEM
      GO TO 3
    4 UPBND(J)=CHIGH(J)
      DNBND(J)=CDOWN(J)
    3 CONTINUE
C
      IF(LOPT.LT.3)GO TO 31
      IF(LOPT.EQ.4)GO TO 41
C
C           ASSIGNING TABULAR TRAJECTORY INPUT TO INTERNAL PROGRAM
C         SYMBOLS.
      TIMER=TRAJT(NK&1)
      TINT=TIMER
      TO=TRAJT(1)
      Z=TRAJZ(NK&1)
      V=TRAJV(NK&1)
      IF(V.NE.0.D0)GO TO 1016
      WRITE(6,1015)
 1015 FORMAT(1H0,10X,'*****WARNING***** CANNOT ACCEPT A ZERO VELOCITY,
     1GOING TO NEXT CASE')
      RETURN
 1016 CONTINUE
      WTOTAL=TRAJW(NK&1)
      RN=TRAJRN(NK&1) /12.D0
      ALPHA=TRJALP(NK&1) /57.2957800
      ALPRIM=ALPHA
      INALPH=1
      DO 74 KK=1,8
      RDOT(2,KK)=0.D0
      RDOT(3,KK)=0.D0
      QDOT(2,KK)=0.D0
   74 QDOT(3,KK)=0.D0
      RDOT(1,1)=0.D0
      RDOT(4,1)=0.D0
      QDOT(1,1)=0.D0
      QDOT(4,1)=0.D0
      GO TO 40
C
C           ASSIGNING TABULAR WIND TUNNEL INPUTS TO INTERNAL PROGRAM
C         SYMBOLS.
   41 CONTINUE
      Z=WTZ(NK&1)
      MINF=WTMINF(NK&1)
      IF(MINF.GE. 5.D0)GO TO 1046
      WRITE(6,1045)
 1045 FORMAT(1H1,10X,'*****WARNING***** CANNOT ACCEPT A MACH NUMBER LESS
     1 THAN 5,  GOING TO NEXT CASE')
      RETURN
 1046 CONTINUE
      REYINF=WTPINF(NK&1)
      PTOTAL=WTPTOT(NK&1)
```

```
      TIMER=NK
      TIME=TIMER
      TC=0.0D0
      INALPH=1
      DO 72 KK=1,8
      MDOT(2,KK)=0.0D0
      MDOT(3,KK)=0.0D0
      QDOT(2,KK)=0.0D0
   72 QDOT(3,KK)=0.0D0
      MDOT(1,1)=0.0D0
      MDOT(4,1)=0.0D0
      QDOT(1,1)=0.0D0
      QDOT(4,1)=0.0D0
C           INPUT ALPHA TO BE USED WITH WIND TUNNEL CONDITIONS.
      ALPHA=ALPTAB(NK&1)/57.29578DC
      ALPRIM=ALPHA
C
C           WHEN TRAJECTORY INPUT OR WIND TUNNEL INPUT TABLES ARE USED
C        THE INTEGRATION SUBROUTINE IS BYPASSED AND PRELIMINARY
C        CALCULATIONS, AERODYNAMIC HEATING, MASS LOSS, AND DRAG
C        CALCULATIONS ARE DONE FOR THE INPUT ALTITUDES ONLY.
   44 CONTINUE
      LL=1
      CALL PRELIM(LL)
      IF(MINF.LT.5.0D0)WRITE(6,32)
   32 FORMAT(1H0,25X,57H***WARNING*** PROGRAM IS NOT VALID FOR MINF LESS
     1 THAN 5.0)
      IF(MOPT.EQ.0) GO TO 39
      IF(XBAR.GT.XUP)GO TO 39
      CALL AERODY
      IF(HEAT.EQ.0) GO TO 39
      CALL MASSLO
      CALL TOMALO
   39 CONTINUE
      CALL DRAGCO
      GO TO 101
C
C
C           VARIABLES ARE INTEGRATED AND THE TRAJECTORY CALCULATED WHEN
C        LOPT IS LESS THAN OR EQUAL TO 2.  THE QUANTITIES BEING
C        INTEGRATED IN ADM4RK ARE AS FOLLOWS, WHERE DVALUE IS THE
C        INTEGRATED VALUE AND DERIV IS THE CORRESPONDING DERIVATIVE
C        WITH ALTITUDE.
C           DVALUE(1)=V
C           DVALUE(2)=GAMF
C           DVALUE(3)=TIME
C           DVALUE(4)=XR
C           DVALUE(5)=W
C           DVALUE(6)=RN
C           DVALUE(7)=RB
C           DVALUE(8)=PSI
C           DVALUE(9)=THEALP
C           DVALUE(10)=PHI
C           DVALUE(11)=Q
C           DVALUE(12)=SHR
C           DVALUE(13)=P
C           DVALUE(14)=YR
C           DVALUE(15)=PSIALP
C           DVALUE(16)=WTH
   11 CONTINUE
```

```
      CALL ADMARK(16,DEL,DVALUE,DERIV,UPBND,DNBND,0.3D0,FREQ   ,ZST,LP,
     1ZUSE,1.D0)
C
      FINF=DVALUE(3)
      Z=ZUSE
      IF(LP.NE.6)GO TO 7
      WRITE(6,1005)
 1005 FORMAT(1H0 10X,34HPROGRAM ERROR. GOING TO NEXT CASE.)
      GO TO 100
C
C           FOR THE LOPT=0 ROTATIONAL TRAJECTORY TESTS ARE DONE TO OBTAIN
C         MAXIMUMS AND MINIMUMS IN ALPRIM.
    7 IF(LOPT.NE.0)GO TO 30
      IF(IKMAX.EQ.200)GO TO 51
      IF(IKMIN.EQ.200)GO TO 51
      GO TO(8,9,10),IALP
    8 IALP=2
      SUMA2=0.D0
      SUMA3=0.D0
      ALSTO1=ALPRIM
      TSTO1=TIME
      ZSTO1=Z
      FSTO1=SMF
      GO TO 20
    9 IALP=3
      ALSTO2=ALPRIM
      TSTO2=TIME
      ZSTO2=Z
      FSTO2=SMF
      GO TO 20
   10 TEMMAX=ALSTO1
      IF(TEMMAX.LT.ALSTO2)TEMMAX=ALSTO2
      IF(TEMMAX.LT.ALPRIM)TEMMAX=ALPRIM
      TEMMIN=ALSTO1
      IF(TEMMIN.GT.ALSTO2)TEMMIN=ALSTO2
      IF(TEMMIN.GT.ALPRIM)TEMMIN=ALPRIM
      IF(IKMAX.EQ.0)GO TO 21
      TEMA=ALSTO1&ALSTO2
      TEMT=TSTO2-TSTO1
      SUMA2=SUMA2&0.5D0*TEMA*TEMT
      SUMA3=SUMA3&0.5D0*TEMA*TEMA*TEMT
   21 CONTINUE
      ALSTO1=ALSTO2
      ALSTO2=ALPRIM
      TSTO1=TSTO2
      TSTO2=TIME
      ZSTO1=ZSTO2
      ZSTO2=Z
      FSTO1=FSTO2
      FSTO2=SMF
      IF(TEMMAX.EQ.ALSTO1)GO TO 11
      IF(TEMMIN.EQ.ALSTO1)GO TO 12
      GO TO 20
   11 IKMAX=IKMAX&1
      IF(IKMAX.EQ.200)WRITE(6,2000)Z
 2000 FORMAT(1H0 20X,46H*****WARNING***** ALPHA MAX TABLE FILLED AT Z=,
     1F12.6,44H NO MORE MAXIMUMS OR MINIMUMS WILL BE STORED)
      IF(IKMAX-2)13,14,15
   13 ALMIC1=ALSTO1
      TMIC1=TSTO1
```

```
      ALMAX(IKMAX)=ALWIG1
      TAMAX(IKMAX)=TWIG1
      FMAX(IKMAX)=FSTO1
      ZMAX(IKMAX)=ZSTO1
      GO TO 20
   14 ALWIG2=ALSTO1
      TWIG2=TSTO1
      ALMAX(IKMAX)=ALWIG2
      TAMAX(IKMAX)=TWIG2
      FMAX(IKMAX)=FSTO1
      ZMAX(IKMAX)=ZSTO1
      GO TO 16
   15 ALWIG1=ALWIG2
      ALWIG2=ALSTO1
      TWIG1=TWIG2
      TWIG2=TSTO1
      ALMAX(IKMAX)=ALWIG2
      TAMAX(IKMAX)=TWIG2
      ZMAX(IKMAX)=ZSTO1
      FMAX(IKMAX)=FSTO1
      GO TO 16
   12 TBARP=TSTO1
      ALBARP=ALSTO1
      IKMIN=IKMIN&1
      IF(IKMIN.EQ.200)WRITE(6,2001)Z
 2001 FORMAT(1H0 20X,46H*****WARNING***** ALPHA MIN TABLE FILLED AT Z=,
     1F13.6,44H NO MORE MINIMUMS OR MAXIMUMS WILL BE STORED)
      ALMIN(IKMIN)=ALBARP
      TAMIN(IKMIN)=TBARP
      FMIN(IKMIN)=FSTO1
      ZMIN(IKMIN)=ZSTO1
      GO TO 20
   16 T=TWIG2-TWIG1
      SMF=1.D0/T
      ABAR2=0.D0
      ABAR3=0.D0
      DO 22 I=1,3
      II=I-1
      TEMI=THETAD**II
      ISUB11=120*II
      ISUB12=ISUB11&1
      DO 22 J=1,3
      JJ=J-1
      TEMJ=1.D0
      IF(JJ.NE.0)TEMJ=LAMSTO**JJ
      ISUB1=ISUB11&9*JJ
      ISUB2=ISUB12&9*JJ
      ABAR2=ABAR2&A(ISUB1)*TEMI*TEMJ
   22 ABAR3=ABAR3&A(ISUB2)*TEMI*TEMJ
      TEMP1=180.D0/PI
      DPATE=TEMP1*(ABAR2*SUMA2&TEMP1*ABAR3*SUMA3)
      SUMA2=0.D0
      SUMA3=0.D0
      IF(ALWIG2.GE.ALST)GO TO 20
      LOPT=1
      LP=1
      T=0.D0
      ALPHA=0.D0
      ALPH14=0.D0
      GO TO 20
```

```
2.  LAMSTO=LAMDA
    ALPENV=ALPRIM
    IF(T.EQ.0.D0)GO TO 30
    ALPENV=ALWIG1&(TWIG1-TIME)*(ALWIG2-ALWIG1)/(TWIG1-TWIG2)
 30 CONTINUE
C
C      FOR THE SIMPLIFIED ANGLE OF ATTACK CALCULATION, LOPT=2, ALPHA
C   ENVELOPE IS SET EQUAL TO INSTANTANEOUS VALUE OF ALPHA.
    IF(LOPT.NE.2)GO TO 91
    ALPENV=ALPRIM
 91 CONTINUE
    IF(LOPT.EQ.1)ALPENV=ALPRIM
C
C      BETA, VDOT/G, AND D(BETA)/DZ CALCULATION
    DOW=QD*AREF*CD/WTOTAL
    BETAST=BETA
    BETA=WTOTAL/(CD*AREF)
    IF(ICOM(10).EQ.1) BETA = WTOTAL/(CD*AREF-TH/QD)
    IF(ICOM(10).EQ.2)BETA=1.C/(WTOTAL/(CD*AREF-TH/QD))
    DENOM=Z-ZSTO
    IF(DENOM.EQ.0.D0)DENOM=1.0D-10
    BETAP=(BETA-BETAST)/DENOM
    ZSTO=Z
    VGDOT=DERIV(1)/(G*DERIV(3))
C
C      TESTS FOR PRINTING IF LOPT IS LESS THAN OR EQUAL TO 2
    IF(Z.LE.(ZSTG1.DDC))GO TO 54
    IF(ABS(PRINTZ-Z).LT.DELPPI)GO TO 50
 54 PRINTZ=Z
    IF(PRINTZ.LE.(ZBAREC.1DC))DELPPI=ZPP2
    FREQ=-DELPPI
C
C      SETTING PRINTOUT VALUES FOR LOPT=3 OR 4 OPTIONS
101 CONTINUE
    IF(LOPT.LT.3)GO TO 10C
    BETA=WTOTAL/(CD*AREF)
    VGDOT=0.0D0
    CVALUE(1)=V
    CVALUE(4)=0.0D0
    CVALUE(6)=RN
    CVALUE(13)=0.0D0
    PSIALP=0.0D0
    GO TO 80
C
C      STORAGE FOR THE TAPE OPTION
10. CONTINUE
    NNTAPE=NNTAPE&1
C
C      STORAGE OF VALUES FOR PLOTTING
    IF(LPLOT.EQ.160)GO TO 80
    LPLOT=LPLOT&1
    IF(LPLOT.EQ.160)WRITE(6,1080)Z,TIME
108 FORMAT(1H0,10X,47H*****WARNING***** PLOTS HAVE BEEN CUT OFF AT Z=,
   1E13.6,10H AND TIME=F7.2)
    ZPLOT(LPLOT)=Z
    TPLOT(LPLOT)=TIME
    PLMINF(LPLOT)=MINF
    CDPLOT(LPLOT)=CD
    CDPPLT(LPLOT)=CDB&CDP
    IF(Z.GE.ZTF)CDPPLT(LPLOT)=CDFINF(2,2,1)
```

```
      IF(Z.LT.ZTR)CDFPLT(LPLOT)=CDFINF(2,1,1)
      CDIPLT(LPLOT)=CDI
      WTOTPL(LPLOT)=WTOTAL/W1
      BETAPL(LPLOT)=BETA
      VPLOT(LPLOT)=DVALUE(1)
      VCGPLT(LPLOT)=VCGDOT
      QPLOT(LPLOT)=QD
      PSPLOT(LPLOT)=PS/2116.0D0
      HSFTPL(LPLOT)=HSFTC
      QD11PL(LPLOT)=QDOT(1,1)
      QD41PL(LPLOT)=QDOT(4,1)
      PE1PLT(LPLOT)=PEPSB(1)
      PE7PLT(LPLOT)=PEPSB(7)
      PE8PLT(LPLOT)=PEPSB(8)
      IF(Z.GE.ZTR)J=2
      IF(Z.LT.ZTR)J=3
      QD7PLT(LPLOT)=QDOT(J,7)
      QD8PLT(LPLOT)=QDOT(J,8)
      ALPLOT(LPLOT)=57.29578D0*ALPENV
      IF(NPLOT(5).LE.0)GO TO 80
      IPLOT=NPLOT(5)
      UCPLOT(LPLOT)=DCCLR(IPLOT)
   80 CONTINUE
C
C
      IF(IOP(74).EQ.1) CALL WAKE
C
C
C          IF NPRINT = 0, PRINTOUT IS OMITTED BUT TAPES AND PLOTS MAY
C        STILL BEING GENERATED.
      IF(NPRINT.EQ.0)GO TO 50
C
C
C          BEGINNING OF PRINTOUT
C
C
C          PRINTOUT OF TRANSLATIONAL QUANTITIES
C          ALL ANGLES ARE CHANGED INTO UNITS OF DEGREES BEFORE BEING
C        PRINTED OUT.
      PSALLP=57.29578D0*DVALUE(15)
      GAMFP=57.29578D0*DVALUE(2)
      PSPG=PS/2116.0D0
      WRITE(6,1032)TIME,Z          ,DVALUE(1),GAMFP,DVALUE(4),BETA,ZTR,
     1QD,MINF,VCGDOT,BETAP,TH,TIMAT(1),DVALUE(14),PSALLP,DCW
 1032 FORMAT(1H0///,54X,24HTRANSLATIONAL QUANTITIES//10X,5HTIME=F7.2,
     15H  Z=E15.6,5H   V=E15.7,8H   GAMF=E12.4,6H   XR=E15.6,8H   BETA=
     2E12.4/25X,4HZTR=E13.6,6H   QD=F14.4,1H   MINF=E12.4,10H   VDOT,G=E
     311.4,9H   BETAP=F11.4/25X,3HTH=E14.4,7H   TXT=E13.4,6H   YR=E14.4,
     410H   PSIALP=E11.4,7H   D/W=F13.4)
      IF(IATMOS.EQ.0)GO TO 75
      IF(Z.GE.UPBNDZ)GO TO 75
      IF(Z.LT.DNBNDZ)GO TO 75
      WRITE(6,1075)UPBNDZ,DNBNDZ
 1075 FORMAT(1H0,20X,*INPUT ATMOSPHERE BEING USED WHEN Z LESS THAN *,
     1E14.6,*, GREATER THAN OR EQUAL TO*,E14.6/)
   75 CONTINUE
C
C          UPAC COEFFICIENT PRINTOUT
      IF(Z.LT.ZTR)J=1
      IF(Z.GE.ZTR)J=2
```

```
      IF(J.EQ.1)JJ=3                                                    VIXFN600
      IF(J.EQ.2)JJ=2                                                    VIXFN601
      REYLA=(REYL*LA)/CAPL                                              VIXFN602
      WRITE(6,1034)CD,CDP,TITLE(J),CDFINF(2,J,1),CDB,CDPU,TITLE(J),     VIXFN603
     1CDFINF(2,J,2),CDI,XBAR,REYLA,XBAR1                                VIXFN604
 1034 FORMAT(1H0 58X,15HDRAG QUANTITIES//25X,3HCD=E14.4,7H    CDP=E13.4, VIXFN605
     113H    CDFINF(BL,A4,5H,WB)=E12.4,10X,4HCDB=E13.4/45X,5HCDPU=E12.4, VIXFN606
     213H    CDFINF(BL,A4,5H,NB)=E12.4,10X,4HCDI=E13.4/45X,5HXBAR=E12.4, VIXFN607
     312H    REYINFLA=,10X,E12.4,10X,6HXBAR1=E11.4)                     VIXFN608
      IF(J.EQ.1)GO TO 1040                                              VIXFN609
      WRITE(6,1039) DELCDP,DECDFP,DECFTC                                VIXFN610
 1039 FORMAT(54X,22HLAMINAR CDI COMPONENTS/45X,6HCDI/P=F11.4,10H   CDI/SVIXFN611
     1F=,12X,E12.4,10X,7HCDI/TC=E11.4)                                  VIXFN612
 1040 CONTINUE                                                          VIXFN613
C                                                                       VIXFN614
C           PRINTOUT OF CONFIGURATION PARAMETERS                        VIXFN615
      LA=LA*12.0D0                                                      VIXFN616
      DVALUE(6)=DVALUE(6)*12.0D0                                        VIXFN617
      WRITE(6,1036)DVALUE(6),THETAD,LA,LAMDA,AREF,WTCTAL,DELW,DELW2,    VIXFN618
     1DELW3                                                             VIXFN619
 1036 FORMAT(1H0 55X,13HCONFIGURATION//25X,3HRN=E14.4,9H    THETA=E11.4, VIXFN620
     116H    LA=E14.4,10H    LAMBDA=F11.4,8H    AREF=E12.4/25X,2HW=E15.4, VIXFN621
     228H    DELW=E12.4,8H    WABL=F12.4,10H    WTHRST=E11.4)           VIXFN622
      LA=LA/12.0D0                                                      VIXFN623
      DVALUE(6)=DVALUE(6)/12.0D0                                        VIXFN624
C                                                                       VIXFN625
C           IF THE MASS LOSS OPTIONS ARE NOT BEING USED, THE PRINTOUT IS VIXFN626
C        BYPASSED.                                                      VIXFN627
      IF(MOPT.EQ.0)GO TO 88                                             VIXFN628
C                                                                       VIXFN629
C           HEATING AND MASS LOSS PRINTOUT                              VIXFN630
      IF(TIMER.EQ.TO)TLAST=0.0D0                                        VIXFN631
      DO 24 I=1,8                                                       VIXFN632
      IF(TLAST.EQ.0.0D0)GO TO 24                                        VIXFN633
      QINT(I)=QINT(I)+0.5D0*(QDOT(JJ,I)+QLAST(I))*(TIMER-TLAST)         VIXFN634
   24 QLAST(I)=QDOT(JJ,I)                                               VIXFN635
      IF(TLAST.EQ.0.0D0)GO TO 26                                        VIXFN636
      QINST=QINST+0.5D0*(QDOT(1,1)+QSTLST)*(TIMER-TLAST)                VIXFN637
      QINSON=QINSON+0.5D0*(QDOT(4,1)+QLTSON)*(TIMER-TLAST)              VIXFN638
      GO TO 28                                                          VIXFN639
   26 DO 27 I=1,8                                                       VIXFN640
   27 QINT(I)=0.0D0                                                     VIXFN641
      QINST=0.0D0                                                       VIXFN642
      QINSON=0.0D0                                                      VIXFN643
   28 QSTLST=QDOT(1,1)                                                  VIXFN644
      QLTSON=QDOT(4,1)                                                  VIXFN645
      TLAST=TIMER                                                       VIXFN646
      WRITE(6,1035)QDOT(1,1),QDOT(4,1),HSRTO,PSPO,(PERSB(LJ),LJ=1,8),   VIXFN647
     1(QDOT(JJ,LJ),LJ=1,8),(MDOT(JJ,LJ),LJ=1,8),(QINT(LJ),LJ=1,8),QINST,VIXFN648
     2QINSON                                                            VIXFN649
 1035 FORMAT(1H0 61X,9HMASS LOSS//22X,11HQDOT(STAG)=E11.4,7X,12HQDOT(SONVIXFN650
     1IC)=E11.4,7X,6HHSRTO=E11.4,7X,5HPSPO=E12.4/12X,46HX STATIONS ARE PVIXFN651
     2ERCENTAGES OF UNABLATED LENGTH/22X,9HTANG. PT.,5X,8HX/LA=0.2,5X,  VIXFN652
     38HX/LA=0.4,5X,8HX/LA=0.6,5X,9HX/LA=0.75,4X,8HX/LA=0.9,5X,8HX/LA=1.VIXFN653
     40,5X,9HCONE R=R8/12X,7HPERPSB= 8E13.4/12X,7HQDOT=   8E13.4/12X,7HMDOVIXFN654
     5T=   ,8E13.4/12X,7HQ INT= ,8E13.4/22X,13HQ INT(STAG)= E11.4,24X,15HVIXFN655
     6Q INT(SONIC)= E11.4,16H  TURBULENT ONLY )                         VIXFN656
C                                                                       VIXFN657
   88 CONTINUE                                                          VIXFN658
      IF(ALPHA.GT.0)GO TO 52                                            VIXFN659
```

```
      IF(LOPT.EQ.1)GO TO 52
C
C          ROTATIONAL QUANTITIES PRINTOUT  FOR LOPT = 0 OR 2
       PSIP=57.29578D0*DVALUE(8)
       THLP=57.29578D0*DVALUE(9)
       PHIP=57.29578D0*DVALUE(10)
      ALPRIP=57.29578D0*ALPRIM
      ALENVP=57.29578D0*ALPENV
      IF(SMF.EQ.0.0D0)TBAR=0.0D0
      IF(SMF.NE.0.0D0)TBAR=1.0D0/SMF
      AAA=PHIP/360.D0
      NNN=AAA
      BBB=NNN
      PHIP=PHIP-360.D0*BBB
      IF(PHIP.LT.0.D0)PHIP=360.D0&PHIP
      WRITE(6,1033)PSIP,THLP,PHIP,ALPRIP,DVALUE(13),DVALUE(11),DVALUE(12
     1),ALENVP,CMO,CM,CN,SMF,TBAR
 1033 FORMAT(1H0 55X,21HROTATIONAL QUANTITIES//25X,4HPSI=E13.4,10H    THE
     1ALP=E11.4,6H  PHI=E13.4,10H    ALPRIM=E11.4/25X,3HPR=E14.4,5H    Q=,
     2E14.4,8H    SMR=E13.4,10H    ALPENV=E11.4/25X,4HCMO=E13.4,6H    CM=
     3E13.4,7H    CN=E14.4,7H    SMF=E14.4,8H    TBAR=E12.4  )
      GO TO 90
C
C          INPUT ANGLE OF ATTACK PRINTOUT
   52 CONTINUE
      ALPHID = ALPHA * 57.29578D0
      WRITE(6,1042) ALPHID
 1042 FORMAT(1H0 41X,37HTABULAR INPUT ANGLE OF ATTACK ALPHA =,E12.4)
      IF(LOPT.LE.2)GO TO 90
C
C          SEQUENCING FOR LOPT=3 OR 4 WHICH CAUSES REPEAT OF CALCULATION
C        FOR NEXT SET OF VALUES IN TABLES, UNLESS FINAL SET OF VALUES
C        HAS ALREADY BEEN USED - THEN CAUSES RETURN TO MAIN PROGRAM.
C         PLOTTING ROUTINES ARE BYPASSED WHEN LOPT = 3 OR 4.
   49 CONTINUE
      IF(MAXVAL.EQ.0)WRITE(6,1025)
      IF(MAXVAL.EQ.0)MAXVAL=1
 1025 FORMAT(1H0,10X,'*****WARNING***** YOU FORGOT TO INPUT MAXVAL FOR L
     1OPT=3 OR 4 CASE')
      IF(NK.EQ.(MAXVAL-1))GO TO 834
      NK=NK&1
      GO TO 3
C
C
C          STOPPING TESTS FOR LOPT LESS THAN OR EQUAL TO 2.
   90 CONTINUE
      IF(TIME.GT.TST)GO TO 51
      IF(Z.LE.(ZST&1.0D0))GO TO 51
      IF(LP.GE.3)GO TO 51
      GO TO 50
   51 IF(NPRINT.NE.0)CALL RITOUT
C
      DO 81 J=1,5
      IF(NPLOT(J).GT.0)GO TO 82
   81 CONTINUE
      GO TO 833
C
C          PLOTTING OPTIONS
   82 CONTINUE
      IF(NPLOT(1).LE.0)GO TO 84
```

```
      CALL AVPLT(TIXZ,T1,ZPLOT,CDPLOT,ZPLOT,3,1)                        VIXFN720
      CALL AVPLT(TIXM,T2,PLMINF,CDPPLT,ZPLOT,3,1)                       VIXFN721
      CALL AVPLT(TIXZ,T3, ZPLOT,CDFPLT,ZPLOT,3,1)                       VIXFN722
      CALL AVPLT(TIXZ,T4, ZPLOT,CDIPLT,ZPLOT,3,1)                       VIXFN723
   84 CONTINUE                                                          VIXFN724
      IF(NPLOT(2).LE.0)GO TO 85                                         VIXFN725
      CALL AVPLT(TIXZ,T6, ZPLOT,BETAPL,ZPLOT,3,1)                       VIXFN726
      CALL AVPLT(TIXT,T7, TPLOT, VPLOT,ZPLOT,3,1)                       VIXFN727
      CALL AVPLT(TIXT,T8, TPLOT,PLMINF,ZPLOT,3,1)                       VIXFN728
      CALL AVPLT (TIXT,T9,TPLOT,VCGPLT,ZPLOT,3,1)                       VIXFN729
      CALL AVPLT(TIXT,T10,TPLOT, ZPLOT,ZPLOT,3,1)                       VIXFN730
      CALL AVPLT(TIXT,T11,TPLOT, QPLOT,ZPLOT,3,1)                       VIXFN731
   85 CONTINUE                                                          VIXFN732
      IF(NPLOT(3).LE.0)GO TO 86                                         VIXFN733
      CALL AVPLT(TIXZ,T5, ZPLOT,WTOTPL,ZPLOT,3,1)                       VIXFN734
      CALL AVPLT(TIXT,T12,TPLOT,PSPLOT,ZPLOT,3,1)                       VIXFN735
      CALL AVPLT(TIXT,T13,TPLOT,HSRTPL,ZPLOT,3,1)                       VIXFN736
      IF(LAMDA.NE.0.0D0)GO TO 851                                       VIXFN737
      CALL AVPLT(TIXT,T14,TPLOT,QD11PL,ZPLOT,3,1)                       VIXFN738
      CALL AVPLT(TIXT,T17,TPLOT,QD8PLT,ZPLOT,3,1)                       VIXFN739
      CALL AVPLT(TIXT,T18,TPLOT,PE1PLT,ZPLOT,3,1)                       VIXFN740
      CALL AVPLT(TIXT,T20,TPLOT,PE8PLT,ZPLOT,3,1)                       VIXFN741
      GO TO 86                                                          VIXFN742
  851 CALL AVPLT(TIXT,T14,TPLOT,QD11PL,ZPLOT,3,1)                       VIXFN743
      CALL AVPLT(TIXT,T15,TPLOT,QD41PL,ZPLOT,3,1)                       VIXFN744
      CALL AVPLT(TIXT,T16,TPLOT,QD7PLT,ZPLOT,3,1)                       VIXFN745
      CALL AVPLT(TIXT,T17,TPLOT,QD8PLT,ZPLOT,3,1)                       VIXFN746
      CALL AVPLT(TIXT,T18,TPLOT,PE1PLT,ZPLOT,3,1)                       VIXFN747
      CALL AVPLT(TIXT,T19,TPLOT,PE7PLT,ZPLOT,3,1)                       VIXFN748
      CALL AVPLT(TIXT,T20,TPLOT,PE8PLT,ZPLOT,3,1)                       VIXFN749
   86 CONTINUE                                                          VIXFN750
      IF(NPLOT(4).LE.0)GO TO 87                                         VIXFN751
      CALL AVPLT(TIXT,T21,TPLOT,ALPLOT,ZPLOT,3,1)                       VIXFN752
   87 CONTINUE                                                          VIXFN753
      IF(NPLOT(5).LE.0)GO TO 833                                        VIXFN754
      CALL AVPLT(TIXT,T22,TPLOT,OCPLOT,ZPLOT,3,1)                       VIXFN755
  833 CONTINUE                                                          VIXFN756
      IF( IOP(74).EQ.0) GO TO 837                                       VIXFN757
      IF( IOP(77).EQ.1) CALL AVPLT(TIXZ,T23,ZPLOT,WL1P,ZPLOT,ICBL,1)    VIXFN758
      IF( IOP(78).EQ.1) CALL AVPLT(TIXZ,T24,ZPLOT,WL2P,ZPLOT,ICBL,1)    VIXFN759
      IF( IOP(79).EQ.1) CALL AVPLT(TIXZ,T25,ZPLOT,WL3P,ZPLOT,ICBL,1)    VIXFN760
      IF( IOP(80).EQ.1) CALL AVPLT(TIXZ,T26,ZPLOT,WR1P,ZPLOT,ICBL,0)    VIXFN761
      IF( IOP(81).EQ.1) CALL AVPLT(TIXZ,T27,ZPLOT,WR2P,ZPLOT,ICBL,0)    VIXFN762
      IF( IOP(82).EQ.1) CALL AVPLT(TIXZ,T28,ZPLOT,WR3P,ZPLOT,ICBL,0)    VIXFN763
  837 CONTINUE                                                          VIXFN764
C                                                                       VIXFN765
C           OPTION FOR TAPE OF V, BETA, Z                               VIXFN766
      IF(ITAPE.EQ.0)GO TO 834                                           VIXFN767
      VVAR =0.0D0                                                       VIXFN768
      WRITE(8)DATE,EMO,CASE,NNTAPE                                      VIXFN769
      WRITE(8)(VVAR,ZPLOT(I),VPLOT(I),VVAR,BETAPL(I),VVAR,VVAR,VVAR,    VIXFN770
     1VVAR,I=1,NNTAPE)                                                  VIXFN771
C                                                                       VIXFN772
C           CHANGING ANGLES BACK TO UNITS OF DEGREES TO PREPARE FOR NEW VIXFN773
C         INPUT QUANTITIES.                                             VIXFN774
  834 CONTINUE                                                          VIXFN775
      THETA=THETA/0.01745329D0                                          VIXFN776
      GAMFS=GAMFS/0.01745329D0                                          VIXFN777
      PHI0=PHI0/0.01745329D0                                            VIXFN778
      PSI0=PSI0/0.01745329D0                                            VIXFN779
```

```
  436 CONTINUE                                                       VIXFN775
      THETA=THETA/0.01745329D0                                       VIXFN776
      GAMEC=GAMEC/0.01745329D0                                       VIXFN777
      PHIO=PHIO/0.01745329D0                                         VIXFN778
      PSIO=PSIO/0.01745329D0                                         VIXFN779
      THEALO=THEALO/0.01745329D0                                     VIXFN780
      RETURN                                                         VIXFN781
      END                                                            VIXFN782
```

```
CWAKE
      SUBROUTINE WAKE
      IMPLICIT REAL*8(A-H,O-Z)
      REAL *8  MUINF, MINF, ME
      COMMON /CPCCUR/ PCCUR(11770)
      COMMON /CIOCUR/ IOCCUR(320)
      COMMON OCCUR,NOCCUR
      COMMON /NINPUT/ AKW,RHOW,CPHOW,DELWH,      TABL,CHCHEM,RHCSL,
     1   RTO,RCON,ACON,     R21,R22,R23,     C(169)
      COMMON /CWAKE/ BETAZ,FRO1,FRO2,FRO3,PHI1,PHI2,PHI3,SIGNL1,SIGNL2,
     1SIGNL3,TAU1,TAU2,TAU3,WKALT,WSTALT,IND,IWAKE,IWPRNT
      COMMON /CRCSEC/ T,TAU,SIGMDS,PHI,XRZ,ZMF,HH,DDW,C1,C2,C3,C4,C5
      COMMON /ERLINK/ BZ,RN,THETAC,DR,GAMMAE,COTA,COVA,PICIPU,PL,
     1    AMU,RHOU,UU,HU,AMUU,AMINFC,RHOIC,HINFC,UINFC,AMULSO,
     2    AMABLD,Z,SC,ZPLT
      DIMENSION OCCUR(1000)
      DIMENSION NOCCUR(10)
      DIMENSION WL1P(160), WL2P(160), WL3P(160), WR1P(160), WR2P(160)
      DIMENSION WR3P(160), ZPLOT(160), A(514), IOP(90)
      DIMENSION BETAZ(10),PHI1(10), PHI2(10), PHI3(10), WKALT(10)
      EQUIVALENCE(OCCUR(001),AREF)
      EQUIVALENCE(OCCUR(014),CT)
      EQUIVALENCE(OCCUR(017),CPE)
      EQUIVALENCE(OCCUR(021),D)
      EQUIVALENCE(OCCUR(026),GAME)
      EQUIVALENCE(OCCUR(034),MUINF)
      EQUIVALENCE(OCCUR(035),MINF)
      EQUIVALENCE(OCCUR(036),ME)
      EQUIVALENCE(OCCUR(048),PE)
      EQUIVALENCE(OCCUR(049),PINF)
       EQUIVALENCE (OCCUR(052),RNMX  )
       EQUIVALENCE (OCCUR(056),RHOINIT)
       EQUIVALENCE (OCCUR(061),RHOE  )
       EQUIVALENCE (OCCUR(072),TINF  )
       EQUIVALENCE (OCCUR(074),THETA )
       EQUIVALENCE (OCCUR(076),TE    )
       EQUIVALENCE (OCCUR(082), V    )
       EQUIVALENCE (OCCUR(083), VE   )
       EQUIVALENCE (OCCUR(086), WDOT  )
       EQUIVALENCE (OCCUR(090), XBAR  )
       EQUIVALENCE (OCCUR(091), ZNK   )
       EQUIVALENCE (OCCUR(092), ZTR   )
       EQUIVALENCE (OCCUR(098), CDP   )
       EQUIVALENCE (OCCUR(099), CDB   )
       EQUIVALENCE (OCCUR(100), CPI   )
       EQUIVALENCE (OCCUR(239), XLCW  )
       EQUIVALENCE (OCCUR(301), A     )
      EQUIVALENCE (IOCCUR(00001), IOP)
      EQUIVALENCE (IOCCUR(00316), IDPL  )
      EQUIVALENCE (IOCCUR(00302), IPLOT )
      EQUIVALENCE (PCCUR(01281), WL1P  )
      EQUIVALENCE (PCCUR(01441), WL2P  )
      EQUIVALENCE (PCCUR(01601), WL3P  )
      EQUIVALENCE (PCCUR(00801), WR1P  )
      EQUIVALENCE (PCCUR(00961), WR2P  )
      EQUIVALENCE (PCCUR(01121), WR3P  )
      EQUIVALENCE (PCCUR(00161), ZPLOT )
      EQUIVALENCE (NOCCUR(03), MOUT  )
       NAMELIST/NAM1/ BZ,RN,THETAC,DR,GAMMAE,COTA,COVA,PICIPU,PL,
```

```
     1    AMU,RHOU,UU,HU,AMOU,AMINFC,RHOIC,HINFC,UINFC,AMBLSD,
     2    AMABLD,Z,SC,ZBLT
C
C
C         SUBROUTINE WAKE CALCULATES SOME QUANTITIES NEEDED BY SUB-
C       ROUTINES FLOWF AND RCSEC, THEN CALLS THESE SUBROUTINES TO
C       CALCULATE THE VEHICLE WAKE LENGTH AND CROSS SECTION AT EACH OF
C       THREE RADAR FREQUENCIES
C
C
      IF(IWRPNT.EQ.0) GO TO 10
      WRITE(6,1000)(A(I),I=105,112),AREF,ZTR,ZNX,D,RNNX,THETA,GAME,CD,
     1 CDP,CDB,CDI,PS,PINF,MINF,RHOIN1,V,TINF,MUINF,ME,RHOE,CPE,TE,
     3 VE,WDOT,XBAR,XLOW
C
10    IF(ZNX.LE.WSTALT.AND.XBAR.LT.XLOW) GO TO 30
      WL1P(LPLOT) = 0.D0
      WL2P(LPLOT) = 0.D0
      WL3P(LPLOT) = 0.D0
      WR1P(LPLOT) = 1.D0
      WR2P(LPLOT) = 1.D0
      WR3P(LPLOT) = 1.D0
      RETURN
C
30    CALL INTERP(10,IWAKE,ZPLOT(LPLOT),WKALT,BETAZ,BZ)
      FUG1 = 1.D3
      FUG2 = 778.D0*32.174D0
C
C          THE FOLLOWING CALCULATIONS ARE FOR RCSEC
      XBZ = BZ
      ZMR = MINF
       HH = ZNX / FUG1
      RCV = D/3.281D0
       C1 = C(1)
       C2 = C(2)
       C3 = C(3)
       C4 = C(4)
       C5 = C(5)
C
C          THE FOLLOWING CALCULATIONS ARE FOR FLOWF.
      RN = RNNX
      THETAC = THETA
      DR = D
      GAMMAE = DABS(GAME)
      CDTA = CD*AREF
       CDVA = (CD-(CDP & CDB & CDI )) * AREF
      PICIPI = PS/PINF
      PU = PINF
      AMU = MINF
      RHOU = RHOIN1
       UU = V / FUG1
      IF(TINF.GT.5.D3) GO TO 66
      IF(TINF.LT.7.D2) GO TO 70
      CPINF = 0.D0
      DO 71 I = 1,6
71    CPINF = CPINF & A(I&104)* TINF**(I-1)
      GO TO 72
66    CPINF = A(111) & A(112)*TINF
      GO TO 72
70    CPINF = .235D00
```

```
72      HU = CPINF*TINF*FUG2
        AMUU = MUINF
        AMINFC = MF
        RHOIC = RHOF
        HINFC = CPF*TF*FUG2
        UINFC = VF/FUG1
        IF(RN.GE..495D0*DB) GO TO 78
        FUG3 = PICIPU*AMINFC/AMU
        FUG4 = DB**3*RHOU *UU*AMUU/(RHOSL*DSIN(THETAC))
         AMBLSD = 1005.D0 * (HINFC / HU)**.25D0 * DSQRT(FUG3 * FUG4)
        GO TO 79
78      FUG5 = 1.D2*RHOL*UU*DB/ AMUU
         AMBLSD = C(76) * RHOU * UU * AREF* FUG5**C(77)
79       AMABLD = -WDOT
         IF(MOPT.EQ.0)  AMABLD = 0.D00
         Z = ZNX / FUG1
        IF(.495D0*DB.GT.RN) GO TO 101
        SC = 0.D0
        GO TO 102
101     SC = (.5D0 * DB - RN * DCOS(THETAC)) / DSIN(THETAC)
102     ZBLT = ZTR/FUG1
C
C
C
        CALL FLOWF(ENET,B1,IND)
        IF(IWPRNT.EQ.1) WRITE(6,2000) ENET, B1
        IF(IWPRNT.EQ.1) WRITE(6,NAM1)
C
        IF((IOP(67) + IOP(70)).EQ.0) GO TO 40
        CALL INTERP(10,IWAKE,ZPLOT(LPLOT),WKALT,PHI1,PHI)
        F = FR01
        TAU = TAU1
        SIGMDS = SIGML1
        CALL RCSEC(B1,ENET,WL1P(LPLOT),WR1P(LPLOT),IOP(67))
        IF(WR1P(LPLOT).GT.0.D0) GO TO 35
        WR1P(LPLOT) = 0.D0
        IF(IOBL.EQ.4) WR1P(LPLOT) = 1.D0
        GO TO 40
C
35      IF(IOBL.EQ.3) WR1P(LPLOT) = 10.D0*DLOG10(WR1P(LPLOT))
C
C
40      IF((IOP(68) + IOP(71)).EQ.0) GO TO 50
        CALL INTERP(10,IWAKE,ZPLOT(LPLOT),WKALT,PHI2,PHI)
         F = FR02
        TAU = TAU2
        SIGMDS = SIGML2
        CALL RCSEC(B1,ENET,WL2P(LPLOT),WR2P(LPLOT),IOP(68))
        IF(WR2P(LPLOT).GT.0.D0) GO TO 45
        WR2P(LPLOT) = 0.D0
        IF(IOBL.EQ.4) WR2P(LPLOT) = 1.D0
        GO TO 50
C
45       IF(IOBL.EQ.3) WR2P(LPLOT) = 10.D0*DLOG10(WR2P(LPLOT))
C
C
50      IF((IOP(69) + IOP(72)).EQ.0) GO TO 60
        CALL INTERP(10,IWAKE,ZPLOT(LPLOT),WKALT,PHI3,PHI)
        F = FR03
        TAU = TAU3
```

```
      SIGMDS = SIGNL3                                                   WAKE 180
      CALL BCSFC(B1,ENET,WL3P(LPLOT),WR3P(LPLOT),IOP(69))               WAKE 181
      IF(WR3P(LPLOT).GT.0.D0) GO TO 55                                  WAKE 182
      WR3P(LPLOT) = 0.D0                                                WAKE 183
      IF(IPBL.EQ.4) WR3P(LPLOT) = 1.D0                                  WAKE 184
      GO TO 60                                                          WAKE 185
C                                                                       WAKE 186
55    IF(IPBL.EQ.3) WR3P(LPLOT)=10.D0*DLOG10(WR3P(LPLCT))               WAKE 187
C                                                                       WAKE 188
C                                                                       WAKE 189
60    RETURN                                                            WAKE 190
C                                                                       WAKE 191
2000  FORMAT(1H0 11HWAKE ENET = 1PE15.7, 1X 4HB1 = E15.7)               WAKE 192
1000  FORMAT(1H03X6HA(105)8X6HA(106)8X6HA(107)8X6HA(108)8X6HA(109)8X6HA(WAKE 193
     1110) 8X6HA(111)8X6HA(112)10X4HGDF/1H 1P9E14.7/1H 11X3HZTP11X3HZNXWAKE 194
     2 13X1HD10X4HRNNX9X5HTHETA10X4HGAMF12X2HC011X3HCDP11X3HCCR/1H      WAKE 195
     3 1P9E14.7/1H 11X3HCDT12X2HPE10X4HPINF10X4HMINF9X6HPHCIN113X1HV10X WAKE 196
     1 4HTINF9X5HMUINF12X2HME/1H 1P9E14.7/1H 10X4HRHOE11X3HCPE12X2HTE   WAKE 197
     2 12X2HVE10X4HWDCT10X4HXBAR10X4HXLOW/1H 1P7E14.7)                  WAKE 198
      END                                                               WAKE 199
```

CIBLE

```
CWRITEM
CWRITEM PROGRAM 25620
      IMPLICIT REAL*8(A-H,O-Z)
      SUBROUTINE WRITEM(IX,Z,T,Q,R,S,DA,DB,SLOPE,N)
C
C
C          EACH TIME SUBROUTINE WRITEM IS CALLED, A SINGLE LINE MAY BE
C       PRINTED IN THE CORRIDOR INTEGRAL TABLE.
C
C
      IF(N.EQ.0) RETURN
       IX = IX/10
      GO TO (10, 20, 30, 40, 50, 60), IX
10    WRITE(6,1000)Z,T,Q,R,S,DA
      RETURN
20    WRITE(6,1100)Z,T,Q,R,S,DA,DB
      RETURN
30    WRITE(6,1400)Z,T,Q,R,S,DA,DB
      RETURN
40    WRITE(6,1300)Z,T,Q,R,S,DA,SLOPE
      RETURN
50    WRITE(6,1500)Z,T,Q,R,S,DA,DB,SLOPE
      RETURN
60    WRITE(6,1400)Z,T,Q,R,S,DA,DB,SLOPE
      RETURN
1000  FORMAT(1H F14.1,4F14.2,1PE14.7,2F14.1,1PE14.7)
1100  FORMAT(1H F14.1,4F14.2,1PE14.7,14X0PF14.1)
1300  FORMAT(1H F14.1,4F14.2,1PE14.7,28X1PE14.7)
1400  FORMAT(1H F14.1,4F14.2,1PE14.7,0PF14.1,14X1PE14.7)
1500  FORMAT(1H F14.1,4F14.2,1PE14.7,14X,0PF14.1,1PE14.7)
      END
```

```
CZPRM
      IMPLICIT REAL*8 (A-H,O-Z)
      SUBROUTINE ZPRM
      COMMON /IGPHL/ IGPL(20), IGPH(20)
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      COMMON /SENSE/ ISEN1, ISEN2
      COMMON /MIN/ALOW(20),UP(20),AMULT(20),CALOW(20),CTP(20)
      COMMON /IOPT/IPROC,IN,NCONS,IPNT,IFX,LIMIT,IRAND
      COMMON /FOPT/ERR,PRAND,FAC,DELTA
      COMMON/OWL/OVECT(20),WRE,LRED
      COMMON /NALTEG/EGSM,NALT
      COMMON/DOPT/DELX(20)
      COMMON /MULT/ SMULT(25)
      EQUIVALENCE (XCOM(1),ALPHA)
      EQUIVALENCE (XCOM(2),BETA)
      EQUIVALENCE (XCOM(3),GAMMA)
      EQUIVALENCE (XCOM(4),DEL)
      EQUIVALENCE (XCOM(5),RATL)
      EQUIVALENCE (XCOM(6),TOL)
C
C
C           SUBROUTINE ZPRM PRESETS THE VALUES OF SOME OF THE VARIABLES
C        USED IN THE PROGRAM.
C
C
      DO 10 I = 1,200
      ICOM(I) = 0
   10 CONTINUE
C
      DO 11 I = 1, 20
       IGPH(I) = 0
       IGPL(I) = 0
      ALOW(I) = 0.D0
      AMULT(I) = 1.D0
      CALOW(I) = 0.D0
      CTP(I) = 0.D0
      DELX(I) = .001D0
      OVECT(I) = 5.D0
      UP(I) = 0.D0
   11 CONTINUE
C
      DO 22 I = 1, 25
      SMULT(I) = 1.D0
   22 CONTINUE
C
      ERR = 0.01D0
      EGSM = 4.D0
      IFX = 2
      IN = 1
      IPROC = 1
      IRAND = 0
      ISEN1 = 0
      ISEN2 = 0
      LIMIT = 30
      NALT = 0
      NCONS = 1
      PRAND = 0.D0
      WRE = .9D0
      XCOM(1) = 3.D0
```

```
      XCON(2) = .500
      XCON(3) = .500
      XCON(4) = .0100
      XCON(5) = .500
      XCON(6) = 1.D-4
C
      RETURN
      END
```

```
CZPRS
      SUBROUTINE ZPRS(A,B)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(516), B(21)
C
C
C        SUBROUTINE ZPRS DEFINES THE ELEMENTS OF THE A AND B ARRAYS
C      WHICH ARE COEFFICIENTS OF THE CURVE FITS CONTAINED IN VARIOUS
C      OTHER SUBROUTINES.
C
C
C
C        COEFFICIENTS FOR THE PROBABILITY DISTRIBUTION BETWEEN FREE
C      MOLECULE AND CONTINUUM FLOW REGIMES, USED IN SUBROUTINE DRAGCO.
      B(01)=0.930930D0
      B(02)=0.118470D0
      B(03)=-0.222490D-1
      B(04)=-0.256360D-2
      B(05)=-0.303250D-2
      B(06)=-0.221750D-2
      B(07)=0.621930D-2
      B(08)=0.309850D-3
      B(09)=-0.134120D-2
      B(10)=-0.797290D-4
      B(11)=0.221050D-3
      B(12)=-0.104390D-5
      B(13)=-0.213930D-4
      B(14)=0.996020D-6
      B(15)=0.123410D-5
      B(16)=-0.781630D-7
      B(17)=-0.404310D-7
      B(18)=0.227690D-8
      B(19)=0.643060D-9
      B(20)=-0.434740D-10
      B(21)=-0.295370D-11
C
C        COEFFICIENTS USED IN FINDING MAXIMUMS AND MINIMUMS OF ALPHA
C      IN SUBROUTINE VIXEN.
      A(11)=4.566917D-01
      A(12)=1.092590D-01
      A(13)=7.526663D-02
      A(14)=9.927520D-02
      A(15)=-1.832002D-02
      A(16)=-9.132577D-03
      A(17)=-4.323047D-03
      A(18)=9.099082D-04
      A(19)=2.959707D-04
      A(20)=1.633470D+01
      A(21)=-6.500104D+00
      A(22)=-1.070414D+01
      A(23)=-2.577606D+00
      A(24)=9.214632D-01
      A(25)=5.961901D-02
      A(26)=1.296622D-01
      A(27)=-3.555784D-02
      A(28)=7.004714D-03
      A(29)=-5.351690D+01
      A(30)=1.517589D+01
      A(31)=-4.108702D-01
```

```
      A(32)=9.747155PC0C
      A(33)=-2.773047D0
      A(34)=9.157596D-02
      A(35)=-4.200407D-C1
      A(36)=1.057159D-01
      A(37)=-3.721217D-3
C
C        COEFFICIENTS USED IN CALCULATING BLUNT CONE PRESSURE DISTRI-
C       BUTIONS FOR CONE HALF ANGLES GREATER THAN OR EQUAL TO
C       20 DEGREES IN AERODY.
      A(41)=1.491575270 1
      A(42)=-9.68495C55D-1
      A(43)=1.795245580 1
      A(44)=-3.4469658D0D0
      A(45)=-5.0879922300
      A(46)=8.0920233PD-1
      A(47)=-1.8517916PD1
      A(48)=1.2319255SDC
      A(49)=-1.0615537401
      A(50)=3.9213215300
      A(51)=5.730922880C
      A(52)=-9.0089157 2D-C1
      A(53)=6.0445217 1DC
      A(54)=-4.4066550 1D-1
      A(55)=5.36326136DC
      A(56)=-1.04904775DC
      A(57)=-1.59869644D0
      A(58)=2.4979807SD-1
C
C        COEFFICIENTS USED IN CALCULATING EDGE TEMPERATURE IN PRELIM.
      A(90)=2.305837D0
      A(91)=1.627017DC
      A(92)=-2.354252D-C1
      A(93)=2.418442D-02
      A(94)=-7.163015D-06
      A(95)=-6.125533D602
      A(96)=9.0910905C1
      A(97)=-1.672942D0
      A(98)= 2.591201D-01
      A(99)=61.241615D-C2
      A(100)=2.1759520603
      A(101)= 2.64793C05C2
      A(102)=-5.4935587601
      A(103)=6.164342D0
      A(104)=-1.846972D-C1
C
C        COEFFICIENTS USED IN CALCULATING SPECIFIC HEATS AT CONSTANT
C       PRESSURE WHEN TEMPERATURE IS IN THE RANGE 700 TO 5000 DEGREES
C       RANKINE INCLUSIVE.  USED IN SUBROUTINE PRELIM.
      A(105)=0.213100
      A(106)=0.4573PD-06
      A(107)= 0.77229D-09
      A(108)=0.47969D-12
      A(109)=0.0D0
      A(110)=0.0D0
C
C        COEFFICIENTS USED IN CALCULATING SPECIFIC HEATS AT CONSTANT
C       PRESSURE WHEN TEMPERATURE IS GREATER THAN 5000 DEGREES RANKINE.
C       USED IN SUBROUTINE PRELIM.
      A(111)=-0.0422DC
```

```
      A(112)=7.00 05
C
C         COEFFICIENTS USED IN BLUNT CONE PRESSURE DISTRIBUTIONS
C      FOR CONE HALF ANGLE ANGLE LESS THAN 20 DEGREES IN AERO.
      A(135)=3.2636863D01
      A(136)=-1.5924494D01
      A(137)=2.1390929D00
      A(138)=4.5075261D00
      A(139)=-3.1997816D00
      A(140)=1.1761463D00
      A(141)=-4.0259457D00
      A(142)=1.3371717D00
      A(143)=-2.2675891D-1
      A(144)=-4.7418269D01
      A(145)=1.9959879D01
      A(146)=-2.7538929D00
      A(147)=-8.2600149D00
      A(148)=5.0601641D00
      A(149)=-1.4883057D00
      A(150)=5.5800904D00
      A(151)=-1.7966998D00
      A(152)=2.5623783D-1
      A(153)=1.4426572D01
      A(154)=-6.1270301D00
      A(155)=8.7365588D-1
      A(156)=3.3502852D00
      A(157)=-1.5080029D00
      A(158)=4.7180038D-1
      A(159)=-1.8490126D00
      A(160)=5.6953762D-1
      A(161)=-6.9064774D-2

C         COEFFICIENTS USED IN CALCU[illegible] TSTAR IN SUBROUTINE DRAGCO.
      A(162)= 1.5276955D03
      A(163)= 2.1399389D00
      A(164)=-2.0710576D-4
      A(165)= 5.5210922D-9
      A(166)=-1.3149173D03
      A(167)= 2.2819959D00
      A(168)=-0.9534279D-4
      A(169)= 1.3648047D-7
      A(170)= 1.0995150D02
      A(171)=-1.9134014D-1
      A(172)= 9.6485959D-5
      A(173)=-1.2079991D-8
C
C         COEFFICIENTS USED IN CALCULATING THE FREE MOLECULE DRAG
C      COEFFICIENT ON THE SPHERICAL NOSE.  USED IN SUBROUTINE DRAGCO.
      A(174)= 0.97366D0
      A(175)=-0.90732D-1
      A(176)=-0.30600D-1
      A(177)= 0.12427D-2
      A(178)= 0.11863D-1
      A(179)= 0.90475D-3
      A(180)=-0.45991D-2
      A(181)=-0.35747D-3
      A(182)=-0.10260D-2
      A(183)= 0.51253D-4
      A(184)=-0.13922D-3
      A(185)=-0.10141D-5
```

```
      A(186)= 0.11221D-6
      A(187)=-0.39507D-6
      A(188)=-0.52692D-6
      A(189)= 0.37457D-7
      A(190)= 0.12721D-7
      A(191)=-0.12579D-8
      A(192)= 0.11965D-6
      A(193)= 0.14016D-10
C
C         COEFFICIENTS USED IN CALCULATING THE STRONG INTERACTION
C        DRAG COEFFICIENT FOR CONE HALF ANGLE LESS THAN 15 DEGREES.
C        USED IN SUBROUTINE DRAGCO.
      A(200)=-3.900218D0
      A(201)=0.5381002D0
      A(202)=0.1757743D0
      A(203)=0.1198765D-01
      A(204)=0.5518595D0
      A(205)=0.5159301D0
      A(206)=3.695611D-04
      A(207)=-0.6825327D-01
C
C         COEFFICIENTS USED IN DETERMINING THE BLUNT CONE FOREBODY
C        PRESSURE DRAG COEFFICIENT FOR CONE HALF ANGLE 4 TO 10 DEGREES
C        INCLUSIVE.  USED IN SUBROUTINE DRAGCO.
      A(211)=  2.8595862AD 1
      A(212)= -1.3418711 3D 1
      A(213)= -9.2344655 D-1
      A(214)= -6.4157242 6D-1
      A(215)= -1.1690280 1D0
      A(216)=  1.6906156 1D0
      A(217)=  1.8400334 1D0
      A(218)= -2.6953467 3D0
      A(219)= -2.5664406 6D0
      A(220)= -3.5693417 7D0
      A(221)=  6.0467792 2D 1
      A(222)= -3.0279348 2D0
      A(223)= -7.9005512 7D0
      A(224)= -3.3763703 2D0
      A(225)= -7.5964451 9D61
      A(226)= -1.6102383 1D61
      A(227)=  2.3273389 8D61
      A(228)=  1.0570783 4D62
      A(229)= -1.6035077 7D61
      A(230)=  7.2199503 0D0
      A(231)= -2.9355727 0D61
      A(232)= -4.4758721 1D61
      A(233)= -8.0673086 6D61
      A(234)=  7.4145260 9D62
      A(235)=  9.5701787 8D61
      A(236)=  2.2664580 9D61
      A(237)= -1.1369862 4D63
      A(238)= -2.4555458 4D61
      A(239)= -2.9718258 8D61
      A(240)=  1.1842267D62
      A(241)=  7.8192592 9D61
      A(242)=  2.6903503D62
      A(243)= -1.8187999 9D63
      A(244)= -1.9052463 6D2
      A(245)= -2.5664327 1D62
      A(246)=  2.9143885 9D63
```

NOT REPRODUCIBLE

```
C
C        COEFFICIENTS USED IN DETERMINING THE BLUNT CONE FOREBODY
C     PRESSURE DRAG COEFFICIENT FOR CONE HALF ANGLE GREATER THAN
C     10 AND LESS THAN OR EQUAL TO 20 DEGREES.  USED IN SUBROUTINE DR
      A(247)=  1.2343642490
      A(248)= -9.6082065810-2
      A(249)=  4.9419185600-1
      A(250)= -4.1177211D-1
      A(251)=  4.9772485490D0
      A(252)= -1.6385192100D1
      A(253)=  1.9414776100-1
      A(254)= -4.4532641900
      A(255)=  2.7365273600D1
      A(256)= -3.4660173400D1
      A(257)=  1.1723986400D1
      A(258)= -1.6716059400D1
      A(259)=  1.3624253200D1
      A(260)= -2.5510120700D2
      A(261)=  8.2206740100D2
      A(262)=  2.1324513500D1
      A(263)=  3.3074564500D2
      A(264)= -1.6006630500D3
      A(265)=  3.6227013900D2
      A(266)= -2.6091522400D2
      A(267)=  3.6799749200D2
      A(268)= -2.3007863200D2
      A(269)=  6.1981713500D3
      A(270)= -1.2462426700D4
      A(271)= -3.1163860400D1
      A(272)= -5.8709289400D3
      A(273)=  2.3100846D4
      A(274)= -1.3259499800D3
      A(275)=  1.3192630400D3
      A(276)= -2.1963811400D3
      A(277)=  1.2103839400D3
      A(278)= -2.0957391300D4
      A(279)= -6.1010700900D4
      A(280)= -5.6657862200D2
      A(281)=  2.9987074700D4
      A(282)= -1.1180287300D5
C
C        COEFFICIENTS USED IN DETERMINING THE SHAPE FACTOR FOR
C     CALCULATING LAMINAR-INDUCED DRAG COEFFICIENT.  USED IN SUB-
C     ROUTINE DRAGCD.
      A(283)=-2.2271731100
      A(284)=1.3322923800-1
      A(285)=3.9985930000-1
      A(286)=1.5928947200-1
      A(287)=-1.8975237900
      A(288)=1.9241583600-1
      A(289)=2.9545776700-1
      A(290)=-1.3320740200-1
C
C        COEFFICIENTS USED IN CALCULATING THE TRANSITION ALTITUDE.
C     USED IN SUBROUTINE PRELIM.
      A(300)=2.4922292E04
      A(301)=9.9402370E09
      A(302)=-3.0976229E02
      A(303)=5.9957280E02
      A(304)=9.1039064E01
```

```
      A(305)=-9.89578800
      A(306)=9.26541480604
      A(307)= 1.20759590506
      A(308)=2.1493710602
      A(309)= 1.09436270606
      A(310)=4.0713080602
      A(311)=2.1394120E1
C
C        COEFFICIENTS USED IN DETERMINING THE BLUNT CONE ECEFFCEY
C     PRESSURE DRAG COEFFICIENT FOR CONE HALF ANGLE GREATER THAN
C     20 AND LESS THAN OR EQUAL TO 40 DEGREES.  USED IN SUBROUTINE
C     DRAGCO.
      A(312)=  3.8975129600
      A(313)= -5.9188305100
      A(314)=  2.7339701061
      A(315)= -3.7491133D0
      A(316)=  3.7055450D61
      A(317)= -1.2134099AD62
      A(318)=  1.0532C173D61
      A(319)= -5.0525534AD61
      A(320)=  1.5035872SD62
      A(321)= -2.00301C98D62
      A(322)=  4.3788C682D62
      A(323)= -2.0475531 7D63
      A(324)=  3.3700141SD62
      A(325)= -3.0000752SD63
      A(326)= -9.7002459SD63
      A(327)= -9.55063071D62
      A(328)=  4.9393483SD63
      A(329)= -1.2734599SD64
      A(330)=  3.9230C074D63
      A(331)= -1.0215C81D64
      A(332)=  4.5599211D64
      A(333)= -9.6976564AD63
      A(334)=  7.7204729SD66
      A(335)= -2.5013418 7D65
      A(336)=  2.76038121D64
      A(337)= -1.3157469D65
      A(338)=  3.4007ECA4D6E
      A(339)= -2.5582199AD64
      A(340)=  7.6899605 8D64
      A(341)= -3.908ICL44D65
      A(342)= -8.8107596AD64
      A(343)= -4.42815PD65
      A(344)=  2.0F09S052D66
      A(345)= -2.4955187D65
      A(346)=  1.1390S333D66
      A(347)= -2.9314593 7D66
C
C        COEFFICIENTS USED IN DETERMINING THE RATIO COP WITH ANGLE
C     OF ATTACK EFFECTS/ COP FOR ANGLE OF ATTACK OF ZERO, WHERE
C     THE ABSOLUTE VALUE OF THE ANGLE OF ATTACK IS GREATER THAN 4 AND
C     LESS THAN OR EQUAL TO 40 DEGREES.  IF THE ABSOLUTE VALUE OF
C     ALPHA IS GREATER THAN 40 DEGREES THEN ALPHA IS SET EQUAL TO 40.
C     USED IN SUBROUTINE DRAGCO.
      A(348)= -8.9760819AD-1
      A(349)=  3.4706 3635D-1
      A(350)=  1.2479244 3D0
      A(351)= -5.7600 73 7D-1
      A(352)=  6.1471 53 8D0
```

```
      A(353)= -6.2415522200
      A(354)=  1.0010310C
      A(355)=  7.2061478800-1
      A(356)= -3.2191836500-1
      A(357)=  1.6075526400
      A(358)= -1.3741112700-1
      A(359)= -4.2207031200-1
      A(360)=  1.5274640500
      A(361)= -9.9420686800
      A(362)= -1.5024605200+1
      A(363)=  5.0627818600
      A(364)= -1.9400500500+1
      A(365)=  1.0993643300+1
      A(366)=  1.1043474900+1
      A(367)= -6.2752800400
      A(368)=  1.9630735100
      A(369)= -1.1795122701
      A(370)= -1.8918523201
      A(371)=  6.6870227700
      A(372)=  3.9367580700
      A(373)= -2.4176607001
      A(374)=  2.6977105001
      A(375)= -9.3161088800
      A(376)=  1.1441032201
      A(377)=  1.4187688001
      A(378)= -3.3651159501
      A(379)= -1.2707537201
      A(380)=  4.5182310200
      A(381)= -3.6461556201
      A(382)=  4.2598850501
      A(383)= -1.3557777501
C
C          COEFFICIENTS USED IN CALCULATING THE STRONG INTERACTICN
C       DRAG COEFFICIENT FOR CONE HALF ANGLE GREATER THAN OR EQUAL TO
C       15 DEGREES.  USED IN SUBROUTINE DRAGCD.
      A(384)=-1.117144550 1
      A(385)=6.0333961000
      A(386)=1.2540985700
      A(387)=-6.9413155400-1
      A(388)=-4.8796628500-2
      A(389)=2.7593112200-2
      A(390)=6.4952995100-4
      A(391)=-3.7302674000-4
      A(392)=4.9774188400
      A(393)=-2.9946866800
      A(394)=-5.5581797200-1
      A(395)=3.5437476100-1
      A(396)=2.1469958000-2
      A(397)=-1.3949187900-2
      A(398)=-2.9395564300-4
      A(399)=1.8204295700-4
C
C          COEFFICIENTS USED TO DETERMINE COEFFICIENTS A(118),-A(119),
C       A(120) USED IN SUBROUTINE DRAGCD TO CALCULATE TURBULENT SKIN
C       FRICTION DRAG COEFFICIENT.
      A(400)=-0.303475E02
      A(401)=0.261500E02
      A(402)=-0.72950DE01
      A(403)=0.120450E01
      A(404)=-0.100240C
```

```
      A(405)=0.51120D-02                                               ZPRS 420
      A(406)=-0.96327D-04                                              ZPRS 421
      A(407)=0.57456D03                                                ZPRS 422
      A(408)=-0.39260D03                                               ZPRS 423
      A(409)=0.11891D03                                                ZPRS 424
      A(410)=-0.19641D02                                               ZPRS 425
      A(411)=0.17049D01                                                ZPRS 426
      A(412)=-0.83948D-01                                              ZPRS 427
      A(413)=0.15912D-02                                               ZPRS 428
      A(414)=-0.20570D04                                               ZPRS 429
      A(415)=0.12804D04                                                ZPRS 430
      A(416)=-0.37451D03                                               ZPRS 431
      A(417)=0.61556D02                                                ZPRS 432
      A(418)=-0.56276D01                                               ZPRS 433
      A(419)=0.26611D00                                                ZPRS 434
      A(420)=-0.50453D-02                                              ZPRS 435
C                                                                      ZPRS 436
C         COEFFICIENTS USED IN DETERMINING THE RATIO CDP WITH ANGLE    ZPRS 437
C        OF ATTACK EFFECTS/ CDP FOR ANGLE OF ATTACK OF ZERO, WHERE     ZPRS 438
C        THE ABSOLUTE VALUE OF ALPHA IS LESS THAN 4 DEGREES AND GREATER ZPRS 439
C        THAN ZERO.  USED IN SUBROUTINE DRAGCO.                        ZPRS 440
      A(421)=  7.7987100D-1                                            ZPRS 441
      A(422)=  4.8657098D-3                                            ZPRS 442
      A(423)= -2.7095153?D-4                                           ZPRS 443
      A(424)=  6.3209?9039D-6                                          ZPRS 444
      A(425)=  6.5269674?D-1                                           ZPRS 445
      A(426)= -2.0116879?D-1                                           ZPRS 446
      A(427)=  1.6953568?D-2                                           ZPRS 447
      A(428)= -4.2126806?D-4                                           ZPRS 448
      A(429)= -3.6441506?D-2                                           ZPRS 449
      A(430)=  3.4995446D-2                                            ZPRS 450
      A(431)= -3.6745991?D-3                                           ZPRS 451
      A(432)=  9.8011533?D-5                                           ZPRS 452
      A(433)=  1.4992746?D-1                                           ZPRS 453
      A(434)= -3.6109271?D-2                                           ZPRS 454
      A(435)=  1.9323491?D-3                                           ZPRS 455
      A(436)= -2.7315082D-5                                            ZPRS 456
      A(437)= -6.7486496?D0                                            ZPRS 457
      A(438)=  2.0062417?D0                                            ZPRS 458
      A(439)= -1.6725229?D-1                                           ZPRS 459
      A(440)=  4.1667530?D-3                                           ZPRS 460
      A(441)=  1.0731207?D0                                            ZPRS 461
      A(442)= -4.2130501?D-1                                           ZPRS 462
      A(443)=  3.9705345?D-2                                           ZPRS 463
      A(444)= -1.0050228?D-3                                           ZPRS 464
      A(445)= -2.6440550?D-1                                           ZPRS 465
      A(446)=  4.7023793D-2                                            ZPRS 466
      A(447)= -2.4411453D-3                                            ZPRS 467
      A(448)=  3.5619831D-5                                            ZPRS 468
      A(449)=  1.4709719?D?1                                           ZPRS 469
      A(450)= -6.9629496?D0                                            ZPRS 470
      A(451)=  4.1416283?D-1                                           ZPRS 471
      A(452)= -1.0376190?D-2                                           ZPRS 472
      A(453)= -3.4799199?D0                                            ZPRS 473
      A(454)=  1.1459987?D0                                            ZPRS 474
      A(455)= -9.0771064?D-2                                           ZPRS 475
      A(456)=  2.5480289?D-3                                           ZPRS 476
C                                                                      ZPRS 477
C         COEFFICIENTS USED IN DETERMINING THE BLUNTNESS CORRECTION    ZPRS 478
C        TO THE SKIN FRICTION DRAG COEFFICIENT IN SUBROUTINE DRAGCO.   ZPRS 479
```

```
C           FOR A BLUNTNESS RATIO GREATER THAN OR EQUAL TO 0.2.                  ZPRS 480
      A(457) = 9.09559981D-01                                                    ZPRS 481
      A(458) = 1.33993866D-01                                                    ZPRS 482
      A(459) =-3.39247271D-00                                                    ZPRS 483
      A(460) = 3.59283188D-10                                                    ZPRS 484
      A(461) =-1.36601870D-11                                                    ZPRS 485
      A(462) = 8.58801823D+00                                                    ZPRS 486
      A(463) =-6.51143480D+00                                                    ZPRS 487
      A(464) = 2.47735029D+00                                                    ZPRS 488
      A(465) =-3.64350270D-01                                                    ZPRS 489
      A(466) = 1.69073239D-02                                                    ZPRS 490
      A(467) =-7.77160227D+01                                                    ZPRS 491
      A(468) = 6.96505886D+01                                                    ZPRS 492
      A(469) =-2.11323256D+01                                                    ZPRS 493
      A(470) = 2.63682113D+00                                                    ZPRS 494
      A(471) =-1.11229610D-01                                                    ZPRS 495
C           FOR A BLUNTNESS RATIO LESS THAN 0.2.                                 ZPRS 496
      A(472) = 2.14018077D+00                                                    ZPRS 497
      A(473) =-1.12954576D+00                                                    ZPRS 498
      A(474) = 3.80856652D-01                                                    ZPRS 499
      A(475) =-4.06886380D-02                                                    ZPRS 500
      A(476) = 2.21265275D-03                                                    ZPRS 501
      A(477) =-1.97107353D+01                                                    ZPRS 502
      A(478) = 1.61930843D+01                                                    ZPRS 503
      A(479) =-4.59456149D+00                                                    ZPRS 504
      A(480) = 5.23362138D-01                                                    ZPRS 505
      A(481) =-2.05329560D-02                                                    ZPRS 506
      A(482) = 2.98504433D+01                                                    ZPRS 507
      A(483) =-2.31135628D+01                                                    ZPRS 508
      A(484) = 6.19713696D+00                                                    ZPRS 509
      A(485) =-4.75613796D-01                                                    ZPRS 510
      A(486) = 2.58207660D-02                                                    ZPRS 511
                                                                                 ZPRS 512
C           COEFFICIENTS USED IN CALCULATING BASE DRAG COEFFICIENT IN            ZPRS 513
C           SUBROUTINE DRAGCO.                                                   ZPRS 514
C           FOR FREE STREAM MACH NUMBER LESS THAN 7.                             ZPRS 515
      A(487)= 3.04562151D-1                                                      ZPRS 516
      A(488)=-4.11272576D0                                                       ZPRS 517
      A(489)= 2.07395569D+1                                                      ZPRS 518
      A(490)=-2.31947425D+1                                                      ZPRS 519
      A(491)= 2.56526225D+1                                                      ZPRS 520
      A(492)=-3.43933596D+2                                                      ZPRS 521
      A(493)= 1.93266242D+3                                                      ZPRS 522
      A(494)=-3.13904376D+3                                                      ZPRS 523
      A(495)=-3.78069826D+2                                                      ZPRS 524
      A(496)= 5.92284350D+3                                                      ZPRS 525
      A(497)=-2.97409434D+1                                                      ZPRS 526
      A(498)= 5.00082593D+4                                                      ZPRS 527
C           FOR FREE STREAM MACH NUMBER GREATER THAN OR EQUAL TO 7.              ZPRS 528
      A(499)= 8.23705525D-1                                                      ZPRS 529
      A(500)=-2.28600636D+1                                                      ZPRS 530
      A(501)= 2.24767224D+2                                                      ZPRS 531
      A(502)=-7.11447646D+2                                                      ZPRS 532
      A(503)= 5.97333564D+1                                                      ZPRS 533
      A(504)=-1.46016249D+3                                                      ZPRS 534
      A(505)= 1.59623113D+4                                                      ZPRS 535
      A(506)=-4.99067523D+4                                                      ZPRS 536
      A(507)=-6.47120825D+2                                                      ZPRS 537
      A(508)= 1.32995714D+4                                                      ZPRS 538
      A(509)= 1.30043494D+5                                                      ZPRS 539
```

```
A(510)= 4.09475475065                                      ZPRS 540
RETURN                                                      ZPRS 541
END                                                         ZPRS 542
```

```
CZREADX
      SUBROUTINE ZREADX
      IMPLICIT REAL*8(A-H,O-Z)
C
C
C          SUBROUTINE ZREADX PROVIDES THE INFORMATION ABOUT THE SIZE
C       AND DIMENSION OF EACH INPUT VARIABLE IT REFERS TO.  THIS
C       INFORMATION IS REQUIRED BY THE INPUT SUBROUTINE READIN, A
C       LIBRARY ROUTINE.
C
C
      COMMON /CIOCUR/ IOCCUR(320)
       COMMON /CPOCUR/ POCCUR(11700)
      COMMON /ORCSEC/ BZERO,BZ,B3,BTWEN,B24,DX,ZNUS,CNE,DS8,X2PCC,X39,
     1IND2,NSTML
      COMMON /NALTFG/FGSM,NALT
      COMMON /NIMPUT/ AKW,RHOW,CRHOW,DELWH,       TABL,CHCHEM,RHCSL,
     1   RTO,RCON,ACON,     R21,R22,R23,     CNUMR(169)
      COMMON /IDNOS/ ID1(50), ID2(50)
      COMMON /IXCOM/ XCOM(200), ICOM(200)
      COMMON/SENSE/ISEN1,ISEN2
      COMMON/MIN/ALOW(20),UP(20), AMULT(20),CALOW(20),CTP(20)
      COMMON/IOPT/IPRCC,IN,NCONS,IPNT,IEX,LIMIT,IRAND
      COMMON/FOPT/FRS,PRAND, FAC, DELTA
      COMMON/DOPT/DELX(20)
      COMMON /TBLS12/ ERNTBL(8,10),ERNRTB(10),ERNUTB(8),
     1    ENTABL(25,9),PSTABL(5),HSTABL(25),
     2    ETABL(12,11),THTTBL(11),EMCTBL(12),
     3        D(11,5,4),XYZTBL(11,3)
      COMMON/BLKO/HH(1600),X(40),GXX(40),S(40),XP(40),TXX(40),
     1GP(40),GS,GSP,GTP,GSS,GTT,GSB,FZZ,FP,FB,FC,E,PZZ,TO,RS,SL,ZZZ,
     1DZZ,AKK,EL,
     2DELKK,FAK,GXX(40,10)
C     WARNING - BLKO COMMON IS BEING SHARED BY DAVDON AND ROSBRK-GRAM
C     OPTIMIZERS
      COMMON/MINSK/N,K, ALB(20), UB(20),XX(20),FXX
      COMMON /MULT/ SMULT(25)
      COMMON /IGOHL/ IGOL(20), IGOH(20)
      COMMON/OWL/OVECT(20),WPF,LRFD
      COMMON /CWAKE/ BETA7,FRO1,FRO2,FRO3,PHI1,PHI2,PHI3,SIGNL1,SIGNL2,
     1SIGNL3,TAU1,TAU2,TAU3,WKALT,WSTALT,IND,IWAKE,IWPRNT
      COMMON OCCUR, NCCCUR
C
C
      DIMENSION OCCUR(4000), NCCCUR(30)
      DIMENSION AA(27),ACOP(140),BCR(40),BCD(40),BCV(40)
      DIMENSION BCWL1(40),BCWL2(40),BCWL3(40),BCWR1(40),BCWR2(40)
      DIMENSION TCWL1(40),TCWL2(40),TCWL3(40),TCWR1(40),TCWR2(40)
      DIMENSION WR2P(160),WR3P(160),ZPLCT(160)
      DIMENSION BETA7(10),PHI1(10), PHI2(10), PHI3(10), WKALT(10)
       DIMENSION SWR3(40),TCR(40),TCD(40),TCV(40)
       DIMENSION NCOMCV(50),NPV(160),SR(40),SD(40),SRS(9),SV(40)
       DIMENSION SWL1(40),SWL2(40),SWL3(40),SWR1(40),SWR2(40)
       DIMENSION BCWR3(40),BETAPL(160),DVH(50),DVL(50),H(40),IOP(90)
       DIMENSION TCWR3(40),TPLCT(160),VOGPLT(160),VPLOT(160)
      DIMENSION        WL1P(160),WL2P(160),WL3P(160),WR1P(160)
C
C
      EQUIVALENCE (OCCUR(3043),CCON)
```

```
C                                                          ZREAD 60
      EQUIVALENCE (PCCUR(11571), AA     )                  ZREAD 61
      EQUIVALENCE (PCCUR(08881), ACDP   )                  ZREAD 62
      EQUIVALENCE (PCCUR(05181), BCB    )                  ZREAD 63
      EQUIVALENCE (PCCUR(05101), BCD    )                  ZREAD 64
      EQUIVALENCE (PCCUR(06501), BCWL1  )                  ZREAD 65
      EQUIVALENCE (PCCUR(05881), BCWL2  )                  ZREAD 66
      EQUIVALENCE (PCCUR(04661), BCWL3  )                  ZREAD 67
      EQUIVALENCE (PCCUR(06261), BCWP1  )                  ZREAD 68
      EQUIVALENCE (PCCUR(05341), BCWP2  )                  ZREAD 69
      EQUIVALENCE (PCCUR(05421), BCWP3)                    ZREAD 70
      EQUIVALENCE (PCCUR(05021), BCV    )                  ZREAD 71
      EQUIVALENCE (PCCUR(00641), BETAPL)                   ZREAD 72
      EQUIVALENCE (PCCUR(11687), DVL    )                  ZREAD 73
      EQUIVALENCE (PCCUR(11607), DVH    )                  ZREAD 74
      EQUIVALENCE (PCCUR(05841), H      )                  ZREAD 75
      EQUIVALENCE (PCCUR(11757), PED    )                  ZREAD 76
      EQUIVALENCE (PCCUR(05821), SB     )                  ZREAD 77
      EQUIVALENCE (PCCUR(05781), SD     )                  ZREAD 78
      EQUIVALENCE (PCCUR(11558), SPS    )                  ZREAD 79
      EQUIVALENCE (PCCUR(05741), SV     )                  ZREAD 80
      EQUIVALENCE (PCCUR(05981), SWL1   )                  ZREAD 81
      EQUIVALENCE (PCCUR(07021), SWL2   )                  ZREAD 82
      EQUIVALENCE (PCCUR(07061), SWL3   )                  ZREAD 83
      EQUIVALENCE (PCCUR(05861), SWP1   )                  ZREAD 84
      EQUIVALENCE (PCCUR(04901), SWP2   )                  ZREAD 85
      EQUIVALENCE (PCCUR(05941), SWP3   )                  ZREAD 86
      EQUIVALENCE (PCCUR(05221), TCB    )                  ZREAD 87
      EQUIVALENCE (PCCUR(05141), TCD    )                  ZREAD 88
      EQUIVALENCE (PCCUR(06541), TCWL1  )                  ZREAD 89
      EQUIVALENCE (PCCUR(05621), TCWL2  )                  ZREAD 90
      EQUIVALENCE (PCCUR(04701), TCWL3  )                  ZREAD 91
      EQUIVALENCE (PCCUR(06301), TCWP1  )                  ZREAD 92
      EQUIVALENCE (PCCUR(05381), TCWP2  )                  ZREAD 93
      EQUIVALENCE (PCCUR(05461), TCWP3  )                  ZREAD 94
      EQUIVALENCE (PCCUR(05061), TCV    )                  ZREAD 95
      EQUIVALENCE (PCCUR(00001), TPLOT  )                  ZREAD 96
      EQUIVALENCE (PCCUR(00481), VOGPLT)                   ZREAD 97
      EQUIVALENCE (PCCUR(00321), VPLOT  )                  ZREAD 98
      EQUIVALENCE (PCCUR(01281), WL1P   )                  ZREAD 99
      EQUIVALENCE (PCCUR(01441), WL2P   )                  ZREAD100
      EQUIVALENCE (PCCUR(01601), WL3P   )                  ZREAD101
      EQUIVALENCE (PCCUR(00801), WP1P   )                  ZREAD102
      EQUIVALENCE (PCCUR(00961), WP2P   )                  ZREAD103
      EQUIVALENCE (PCCUR(01121), WP3P   )                  ZREAD104
      EQUIVALENCE (PCCUR(00161), ZPLOT  )                  ZREAD105
C                                                          ZREAD106
      EQUIVALENCE (IOCCUR(00314), IDBL   )                 ZREAD107
      EQUIVALENCE (IOCCUR(00001), IOP    )                 ZREAD108
      EQUIVALENCE (IOCCUR(00301), IREF   )                 ZREAD109
      EQUIVALENCE (IOCCUR(00302), IPLOT  )                 ZREAD110
      EQUIVALENCE (IOCCUR(00303), MODE   )                 ZREAD111
      EQUIVALENCE (IOCCUR(00001), NCCMDV)                  ZREAD112
      EQUIVALENCE (IOCCUR(00304), NCP    )                 ZREAD113
      EQUIVALENCE (IOCCUR(00305), NOFCDY)                  ZREAD114
      EQUIVALENCE(IOCCUR(00306), NOVCH )                   ZREAD115
      EQUIVALENCE(IOCCUR(00307), NPA    )                  ZREAD116
      EQUIVALENCE (IOCCUR(00161), NPV    )                 ZREAD117
C                                                          ZREAD118
C                                                          ZREAD119
```

```
C                                                                       ZREAC120
C                                                                       ZREAC121
      CALL SETUP(8HAA      ,8,AA,27)                                    ZREAC122
      CALL SETUP(8HACCE    ,8,ACCE,140)                                 ZREAC123
      CALL SETUP ('ACCN    ',8,ACCN)                                    ZREAC124
      CALL SETUP ('AKW     ',8,AKW)                                     ZREAC125
      CALL SETUP( 8HALOW   ,8, ALOW,20)                                 ZREAC126
      CALL SETUP(8HAMULT   ,8,AMULT,20)                                 ZREAC127
C                                                                       ZREAC128
      CALL SETUP(8HBCP     ,8,BCP,40)                                   ZREAC129
      CALL SETUP(8HBCD     ,8,BCD,40)                                   ZREAC130
      CALL SETUP ('BCCN    ',8,BCCN)                                    ZREAC131
      CALL SETUP(8HBCV     ,8,BCV,40)                                   ZREAC132
      CALL SETUP(8HBCWL1   ,8,BCWL1,40)                                 ZREAC133
      CALL SETUP(8HBCWL2   ,8,BCWL2,40)                                 ZREAC134
      CALL SETUP(8HBCWL3   ,8,BCWL3,40)                                 ZREAC135
      CALL SETUP(8HBCWR1   ,8,BCWR1,40)                                 ZREAC136
      CALL SETUP(8HBCWR2   ,8,BCWR2,40)                                 ZREAC137
      CALL SETUP(8HBCWR3   ,8,BCWR3,40)                                 ZREAC138
      CALL SETUP(8HBETAPL  ,8,BETAPL,160)                               ZREAC139
      CALL SETUP(8HBETAZ   ,8,BETAZ,10)                                 ZREAC140
      CALL SETUP(8HBTWEN   ,8,BTWEN)                                    ZREAC141
      CALL SETUP(8HBZERO   ,8,BZERO)                                    ZREAC142
      CALL SETUP(8HB2      ,8,B2)                                       ZREAC143
      CALL SETUP(8HB3      ,8,B3)                                       ZREAC144
      CALL SETUP ('B21     ',8,B21)                                     ZREAC145
      CALL SETUP ('B22     ',8,B22)                                     ZREAC146
      CALL SETUP ('B23     ',8,B23)                                     ZREAC147
      CALL SETUP(8HB24     ,8,B24)                                      ZREAC148
C                                                                       ZREAC149
      CALL SETUP( 8HCALOW  ,8, CALOW,20)                                ZREAC150
      CALL SETUP ('CCPN    ',8,CCPN)                                    ZREAC151
      CALL SETUP(8HCNF     ,8,CNF)                                      ZREAC152
      CALL SETUP ('CNUMB   ',8,CNUMB,169)                               ZREAC153
      CALL SETUP ('CPHOW   ',8,CPHOW)                                   ZREAC154
      CALL SETUP(8HCTP     ,8,CTP,20)                                   ZREAC155
C                                                                       ZREAC156
      CALL SETUP(8HDELTA      ,8,DELTA )                                ZREAC157
      CALL SETUP ('DELWH   ',8,DELWH)                                   ZREAC158
      CALL SETUP(8HDELX       ,8,DELX,20)                               ZREAC159
      CALL SETUP ('DHCHEP  ',8,DHCHEP)                                  ZREAC160
      CALL SETUP(8HDSR     ,8,DSR)                                      ZREAC161
      CALL SETUP ('DTABL   ',8,D,11,8,4)                                ZREAC162
      CALL SETUP(8HDVH     ,8,DVH,50)                                   ZREAC163
      CALL SETUP(8HDVL     ,8,DVL,50)                                   ZREAC164
      CALL SETUP(8HDX      ,8,DX)                                       ZREAC165
C                                                                       ZREAC166
      CALL SETUP ('EMCTRL  ',8,EMCTRL,12)                               ZREAC167
      CALL SETUP ('ENTABL  ',8,ENTABL, 25,9)                            ZREAC168
      CALL SETUP ('ERNPTB  ',8,ERNPTB,10)                               ZREAC169
      CALL SETUP ('ERNTBL  ',8,ERNTBL,8,10)                             ZREAC170
      CALL SETUP ('ERNUTR  ',8,ERNUTR,8)                                ZREAC171
      CALL SETUP(8HERR      ,8,ERR  )                                   ZREAC172
      CALL SETUP ('ETABL   ',8,ETABL,12,11)                             ZREAC173
C                                                                       ZREAC174
      CALL SETUP(8HFAC          ,8,FAC    )                             ZREAC175
      CALL SETUP(8HFDSM    ,8,FDSM)                                     ZREAC176
      CALL SETUP(8HFR01    ,8,FR01)                                     ZREAC177
      CALL SETUP(8HFR02    ,8,FR02)                                     ZREAC178
      CALL SETUP(8HFR03    ,8,FR03)                                     ZREAC179
```

```
C
      CALL SETUP(8HH       ,8,H,40)
      CALL SETUP(8HHH      ,8,HH,160CT
      CALL SETUP ('HSTABL  ',8,HSTABL,25)
C
      CALL SETUP(8HICCM    ,4,ICCM,200)
      CALL SETUP(8HIDBL    ,4,IDBL)
      CALL SETUP(8HIDND     ,4, ID1,50)
      CALL SETUP(8HIDC      ,4,ID2, 50)
      CALL SETUP(8HIEX      ,4,IEX  )
      CALL SETUP(8HIGDH    ,4,IGDH,20)
      CALL SETUP(8HIGDL    ,4,IGDL,20)
      CALL SETUP(8HIN       ,4,IN,  )
      CALL SETUP ('IND     ',4,IND1)
       CALL SETUP(8HIND2    ,4,IND2)
      CALL SETUP(8HIOP     ,4,IOP,
      CALL SETUP(8HIPNT     ,4,IPNT)
      CALL SETUP(8HIPPOC    ,4,IPPOC)
      CALL SETUP(8HIRAND    ,4,IRAND)
      CALL SETUP(8HIREF    ,4,IREF)
      CALL SETUP(8HISEN1   ,4,ISEN1)
      CALL SETUP(8HISEN2   ,4,ISEN2)
       CALL SETUP(8HIWAKE    ,4,IWAKE)
       CALL SETUP(8HIWPRNT    ,4,IWPRNT)
C
      CALL SETUP(8HK          ,4, K)
C
      CALL SETUP(8HLIMIT     ,4,LIMIT)
      CALL SETUP(8HLPLOT    ,4,LPLOT)
      CALL SETUP(8HLRED     ,4,LRED)
C
      CALL SETUP(8HMODE     ,4,MODE)
C
      CALL SETUP(8HNALT     ,4,NALT)
      CALL SETUP(8HNCCMDV  ,4,NCCMDV,80)
      CALL SETUP(8HNCCNS     ,4,NCCNST)
      CALL SETUP(8HNCP      ,4,NCP)
      CALL SETUP(8HNDECDV  ,4,NDECDV)
      CALL SETUP(8HNDVCH    ,4,NDVCH)
      CALL SETUP(8HNPA      ,4,NPA)
      CALL SETUP(8HNPV      ,4,NPV,1AC)
       CALL SETUP(8HNSTWL    ,4,NSTWL)
C
      CALL SETUP(8HOVECT    ,8,OVECT,20)
C
      CALL SETUP(8HPEC      ,8,PEC)
       CALL SETUP(8HPHI1      ,8,PHI1 ,10)
       CALL SETUP(8HPHI2      ,8,PHI2 ,10)
       CALL SETUP(8HPHI3      ,8,PHI3 ,10)
      CALL SETUP(8HPRAND      ,8,PRAND)
C
      CALL SETUP ('RHCSL   ',8,RHCSL)
      CALL SETUP ('RHCW    ',8,RHCW)
      CALL SETUP ('RSTABL  ',8,RSTABL,9)
      CALL SETUP ('RTC     ',8,RTC)
C
       CALL SETUP(8HSP          ,8,SP,40)
       CALL SETUP(8HSC          ,8,SC,40)
       CALL SETUP(8HSIGNL1    ,8,SIGNL1)
       CALL SETUP(8HSIGNL2    ,8,SIGNL2)
```

```
      CALL SETUP(8HS...      ,8,...)
      CALL SETUP(8HSIGNL1    ,8,SIGNL1)
      CALL SETUP(8HSIGNL2    ,8,SIGNL2)

      CALL SETUP(8HSIGNL3    ,8,SIGNL3)
      CALL SETUP(8HSMULT     ,8,SMULT,25)
      CALL SETUP(8HSRS       ,8,SRS,5)
      CALL SETUP(8HSV        ,8,SV,40)
      CALL SETUP(8HSWL1      ,8,SWL1,40)
      CALL SETUP(8HSWL2      ,8,SWL2,40)
      CALL SETUP(8HSWL3      ,8,SWL3,40)
      CALL SETUP(8HSWR1      ,8,SWR1,40)
      CALL SETUP(8HSWR2      ,8,SWR2,40)
      CALL SETUP(8HSWR3      ,8,SWR3,40)

      CALL SETUP ('TABL      ',8,TABL)
      CALL SETUP(8HTAU1      ,8,TAU1)
      CALL SETUP(8HTAU2      ,8,TAU2)
      CALL SETUP(8HTAU3      ,8,TAU3)
      CALL SETUP(8HTCB       ,8,TCB,40)
      CALL SETUP(8HTCP       ,8,TCP,40)
      CALL SETUP(8HTCV       ,8,TCV,40)
      CALL SETUP(8HTCWL1     ,8,TCWL1,40)
      CALL SETUP(8HTCWL2     ,8,TCWL2,40)
      CALL SETUP(8HTCWL3     ,8,TCWL3,40)
      CALL SETUP(8HTCWR1     ,8,TCWR1,40)
      CALL SETUP(8HTCWR2     ,8,TCWR2,40)
      CALL SETUP(8HTCWR3     ,8,TCWR3,40)
      CALL SETUP ('THTTBL    ',8,THTTBL,11)
      CALL SETUP(8HTPLOT     ,8,TPLOT,160)

      CALL SETUP(8HUP        ,8, UP,20)

      CALL SETUP(8HVPLOT     ,8,VPLOT,160)
      CALL SETUP(8HVOGPLT    ,8,VOGPLT,160)
C
      CALL SETUP(8HWKALT     ,8,WKALT,10)
      CALL SETUP(8HWL1P      ,8,WL1P,160)
      CALL SETUP(8HWL2P      ,8,WL2P,160)
      CALL SETUP(8HWL3P      ,8,WL3P,160)
      CALL SETUP(8HWRF       ,8,WRF)
      CALL SETUP(8HWR1P      ,8,WR1P,160)
      CALL SETUP(8HWR2P      ,8,WR2P,160)
      CALL SETUP(8HWR3P      ,8,WR3P,160)
      CALL SETUP(8HWSTALT    ,8,WSTALT)
C
      CALL SETUP(8HXCOM      ,8,XCOM,200)
      CALL SETUP ('XDTABL    ',8,XYZTBL,11)
      CALL SETUP(8HX2BCD     ,8,X2BCD)
      CALL SETUP(8HX3B       ,8,X3B)

      CALL SETUP ('YDTABL    ',8,XYZTBL(1,2), 11)

      CALL SETUP ('ZDTABL    ',8,XYZTBL(1,3), 11)
      CALL SETUP(8HZNUS      ,8,ZNUS)
      CALL SETUP(8HZPLOT     ,8,ZPLOT,160)

      RETURN
      END
```

```
CACOS
      IMPLICIT REAL*8 (A-H,O-Z)                                          ACOS   1
CJACOS  ACOS  ACOSR  ACOSD                                               ACOS   2
      REAL FUNCTION ACOSR*8(W)                                           ACOS   3
      ENTRY ACOS(W)                                                      ACOS   4
C  THE ANGLE RETURNED WILL BE BETWEEN O AND PIE                          ACOS   5
      D = 1.0D0                                                          ACOS   6
    2 WS = 1.0D0 - W * W                                                 ACOS   7
      ACOSR = ATANOR (DSQRT(WS),W) / D                                   ACOS   8
    3 RETURN                                                             ACOS   9
      ENTRY ACOSD(W)                                                     ACOS  10
      D = .17453292519943290-1                                           ACOS  11
      GO TO 2                                                            ACOS  12
      END                                                                ACOS  13
                                                                         ACOS  14
```

```
CASIN
      IMPLICIT REAL*8 (A-H,O-Z)
CJASIN  ASIN  ASINR  ASIND
      REAL FUNCTION ASINR*8(W)
      ENTRY ASIN(W)
C  THE ANGLE RETURNED WILL BE BETWEEN 0 AND PIE/2 OR 3*PIE/2 AND 2*PIE
      D = 1.0D0
    2 WS = 1.0D0 - W * W
    1 ASINR = ATANQR (W,DSQRT(WS)) / D
    3 RETURN
      ENTRY ASIND(W)
      D = .1745329251994329D-1
      GO TO 2
      END
```

```
CATAN
      IMPLICIT REAL*8 (A-H,O-Z)
CJTNQR  ATANQR  ATANQD
      REAL FUNCTION ATANQR*8(W,H)
      ENTRY ATANQ(W,H)
      D = 1.0D0
 110  X = DATAN2 (W,H)
      IF (X) 1, 2, 2
   1  X = X+6.2831853071795B6
   2  ATANQR = X / D
   4  RETURN
      ENTRY ATANQD(W,H)
      D = .17453292519943295D-1
      GO TO 110
      END
```

```
CENDJOB
      SUBROUTINE ENDJOB
C
C        ENDJOB IS A DUMMY SUBROUTINE
C
      RETURN
      END
```

```
CEZPLT
      SUBROUTINE EZPLT
C
C          EZPLT IS A DUMMY SUBROUTINE
C
      RETURN
      END
```

```
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                                                                FRAME   1
      SUBROUTINE FRAMEV                                                FRAME   2
C                                                                      FRAME   3
C     FRAMEV IS A DUMMY SUBROUTINE                                     FRAME   4
C                                                                      FRAME   5
      RETURN                                                           FRAME   6
      END                                                              FRAME   7
```

```
CIDFRMV
      SUBROUTINE IDFRMV
C
C     IDFRMV IS A DUMMY SUBROUTINE
C
      RETURN
      END
```

```
CMIBCOM
      SUBROUTINE MIBCOM
C
C       MIBCOM IS A DUMMY SUBROUTINE
C
      RETURN
      END
```

```
CPLTND
      SUBROUTINE PLTND
C
C     PLTND IS A DUMMY SUBROUTINE
C
      RETURN
      END
```

```
READ     TITLE -                                        VARIABLE FIELD INPUT ROUTINEX
               S FOR OS/360-
         PRINT ON,DATA,GEN
         MACRO
+NAME    COL72 +HOW,+TO
+NAME    LA    RC,1(0,RC)      INCREMENT COLUMN COUNTER
         CR    RC,RCMAX        SEE IF COLUMN 72 HAS BEEN REACHED
         B+HOW +TO             BRANCH ACCORDINGLY
         MEND
         MACRO
+NAME    COMP  +WITH,+HOW,+TO
+NAME    CLI   0(RC),+WITH     EXAMINE THE CHARACTER ON THE CARD
         B+HOW +TO             BRANCH ACCORDINGLY
         MEND
         MACRO
+NAME    CHECK +REG,+HOW,+TO
+NAME    LTR   +REG,+REG       TEST REGISTER
         B+HOW +TO             BRANCH ACCORDINGLY
         MEND
         MACRO
+NAME    CNV   +LOC
+NAME    UNPK  +LOC(12),D8(6)(2)  UNPACK FOR PRINTING
         OI    +LOC+11,X'F0'   DELETE SIGN
         MEND
         MACRO
+NAME    CONV  +REG,+LOC
+NAME    SLL   +REG,4          HALF-BYTE SHIFT TO MAKE ROOM FOR SIGN
         ST    +REG,D8+4       STORE IN RIGHT HALF OF DOUBLE WORD
         OI    D8+7,X'0C'      INSERT PLUS SIGN
         CNV   +LOC
         MEND
LA000000 START 0
         ENTRY READIN,WHERE,SETUP,HEDING
         EXTRN EXIT,FDXPI#,IUCOM#
         EJECT
*        SUBROUTINE READIN(INCOL1) OR READIN(INCOL1,*) WHEN INCOL1=8888
*        A. L. ATKINSON, R430, AUGUST 1966.  BAL VERSION - JUNE 1967
*        LATEST MODIFICATION  -  9 FEBRUARY 1968
         SPACE 2
READIN   BC    15,12(0,15)     BRANCH AROUND ID
         DC    XL2'0700'
         DC    CL6'READIN'
         STM   14,12,12(13)    SAVE REGISTERS
         L     RW,0(0,1)       PICK UP ADDRESS OF INCOL1
         LR    12,15           PICK UP PROGRAM BASE REGISTER
         USING READIN,12       INFORM ASSEMBLER OF PROGRAM BASE REGISTER
         L     RD,ADATA        PICK UP BASE REGISTER FOR PROGRAM DATA
         USING DATA,RD         INFORM ASSEMBLER OF THAT BASE REGISTER
         LR    4,13            PICK UP ADDRESS OF OLD SAVE AREA
         LA    13,SAVE         PICK UP ADDRESS OF NEW SAVE AREA
         ST    13,8(0,4)       SAVE ADDRESS OF NEW SAVE AREA IN OLD ONE
         ST    4,SAVE+4        SAVE ADDRESS OF OLD SAVE AREA IN NEW ONE
         ST    RW,ADINCOL1     SAVE ADDRESS OF INCOL1
STARTOLD NOP   STARTNEW
         LH    RT,NUMTAB
         CHECK RT,NP,ERROR1    SEE IF THE INPUT TABLE WAS CONSTRUCTED
         L     RW,0(0,RW)      PICK UP INCOL1
         C     RW,C8888        WAS IT INITIALIZED TO 8888
         BNE   SKIP            IF NOT, BRANCH AROUND
         MVI   IEND+1,X'00'    SET NON-STANDARD RETURN SWITCH
```

```
SKIP      MVI   STARTØLD,1,X'F0'                                     EAD  61
STARTNEW  SR    RA,RA                                                EAD  62
          ST    RA,INCØL1        ZERØ ØUT INCØL1                     EAD  63
          LA    RCMAX,CARD+71 PICK UP ADDRESS ØF CARD CØLUMN 72      EAD  64
TITLE     NØP   NEXTCARD                                             EAD  65
          SDR   0,0                                                  EAD  66
          STD   0,D8             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  0,24             ØBTAIN HØURS                        EAD  71
          CØNV  0,HRS            CØNVERT FØR PRINTING                EAD  72
          CVB   0,D8             CØNVERT TØ BINARY                   EAD  73
          C     0,=F'10'                                             EAD  74
          BNL   ØKK              IF LESS THAN TEN,                   EAD  75
          MVI   HRS,X'40'            INSERT BLANK ØN LEFT            EAD  76
ØKK       SR    0,0                                                  EAD  77
          SLDL  0,8              ØBTAIN MINUTES                      EAD  78
          CØNV  0,MNS            CØNVERT FØR PRINTING                EAD  79
          SR    0,0                                                  EAD  80
          SLDL  0,8              ØBTAIN SECØNDS                      EAD  81
          CØNV  0,SCS            CØNVERT FØR PRINTING                EAD  82
          SR    0,0                                                  EAD  83
          SLDL  0,8              ØBTAIN HUNDREDTHS ØF A SECØND       EAD  84
          CØNV  0,HSC                                                EAD  85
          SDR   0,0                                                  EAD  86
          STD   0,D8                                                 EAD  87
          L     5,DATE                                               EAD  88
          LR    1,5                                                  EAD  89
          STH   5,D8+6           ØBTAIN DAY ØF THE YEAR              EAD  90
          CVB   5,D8             CØNVERT TØ BINARY INTEGER           EAD  91
          SRL   1,16             ØBTAIN YEAR                         EAD  92
          CØNV  1,YR             CØNVERT FØR PRINTING                EAD  93
          CVB   2,D8             CØNVERT TØ A BINARY INTEGER         EAD  94
          SR    6,6              MØNTH CØUNTER                       EAD  95
          LR    3,6                                                  EAD  96
          SRDL  2,2              DIVIDE YEAR BY FØUR                 EAD  97
          LTR   3,3              IS THIS A LEAP YEAR                 EAD  98
          LA    2,4              INDEXING FØR SEARCH FØR MØNTH       EAD  99
          LA    3,44                                                 EAD 100
          BNZ   NØLEAP                                               EAD 101
          C     5,=F'60'         CHECK FØR FEBRUARY 29TH             EAD 102
          BL    NØLEAP                                               EAD 103
          BH    LEAP                                                 EAD 104
FEB29     LA    6,2                                                  EAD 105
          LA    5,29                                                 EAD 106
          B     DØNE                                                 EAD 107
LEAP      LA    6,8              MØNTH IS MARCH, AT LEAST            EAD 108
          BCTR  5,0              REDUCE DATE TØ ACCØUNT FØR LEAP YEAR EAD 109
NØLEAP    C     5,JAN(6)         CØMPARE WITH LAST DAY ØF THE MØNTH  EAD 110
          BNH   MØNTH            FØUND THE MØNTH                     EAD 111
          BXLE  6,2,NØLEAP       LØØP AGAIN                          EAD 112
MØNTH     S     5,INCØL1(6)      CØNVERT TØ DAY ØF MØNTH             EAD 113
          SRL   6,2              ØBTAIN NUMBER                       EAD 114
          LA    6,1(0,6)             ØF MØNTH                        EAD 115
DØNE      CVD   5,D8             CØNVERT DAY ØF MØNTH TØ DECIMAL     EAD 116
          CNV   DAY              CØNVERT FØR PRINTING                EAD 117
          CVD   6,D8             CØNVERT MØNTH TØ DECIMAL            EAD 118
          CNV   MTH              CØNVERT FØR PRINTING                EAD 119
```

```
         C       6,=F-10-                                                  EAD  120
         BNL     ØK           IF LESS THAN TEN,                            EAD  121
         MVI     MTH,X-40-      INSERT BLANK ØN LEFT                       EAD  122
ØK       L       15,VIBCØM    WRITE HEADING FØR A NEW PAGE ØF DATA         EAD  123
         CNØP    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,INCØL1    PICK UP INCØL1                               EAD  129
         CHECK   RW,P,RESTØRE SEE IF CARD WAS A TRANSFER CARD              EAD  130
READCARD L       15,VIBCØM                                                 EAD  131
         CNØP    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    NØP     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 ØF CARD CØLUMN 1             EAD  148
         SR      R5,R5        ZERØ ØUT INCREMENTING FLAG                   EAD  149
NFLAG    NØP     ACARD2                                                    EAD  150
         CØMP    BLANK,E,ETC  EXAMINE FIRST CHARACTER ØN THE CARD          EAD  151
         CØMP    E,E,IEND                                                  EAD  152
         CØMP    H,E,HCARD                                                 EAD  153
         CØMP    Z,E,HEXCARD                                               EAD  154
         CØMP    X,E,HEXCARD                                               EAD  155
         CØMP    A,E,ACARD                                                 EAD  156
         CØMP    DØLLAR,E,READCARD                                         EAD  157
         CØMP    STAR,E,READCARD                                           EAD  158
         CØMP    N,E,NNNN                                                  EAD  159
         CØMP    F,E,FFFF                                                  EAD  160
         CØMP    P,E,PPPP                                                  EAD  161
         CØMP    ZERØ,NH,ETC                                               EAD  162
         NI      0)RC*,X-0F-  ØBTAIN DIGIT FRØM TRANSFER CARD              EAD  163
         SR      RW,RW                                                     EAD  164
         IC      RW,0)0,RC*                                                EAD  165
         ST      RW,INCØL1    STØRE IT IN INCØL1                           EAD  166
ETC      LA      RC,1)0,RC*   INCREMENT CØLUMN CØUNTER                     EAD  167
NEWFIELD CR      RC,RCMAX     LØØK FØR A NEW FIELD                         EAD  168
         BH      NEXTCARD                                                  EAD  169
S10      CØMP    BLANK,E,NEXT EXAMINE NEXT CHARACTER ØN THE CARD           EAD  170
         CØMP    CØMMA,E,NEXT                                              EAD  171
         CØMP    EQ029,E,NEXT                                              EAD  172
         CØMP    EQ026,NE,CHAR1                                            EAD  173
NEXT     CØL72   NH,S10                                                    EAD  174
S9       B       NEXTCARD                                                  EAD  175
NNNN     CLC     0)7,RC*,NØPRINT                                           EAD  176
         BE      NØPRINTS                                                  EAD  177
         CLC     0)6,RC*,NØFLIP                                            EAD  178
```

```
         BNE   ETC
         MVI   F1.3,X-F0-    SUPPRESS PAGE FLIP FOR EACH CASE
         B     READCARD
NOPRINTS MVI   PRINT.1,X-F0- SUPPRESS PRINTING OF DATA CARDS
         MVI   TITLE.1,X-F0-
         B     READCARD
FFFF     CLC   014,RC*,FLIP
         BNE   ETC
         MVI   F1.3,X-F1-    FLIP PAGE FOR EACH CASE
         B     READCARD
PPPP     CLC   015,RC*,PRINTS
         BNE   ETC
         MVI   PRINT.1,X-00- PRINT DATA CARDS
         MVI   TITLE.1,X-00-
         B     READCARD
ADATA    DC    A(DATA*
CHAR1    MVI   LOG.1,X-00-   EXAMINE FIRST CHARACTER IN NEW FIELD
         SR    RS,RS         ZERO OUT DECIMAL POINT LOCATOR
         LR    RB,RS         ZERO OUT EXPONENT LOCATOR
         COMP  ZERO,NL,S503  CHECK FOR NUMERIC
         COMP  A,NL,LETTER   CHECK FOR ALPHABETIC
         COMP  DOLLAR,E,LETTER DOLLAR SIGN IS ALPHABETIC
         COMP  MINUS,E,S510
         COMP  SLASH,E,ALPHA
         COMP  LP026,E,SUBSCRPT
         COMP  LP029,E,SUBSCRPT
         COMP  DECIMAL,E,S508
         COMP  PLUS026,E,S509
         COMP  PLUS029,E,S509
         B     ERROR
         EJECT
*        NUMERIC FIELD
         SPACE 2
S508     MVI   SIGN.1,X-C0-  DECIMAL POINT, SET SIGN PLUS
         MVI   LOG.1,X-F0-   INITIALIZE LOGICAL FLAG
S5080    CHECK RS,NZ,ERROR   CHECK FOR MORE THAN ONE DECIMAL POINT
         LR    RS,RC         SET DECIMAL POINT LOCATOR
         B     S5090
S510     MVI   SIGN.1,X-D0-  SET SIGN MINUS
         B     S5090
LOG      NOP   LOGICAL       BRANCH ONLY IF DECIMAL POINT PRECEDES
         B     ERROR
S509     MVI   SIGN.1,X-C0-  SET SIGN PLUS
S5090    COL72 H,ERROR
         COMP  ZERO,NL,NUM
         COMP  DECIMAL,E,S5080
         COMP  F,NL,LOG
         B     ERROR
S503     MVI   SIGN.1,X-C0-  NUMERIC, SET SIGN PLUS
NUM      LR    RT,RC         PROCESS NUMERIC FIELD
         LA    RW,910,RT*
         ST    RW,D8
NEXTC    COL72 H,S501        EXAMINE NEXT CHARACTER IN NUMERIC FIELD
         COMP  ZERO,NL,S513
         COMP  D,NL,S504
         COMP  DECIMAL,E,S502
         COMP  MINUS,E,S521
         COMP  PLUS026,E,S521
         COMP  PLUS029,E,S521
```

```
         CØMP  BLANK,E,S501                                          EAD  238
         CØMP  LP026,E,LEFTP                                         EAD  239
         CØMP  LP029,E,LEFTP                                         EAD  240
         CØMP  RP026,E,RIGHTP                                        EAD  241
         CØMP  RP029,E,RIGHTP                                        EAD  242
         CØMP  CØMMA,NE,ERRØR                                        EAD  243
S501     CHECK RS,NZ,FLØAT     CHECK FØR FLØATING PØINT              EAD  244
INTGER   CHECK R5,Z,INTGER1    CHECK FØR INCREMENTING                EAD  245
         BP    ERRØR           UNBALANCED PARENTHESES                EAD  246
         SR    R5,R5           ZERØ ØUT INCREMENTING FLAG            EAD  247
         B     INCREM                                                EAD  248
INTGER1  BAL   14,LA000C       CØNVERT BCD TØ BINARY INTEGER         EAD  249
         CLI   LENGTH.1,3      CHECK LENGTH ØF VARIABLE              EAD  250
         BH    INT4                                                  EAD  251
         MVI   STØREI,X-40-    CØNVERT STØRE INSTRUCTIØN TØ -STH-    EAD  252
STØREI   ST    RW,010,RA*      STØRE INTEGER VALUE                   EAD  253
         AR    RA,RI           INCREMENT STØRAGE ADDRESS             EAD  254
DASHI    B     NEWFIELD                                              EAD  255
DASHI1   NØP   DASHI2                                                EAD  256
         LH    RS,NMAX                                               EAD  257
         SH    RS,NMIN                                               EAD  258
         A     RS,ØNE                                                EAD  259
         MVI   DASHI1.1,X-F0-                                        EAD  260
DASHI2   BCT   RS,STØREI                                             EAD  261
         MVI   DASHI1.1,X-00-                                        EAD  262
         B     DASHDØNE                                              EAD  263
LEFTP1   S     RC,ØNE                                                EAD  264
LEFTP    CHECK RS,NZ,FLØATL    CHECK FØR FLØATING PØINT              EAD  265
         CHECK R5,Z,FIRSTL     IS THIS THE FIRST LEFT PARENTHESIS    EAD  266
         BP    ERRØR           UNBALANCED PARENTHESES                EAD  267
         LA    R5,1            SET INCREMENTING FLAG                 EAD  268
INCREM   BAL   14,LA000C       CØNVERT BCD TØ BINARY INTEGER         EAD  269
         LR    14,RW           PICK UP NSTØP                         EAD  270
         SR    14,R7           SUBTRACT NSTART                       EAD  271
         SRDA  14,32           RIGHT ADJUST FØR DIVISIØN             EAD  272
         DR    14,R11          DIVIDE BY NINCRM                      EAD  273
         CHECK R5,P,MULT       CHECK FØR MULTIPLE INCREMENTING       EAD  274
         LA    15,110,15*      STANDARD INCREMENTING, SØ STØRE NSTØP ALSØ EAD  275
MULT     CLI   LENGTH.1,3      CHECK LENGTH ØF VARIABLE              EAD  276
         BH    INC4                                                  EAD  277
         MVI   LPINC,X-40-     CØNVERT STØRE INSTRUCTIØN TØ -STH-    EAD  278
LPINC    ST    R7,010,RA*      STØRE INTEGER VALUE                   EAD  279
         AR    R7,R11          INCREMENT INTEGER VALUE               EAD  280
         AR    RA,RI           INCREMENT STØRAGE ADDRESS             EAD  281
         BCT   15,LPINC                                              EAD  282
         CHECK R5,P,QQQ        CHECK FØR MULTIPLE INCREMENTING       EAD  283
         CØL72 H,NEXTCARD                                            EAD  284
         B     NEWFIELD                                              EAD  285
INC4     MVI   LPINC,X-50-     CØNVERT STØRE INSTRUCTIØN TØ -ST-     EAD  286
         B     LPINC                                                 EAD  287
FIRSTL   LA    R5,1            SET INCREMENTING FLAG                 EAD  288
         BAL   14,LA000C       CØNVERT BCD TØ BINARY INTEGER         EAD  289
QQQ      LR    R7,RW           PICK UP NSTART                        EAD  290
         CØL72 H,ERRØR                                               EAD  291
         B     NEWFIELD                                              EAD  292
RIGHTP1  S     RC,ØNE                                                EAD  293
RIGHTP   CHECK RS,NZ,FLØATR    CHECK FØR FLØATING PØINT              EAD  294
         CHECK R5,NP,ERRØR     ERRØR IF NØ LEFT PARENTHESIS HAS PRECEDED  EAD  295
         LCR   R5,R5           CHANGE SIGN ØF INCREMENTING FLAG      EAD  296
```

```
         BAL   14,LA000C      CØNVERT BCD TØ BINARY INTEGER                EAD  297
         LR    R11,RW         PICK UP NINCRM                               EAD  298
         CØL72 H,ERRØR                                                     EAD  299
         B     NEWFIELD                                                    EAD  300
S513     CHECK RS,NZ,NEXTC    BRANCH IF DECIMAL PØINT HAS BEEN SEEN        EAD  301
         C     RC,D8         IF MØRE THAN NINE DIGITS TØ LEFT ØF           EAD  302
         BH    GR9            DECIMAL PØINT, BRANCH ØUT                    EAD  303
         B     NEXTC                                                       EAD  304
S504     CØMP  E,H,ERRØR      EXPØNENT FIELD                               EAD  305
         CHECK R8,NZ,ERRØR    ERRØR IF MØRE THAN ØNE EXPØNENT FIELD        EAD  306
         CLI   1)RC*,BLANK    CHECK FØR BLANK AFTER D ØR E                 EAD  307
         BNE   S5041                                                       EAD  308
         MVI   1)RC*,PLUS026  INSERT PLUS SIGN ØVER BLANK                  EAD  309
S5041    LR    R8,RC          PICK UP LØCATIØN ØF EXPØNENT                 EAD  310
         CHECK RS,NZ,NEXTC    SEE IF DECIMAL PØINT HAS BEEN FØUND          EAD  311
         LR    RS,RC          IF NØT, ASSUME IT IS TØ LEFT ØF EXPØNENT     EAD  312
         B     NEXTC          EXAMINE NEXT CHARACTER                       EAD  313
INT4     MVI   STØREI,X-50-   CØNVERT STØRE INSTRUCTIØN TØ -ST-            EAD  314
         B     STØREI                                                      EAD  315
S521     CHECK R8,NZ,NEXTC    BRANCH IF E ØR D HAS BEEN SEEN               EAD  316
         B     S5041                                                       EAD  317
S502     CHECK RS,NZ,ERRØR    ERRØR IF MØRE THAN ØNE DECIMAL PØINT         EAD  318
         LR    RS,RC          SET DECIMAL PØINT LØCATØR                    EAD  319
         B     NEXTC                                                       EAD  320
GR9      CØL72 H,INTG         MØRE THAN NINE DIGITS TØ LEFT ØF DECIMAL     EAD  321
         CØMP  BLANK,E,INTG                                                EAD  322
         CØMP  LP026,E,LEFTP1                                              EAD  323
         CØMP  LP029,E,LEFTP1                                              EAD  324
         CØMP  RP026,E,RIGHTP1                                             EAD  325
         CØMP  RP029,E,RIGHTP1                                             EAD  326
         CØMP  CØMMA,NE,ERRØR                                              EAD  327
INTG     S     RC,ØNE                                                      EAD  328
         B     INTGER                                                      EAD  329
         EJECT                                                             EAD  330
*        THE PRØCESSING ØF FLØATING PØINT DATA                             EAD  331
         SPACE 2                                                           EAD  332
*        ADDRESS ØF FIRST NUMERICAL CHARACTER IN RT                        EAD  333
*        ADDRESS ØF NEXT BYTE AFTER LAST CHARACTER IN RC                   EAD  334
*        ADDRESS ØF DECIMAL PØINT IN RS                                    EAD  335
*        ADDRESS ØF EXPØNENT IN R8                                         EAD  336
*        IF NØ EXPØNENT, R8 IS SET TØ ZERØ                                 EAD  337
*        FLØATING PØINT ANSWER IN REGISTER 2                               EAD  338
FLØATL   CHECK R5,Z,FLØATLP   IS THIS THE FIRST LEFT PARENTHESIS           EAD  339
         BP    ERRØR          UNBALANCED PARENTHESES                       EAD  340
MULTINC  LA    R5,1           SET INCREMENTING FLAG                        EAD  341
         B     FLØATN-4                                                    EAD  342
FLØATLP  MVI   STØREF.1,X-F0-                                              EAD  343
         LA    R5,1           SET INCREMENTING FLAG                        EAD  344
         B     FLØATN                                                      EAD  345
FLØATR   CHECK R5,NP,ERRØR    ERRØR IF NØ LEFT PARENTHESIS HAS PRECEDED    EAD  346
         LCR   R5,R5          CHANGE SIGN ØF INCREMENTING FLAG             EAD  347
         MVI   STØREF.5,X-F0-                                              EAD  348
         B     FLØATN                                                      EAD  349
FLØAT    CHECK R5,Z,FLØATN    CHECK FØR INCREMENTING                       EAD  350
         BP    ERRØR          UNBALANCED PARENTHESES                       EAD  351
         SR    R5,R5          ZERØ ØUT INCREMENTING FLAG                   EAD  352
         MVI   STØREF.9,X-F0-                                              EAD  353
FLØATN   ST    RC,CØLNØ       SAVE CARD CØLUMN CØUNTER                     EAD  354
         SDR   0,0                                                         EAD  355
```

```
         CHECK R8,NZ,EXPON    CHECK FOR EXPONENT IN FIELD                 EAD 356
GOON     CR    RS,RT                                                     EAD 357
         BL    LTONE                                                     EAD 358
         ST    RC,COL                                                    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,COL                                                    EAD 363
LTONE    LR    RW,RC                                                     EAD 364
         LA    RT,1)0,RS*                                                EAD 365
         LDR   2,0                                                       EAD 366
         SR    RW,RT                                                     EAD 367
         BNP   EXPF                                                      EAD 368
         C     RW,NINE                                                   EAD 369
         BH    MORE                                                      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      NOP   MORE1                                                     EAD 377
MR2      NOP   MORE2                                                     EAD 378
EXPF     NOP   EXPN                                                      EAD 379
         B     STOREF                                                    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     GOON                                                      EAD 385
MORE     MVI   MR1.1,X-F0-                                               EAD 386
         ST    RC,COL                                                    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
MORE1    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,COL                                                    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
MORE2    MVI   MR2.1,X-00-                                               EAD 407
         L     RC,COL                                                    EAD 408
         B     EXPF                                                      EAD 409
REAL4    MVI   STOREFL,X-70- CONVERT STORE INSTRUCTION TO -STE-          EAD 410
         B     STOREFL                                                   EAD 411
NEXTE    COL72 H,ERROR                                                   EAD 412
         B     NXTE                                                      EAD 413
POS      MVI   SIGN.1,X-C0-                                              EAD 414
```

```
NEXTN    COL72 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,0)0,RC*                                               EAD 424
         COMP  D,E,NEXTE                                                EAD 425
         COMP  E,E,NEXTE                                                EAD 426
NXTE     COMP  ZERO,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,COLNO      RESTORE CARD COLUMN COUNTER                EAD 433
         BAL   14,LA000C     CONVERT EXPONENT TO A BINARY INTEGER       EAD 434
         ST    RW,D8                                                    EAD 435
         ST    RA,COL                                                   EAD 436
         LA    1,ARGLIST                                                EAD 437
         STD   2,DOUBLE                                                 EAD 438
         L     15,VFDXPI                                                EAD 439
         BALR  14,15                                                    EAD 440
         LD    2,DOUBLE                                                 EAD 441
         L     RA,COL                                                   EAD 442
         MDR   2,0                                                      EAD 443
STOREF   NOP   FLTLP                                                    EAD 444
         NOP   FLTRP                                                    EAD 445
         NOP   FINCREM                                                  EAD 446
         CLI   LENGTH.1,6    CHECK LENGTH OF VARIABLE                   EAD 447
         BL    REAL4                                                    EAD 448
         MVI   STOREFL,X-60- CONVERT STORE INSTRUCTION TO -STD-         EAD 449
STOREFL  STD   2,0)0,RA*     STORE FLOATING POINT VALUE                 EAD 450
         AR    RA,RI         INCREMENT STORAGE ADDRESS                  EAD 451
DASHF    B     NEWFIELD                                                 EAD 452
DASHF1   NOP   DASHF2                                                   EAD 453
         LH    RS,NMAX                                                  EAD 454
         SH    RS,NMIN                                                  EAD 455
         A     RS,ONE                                                   EAD 456
         MVI   DASHF1.1,X-F0-                                           EAD 457
DASHF2   BCT   RS,STOREFL                                               EAD 458
         MVI   DASHF1.1,X-00-                                           EAD 459
         B     DASHDONE                                                 EAD 460
FLTLP    MVI   STOREF.1,X-00-                                           EAD 461
         LDR   6,2           PICK UP FSTART                             EAD 462
         COL72 H,ERROR                                                  EAD 463
         B     NEWFIELD                                                 EAD 464
FLTRP    MVI   STOREF.5,X-00-                                           EAD 465
         LDR   4,2           PICK UP FINCRM                             EAD 466
         COL72 H,ERROR                                                  EAD 467
         B     NEWFIELD                                                 EAD 468
FINCREM  MVI   STOREF.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,CFLØAT        CØNVERT TØ AN INTEGER                          EAD  474
         STD    0,D8                                                           EAD  475
         L      15,D8,4                                                        EAD  476
         CHECK  R5,P,MULTF      CHECK FØR MULTIPLE INCREMENTING                EAD  477
         LA     15,1)0,15*      STANDARD INCREMENTING, SØ STØRE FSTØP ALSØ     EAD  478
MULTF    CLI    LENGTH,1,6      CHECK LENGTH ØF VARIABLE                       EAD  479
         BL     RINC4                                                          EAD  480
         MVI    LPINCF,X-60-    CØNVERT STØRE INSTRUCTIØN TØ -STD-             EAD  481
LPINCF   STD    6,0)0,RA*       STØRE FLØATING PØINT VALUE                     EAD  482
         ADR    6,4             INCREMENT FLØATING PØINT VALUE                 EAD  483
         AR     RA,RI           INCREMENT STØRAGE ADDRESS                      EAD  484
         BCT    15,LPINCF                                                      EAD  485
         CHECK  R5,P,FLTLP,4    CHECK FØR MULTIPLE INCREMENTING                EAD  486
         CØL72  H,NEXTCARD                                                     EAD  487
         B      NEWFIELD                                                       EAD  488
RINC4    MVI    LPINCF,X-70-    CØNVERT STØRE INSTRUCTIØN TØ -STE-             EAD  489
         B      LPINCF                                                         EAD  490
         EJECT                                                                 EAD  491
*        THE PRØCESSING ØF 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
ACARD1   LA     R8,1            LØØK FØR 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   LP026,E,SUBSCRPT                                               EAD  511
         CØMP   LP029,E,SUBSCRPT                                               EAD  512
         B      ERRØR                                                          EAD  513
S323     LR     RT,RC           PRØCESS 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       CØNVERT BCD TØ BINARY INTEGER                  EAD  521
         STH    RW,NUMWD        SAVE AS NUMBER ØF WØRDS                        EAD  522
         MH     RW,LENGTH       MULTIPLY BY LENGTH ØF VARIABLE                 EAD  523
         C      RW,C72          CØMPARE RESULT WITH 72                         EAD  524
         BH     ERRØR           ERRØR 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,ONE                                                      EAD 535
         STC   RT,MVA+1                                                    EAD 536
         LH    RT,NUMWD                                                    EAD 537
MVA      MVC   0(0,RA),0(RC)                                               EAD 538
         AR    RC,RW                                                       EAD 539
         AR    RA,RI                                                       EAD 540
         BCT   RT,MVA                                                      EAD 541
         MVI   NFLAG,X'00' RESET A-CARD FLAGS                              EAD 542
         MVI   NFLAG1,X'00'                                                EAD 543
         MVI   NFLAG2,X'00'                                                EAD 544
         MVI   NFLAG3,X'00'                                                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+1(79),1(RC) MOVE TO OUTPUT FORMAT FOR 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,X'00'                                                EAD 555
         LA    R8,1                                                        EAD 556
         MVI   EITHER+1,X'F0'                                              EAD 557
         LR    RW,RC                                                       EAD 558
         A     RW,C7                                                       EAD 559
         MVI   S420+1,X'00'                                                EAD 560
         LR    RT,RC                                                       EAD 561
MOREL    BXLE  RC,R8,*+8     CHECK FOR COLUMN 72                           EAD 562
         B     S420                                                        EAD 563
         CR    RC,RW         SEE IF EIGHT CHARACTERS HAVE BEEN SEEN        EAD 564
         BH    S405                                                        EAD 565
PASS1    COMP  BLANK,E,S420  LOOK FOR A FIELD TERMINATOR                   EAD 566
         COMP  EQ026,E,S420                                                EAD 567
         COMP  EQ029,E,S420                                                EAD 568
         COMP  LP026,E,S419                                                EAD 569
         COMP  LP029,E,S419                                                EAD 570
EITHER   B     MOREL                                                       EAD 571
EXTRA    BXLE  RC,R8,PASS1   CHECK FOR COLUMN 72                           EAD 572
         B     S420                                                        EAD 573
S405     MVI   EITHER+1,X'00'  EIGHT CHARACTERS HAVE BEEN SEEN             EAD 574
         MVI   S420+1,X'F0'                                                EAD 575
         MVI   MVC1+1,7                                                    EAD 576
         B     PASS1                                                       EAD 577
S419     MVI   LPAREN,X'F0'  LEFT PARENTHESIS TERMINATES NAME              EAD 578
S420     NOP   S4200                                                       EAD 579
         LR    RW,RC                                                       EAD 580
         SR    RW,RT                                                       EAD 581
         S     RW,ONE                                                      EAD 582
         STC   RW,MVC1+1                                                   EAD 583
S4200    MVC   D8(8),BLANKS                                                EAD 584
MVC1     MVC   D8(0),0(RT)                                                 EAD 585
LA000L   LH    RT,NUMTAB     LOOK FOR VARIABLE NAME IN TABLE               EAD 586
         L     RW,STABLE                                                   EAD 587
LP2      CLC   0(8,RW),D8                                                  EAD 588
         BE    FOUND                                                       EAD 589
         LA    RW,28(0,RW)                                                 EAD 590
         BCT   RT,LP2                                                      EAD 591
```

```
         B      ERRØR2                                                EAD  592
FØUND    L      RT,8)0,RW*      NAME HAS BEEN FØUND                   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,NØSUBS     SEE IF VARIABLE IS AN ARRAY           EAD  601
         LH     RI,LENGTH                                             EAD  602
         SLA    RT,1                                                  EAD  603
         ST     RT,NUMSCR2                                            EAD  604
NØSUBS   L      RA,NADD         PICK UP STØRAGE ADDRESS               EAD  605
LPAREN   NØP    S423            BRANCH ØNLY IF LEFT PARENTHESIS WAS SEEN EAD  606
NFLAG1   NØP    ACARD1                                                EAD  607
         B      NEWFIELD                                              EAD  608
S423     CHECK  RT,Z,ERRØR      ERRØR IF VARIABLE IS NØT AN ARRAY     EAD  609
         EJECT                                                        EAD  610
*        THE PRØCESSING ØF SUBSCRIPT FIELDS                           EAD  611
         SPACE  2                                                     EAD  612
SUBSCRPT MVI    DASHDNE.1,X-FO-                                       EAD  613
         B      DASHDØNE                                              EAD  614
SUBSC    MVI    DØNE1.1,X-DO-                                         EAD  615
         MVI    SIGN.1,X-CO-                                          EAD  616
         MVI    DNE.1,X-FO-                                           EAD  617
         SR     RS,RS                                                 EAD  618
SUBLØØP  CØL72  H,ERRØR                                               EAD  619
         CØMP   BLANK,E,SUBLØØP                                       EAD  620
         CØMP   ZERØ,L,ERRØR                                          EAD  621
         LR     RT,RC                                                 EAD  622
NEXTNS   CØL72  H,ERRØR                                               EAD  623
         CØMP   ZERØ,NL,NEXTNS                                        EAD  624
         CØMP   CØMMA,E,ENDSUB                                        EAD  625
         CØMP   RP026,E,ESUBS                                         EAD  626
         CØMP   RP029,E,ESUBS                                         EAD  627
         CØMP   BLANK,E,SKIPS                                         EAD  628
         CØMP   MINUS,NE,ERRØR                                        EAD  629
S1048    NØP    ERRØR                                                 EAD  630
         MVI    S1048.1,X-FO-                                         EAD  631
         MVI    ND1.1,X-FO-                                           EAD  632
         B      ENDSUB                                                EAD  633
ESUBS    MVI    DNE.1,X-00-                                           EAD  634
ENDSUB   NØP    S1046                                                 EAD  635
         LA     RS,2)0,RS*                                            EAD  636
         C      RS,NUMSCR2                                            EAD  637
         BH     ERRØR                                                 EAD  638
         BAL    14,LA000C                                             EAD  639
         STH    RW,SUB-2)RS*                                          EAD  640
ND1      NØP    S1049                                                 EAD  641
BLNK     NØP    LCØM                                                  EAD  642
DNE      NØP    SUBLØØP                                               EAD  643
PAR      SRA    RS,1                                                  EAD  644
         LR     RW,RS                                                 EAD  645
ND2      NØP    S1047                                                 EAD  646
ØUTSUB   CØL72  H,SUBDØNE                                             EAD  647
         CØMP   BLANK,E,SUBDØNE                                       EAD  648
         CØMP   EQ026,E,SUBDØNE                                       EAD  649
         CØMP   EQ029,E,SUBDØNE                                       EAD  650
```

```
         CLI   0(RC*,ZERO                                          EAD 651
DONE1    BNH   ERROR                                               EAD 652
         NI    0(RC*,X-0F-                                         EAD 653
         SR    RW,RW                                               EAD 654
         IC    RW,0(0,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  RP026,E,PAR                                         EAD 666
         COMP  RP029,E,PAR                                         EAD 667
         COMP  MINUS,E,S1045                                       EAD 668
         B     ERROR                                               EAD 669
SUBDONE  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
SLOOP    LH    R8,SUB,2(RW*                                        EAD 677
         S     R8,ONE                                              EAD 678
         MH    R8,NS(RW*                                           EAD 679
         AR    RT,R8                                               EAD 680
         LA    RW,2(0,RW*                                          EAD 681
         BCT   RS,SLOOP                                            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,NS(RW*                                           EAD 691
NFLAG2   NOP   ACARD1                                              EAD 692
         COL72 H,NEXTCARD                                          EAD 693
         B     NEWFIELD                                            EAD 694
S1045    EX    0,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    S1048.1,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
DASHDØNE 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  NØP    SUBSC                                             EAD  725
         B      NEWFIELD                                          EAD  726
         EJECT                                                    EAD  727
*        THE PRØCESSING ØF HEXADECIMAL DATA                       EAD  728
         SPACE 2                                                  EAD  729
HEXCARD  CØL72  H,READCARD                                        EAD  730
         CØMP   BLANK,E,HEXCARD                                   EAD  731
         LR     RT,RC                                             EAD  732
STAY     CØMP   ZERØ,NL,NUMB                                      EAD  733
         CØMP   F,H,ERRØR                                         EAD  734
         CØMP   A,L,ERRØR                                         EAD  735
         IC     RW,010,RC*                                        EAD  736
         AH     RW,CONV      CØNVERT -C1- - -C6- TØ -FA- - -FF-   EAD  737
         STC    RW,010,RC*                                        EAD  738
NUMB     CØL72  H,STØREH                                          EAD  739
         CØMP   BLANK,NE,STAY                                     EAD  740
STØREH   LR     RW,RC                                             EAD  741
         SR     RW,RT                                             EAD  742
         LH     R8,LENGTH                                         EAD  743
         SLA    R8,1                                              EAD  744
         CR     RW,R8                                             EAD  745
         BH     ERRØR                                             EAD  746
         SDR    0,0                                               EAD  747
         STD    0,D8                                              EAD  748
         STD    0,DØUBLE                                          EAD  749
         S      RW,ØNE                                            EAD  750
         STC    RW,LENG.1                                         EAD  751
         NI     PK.1,X-F0-                                        EAD  752
LENG     ØI     PK.1,X-00-                                        EAD  753
PK       PACK   D8-119*,010,RT* CØNVERT BCD TØ HEXADECIMAL        EAD  754
         LM     RT,RW,D8                                          EAD  755
         SRDL   RT,4         SHIFT ØUT SIGN                       EAD  756
         STM    RT,RW,D8                                          EAD  757
         IC     RW,D8-1      INSERT FIRST HALF-BYTE               EAD  758
         SLL    RW,4                                              EAD  759
         STC    RW,LMØST.1                                        EAD  760
LMØST    ØI     D8,X-00-                                          EAD  761
         L      RS,D8,4                                           EAD  762
         LD     4,D8                                              EAD  763
         CLI    LENGTH.1,6   CHECK LENGTH ØF VARIABLE             EAD  764
         BL     TIMES4                                            EAD  765
         MVI    STØRHEX,X-60- CØNVERT STØRE INSTRUCTIØN TØ -STD-  EAD  766
STØRHEX  STD    4,010,RA*    STØRE HEXADECIMAL DATA               EAD  767
         AR     RA,RI        INCREMENT STØRAGE ADDRESS            EAD  768
```

```
DASHH    B      NEXTH
DASHH1   NØP    DASHH2
         LH     RW,NMAX
         SH     RW,NMIN
         A      RW,ØNE
         MVI    DASHH1,1,X-F0-
DASHH2   BCT    RW,STØRHEX
         MVI    DASHH1,1,X-00-
         MVI    DASHA,1,X-F0- RESET DASH SWITCHES
         MVI    DASHL,1,X-F0-
         MVI    DASHH,1,X-F0-
         MVI    DASHF,1,X-F0-
         MVI    DASHI,1,X-F0-
NEXTH    CR     RC,RCMAX
         BH     READCARD
         B      HEXCARD
TIMES4   MVI    STØRHEX,X-50- CØNVERT STØRE INSTRUCTIØN TØ -STE-
         B      STØRHEX
         EJECT
*        THE PRØCESSING ØF ALPHAMERIC DATA
         SPACE 2
ALPHA    LH     RS,LENGTH      PICK UP LENGTH ØF VARIABLE
         S      RS,ØNE
         STC    RS,MVS,1
         STC    RS,MVS1,1
         CØL72  H,ERRØR
         CØMP   BLANK,E,NØTHING
         LR     RT,RC
         LR     RW,RC
         AR     RW,RS
MØREA    CØL72  H,STØREA
         CØMP   BLANK,E,STØREA
         CR     RC,RW
         BH     ERRØR
         B      MØREA
STØREA   LR     RW,RC
         SR     RW,RT
         S      RW,ØNE
         STC    RW,MVS1,1
MVS      MVC    010,RA*,BLANKS
MVS1     MVC    010,RA*,01RT* STØRE ALPHAMERIC DATA
NEXTA    AR     RA,RI          INCREMENT STØRAGE ADDRESS
DASHA    B      NEWFIELD
DASHA1   NØP    DASHA2
         LH     RS,NMAX
         SH     RS,NMIN
         A      RS,ØNE
         MVI    DASHA1,1,X-F0-
DASHA2   BCT    RS,MVS
         MVI    DASHA1,1,X-00-
         B      DASHDØNE
NØTHING  LA     RT,BLANKS
         B      MVS1
         EJECT
*        THE PRØCESSING ØF LØGICAL DATA
         SPACE 2
LØGICAL  CØMP   T,E,TRUE       CHECK FØR -TRUE-
         CØMP   F,E,LØGIC-2    CHECK FØR -FALSE-
         CØMP   Y,E,TRUE       CHECK FØR -YES-
```

```
         CØMP  N,NE,ERRØR   ERRØR IF NØT -NØ-                          EAD  828
         SR    RW,RW                                                   EAD  829
LØGIC    CØL72 H,STØREL                                                EAD  830
         CØMP  BLANK,E,STØREL  LØØK FØR A FIELD TERMINATØR             EAD  831
         CØMP  CØMMA,E,STØREL                                          EAD  832
         B     LØGIC                                                   EAD  833
TRUE     LA    RW,1                                                    EAD  834
         B     LØGIC                                                   EAD  835
TIMES4L  MVI   STØRLØG,X-50- CØNVERT STØRE INSTRUCTIØN TØ -STE-        EAD  836
         B     STØRLØG                                                 EAD  837
STØREL   CLI   LENGTH,1,3    CHECK LENGTH ØF VARIABLE                  EAD  838
         BH    TIMES4L                                                 EAD  839
         MVI   STØRLØG,X-42- CØNVERT STØRE INSTRUCTIØN TØ -STC-        EAD  840
STØRLØG  ST    RW,0)0,RA*    STØRE LØGICAL VALUE                       EAD  841
         AR    RA,RI         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-F0-                                          EAD  848
DASHL2   BCT   RS,STØRLØG                                              EAD  849
         MVI   DASHL1,1,X-00-                                          EAD  850
         B     DASHDØNE                                                EAD  851
         EJECT                                                         EAD  852
*        THIS RØUTINE CØNVERTS BCD TØ A SIGNED BINARY INTEGER          EAD  853
         SPACE 2                                                       EAD  854
*        ADDRESS ØF FIRST NUMERICAL CHARACTER IN RT                    EAD  855
*        ADDRESS ØF NEXT BYTE AFTER LAST CHARACTER IN RC               EAD  856
*        BINARY INTEGER ANSWER IN RW                                   EAD  857
*        TØ INSERT MINUS SIGN, USE MVI    SIGN,1,X-D0-                 EAD  858
*        TØ INSERT PLUS SIGN, USE MVI    SIGN,1,X-C0-                  EAD  859
LA000C   LR    RW,RC                                                   EAD  860
         SR    RW,RT                                                   EAD  861
         S     RW,ØNE                                                  EAD  862
         STC   RW,LEN,1      INSERT FIELD LENGTH INTØ PACK INSTRUCTIØN EAD  863
         NI    PACK,1,X-F0-                                            EAD  864
LEN      ØI    PACK,1,X-00-                                            EAD  865
         LR    RW,RC         GET ADDRESS ØF RIGHTMØST BYTE             EAD  866
         S     RW,ØNE                                                  EAD  867
         NI    0)RW*,X-0F-   CLEAR ZØNE FØR SIGN                       EAD  868
SIGN     ØI    0)RW*,X-C0-   INSERT SIGN                               EAD  869
         SDR   0,0                                                     EAD  870
         STD   0,D8                                                    EAD  871
PACK     PACK  D8)8*,0)0,RT* PACK BCD                                  EAD  872
         CVB   RW,D8         CØNVERT TØ BINARY                         EAD  873
         BR    14            RETURN                                    EAD  874
         EJECT                                                         EAD  875
*        THIS RØUTINE CØNVERTS A SIGNED BINARY INTEGER TØ DØUBLE       EAD  876
*        PRECISIØN FLØATING PØINT, USING THE SAME PRØCEDURE AS FØRTRAN EAD  877
         SPACE 2                                                       EAD  878
*        BINARY INTEGER IN RW                                          EAD  879
*        FLØATING PØINT ANSWER IN REGISTER 0                           EAD  880
LA000F   LD    0,CFLØAT                                                EAD  881
         STD   0,D8                                                    EAD  882
         LTR   RW,RW                                                   EAD  883
         BM    *,16                                                    EAD  884
         ST    RW,D8,4                                                 EAD  885
         AD    0,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
*        ERRØR PRØCESSING                                              EAD 893
         SPACE 2                                                       EAD 894
ERRØR1   LA     RW,FE1         ILLEGAL ENTRY TØ SETUP ØR READIN        EAD 895
         ST     RW,FERRØR                                              EAD 896
         B      ERR                                                    EAD 897
ERRØR2   MVC    NAME)8*,D8     NAME ØF VARIABLE NØT FØUND IN TABLE     EAD 898
         LA     RW,FE2                                                 EAD 899
         ST     RW,FERRØR                                              EAD 900
         B      ERR                                                    EAD 901
ERRØR    LA     RW,FE          ERRØR IN FØRMAT ØN LAST CARD            EAD 902
         ST     RW,FERRØR                                              EAD 903
ERR      L      15,VIBCØM      WRITE ERRØR MESSAGE                     EAD 904
         CNØP   0,4                                                    EAD 905
         BAL    14,4)0,15*                                             EAD 906
         DC     F-6-                                                   EAD 907
FERRØR   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-ØF-DATA PRØCESSING                                        EAD 913
         SPACE 2                                                       EAD 914
IEND     B      LEAVE          END-ØF-JØB CARD                         EAD 915
         LA     15,4           SET UP NØN-STANDARD RETURN              EAD 916
         B      RESTØRE,2                                              EAD 917
RESTØRE  SR     15,15          NØRMAL RETURN TØ CALLING PRØGRAM        EAD 918
         L      13,SAVE,4      RESTØRE ØLD SAVE AREA ADDRESS           EAD 919
         L      RW,ADINCØL1                                            EAD 920
         L      RT,INCØL1                                              EAD 921
         ST     RT,0)0,RW*                                             EAD 922
         LM     2,12,28)13*    RESTØRE 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
         DRØP   RD             DØN-T LET ASSEMBLER USE RD AS A BASE    EAD 927
         EJECT                                                         EAD 928
*        SUBRØUTINE HEDING)NTAPE*                                      EAD 929
*        THIS RØUTINE EJECTS A PAGE AND WRITES THE H-CARD              EAD 930
         SPACE 2                                                       EAD 931
         USING  *,12           INFØRM ASSEMBLER ØF 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 ØF ØLD SAVE AREA           EAD 935
         L      13,0)0,1*      PICK UP ADDRESS ØF TAPE NUMBER          EAD 936
         L      13,0)0,13*     PICK UP TAPE NUMBER                     EAD 937
         ST     13,TAPENØ      INSERT IT IN ARG LIST FØR IBCØM         EAD 938
         LA     13,SAVE        PICK UP ADDRESS ØF NEW SAVE AREA        EAD 939
         L      15,VIBCØM                                              EAD 940
         CNØP   0,4                                                    EAD 941
         BAL    14,4)0,15*     FLIP PAGE AND WRITE HEADING             EAD 942
TAPENØ   DC     F-6-                                                   EAD 943
         DC     A)FH*                                                  EAD 944
         BAL    14,16)0,15*                                            EAD 945
```

```
         L        13,SAVE,4        PICK UP ADDRESS ØF ØLD SAVE AREA
         LM       12,14,12)13*     RESTØRE REGISTERS
         MVI      12)13*,X-FF-     INSERT RETURN FLAG
         BR       14               RETURN
         EJECT
*        SUBRØUTINE WHERE)TABLE*
*        THIS RØUTINE PICKS UP THE LØCATIØN ØF -TABLE- FØR -SETUP-
*         -WHERE- HAS BEEN MØDIFIED 5-2-68
*         TØ ACTIVATE MIDCØM. )W. SØHN*
*
         EXTRN MIBCØM
WHERE    BC       15,12)15*
         DC       X-7-
         DC       CL7-WHERE  -
*        SAVE PRØCEDURE
         STM      14,12,12)13*
         BALR     10,0
         USING    *,10
         LR       2,13
         LA       13,SAVEREG
         ST       13,8)2*
         ST       2,4)13*
         BC       15,FIRST
SAVEREG  DS       18F
*        END SAVE PRØCEDURE
FIRST    NØP      NOTFIRST
         L        RT,0)1*
         ST       RT,TABLE
         ST       RT,STABLE
         MVI      FIRST,1,X-F0-
         LA       1,ARGULST
         L        15,ADCØN
         BALR     14,15
*        RETURN PRØCEDURE
NØTFIRST L        13,SAVEREG,4
         LM       2,12,28)13*
         L        14,12)13*
         MVI      12)13*,X-FF-
         BCR      15,14
*        END RETURN PRØCEDURE
ADCØN    DC       A)MIBCØM*
*        ARGUMENT LIST
ARGULST  DC       X-80-
         DC       AL3)**
*        END SUBRØUTINE WHERE
         EJECT
*        SUBRØUTINE SETUP)NAME,LENGTH,VARIABLE,DIMENSIØNS,...*
*        THIS RØUTINE CØNSTRUCTS THE TABLE ØF INPUT VARIABLES
*   WØRD                      CØNTENTS
*        ********************************************************
*        *                                                      *
*   1    *                                                      *
*        *             EBCDIC NAME ØF VARIABLE                  *
*   2    *                                                      *
*        *                                                      *
*        ********************************************************
*        *                                                      *
*   3    *             CØRE ADDRESS ØF VARIABLE                 *
*        *                                                      *
```

```
*        *****************************************************************
*        *                                                               *
*   4    *    LENGTH SPECIFICATION      *    NUMBER ØF SUBSCRIPTS       *
*        *                                                               *
*        *****************************************************************
*        *                                                               *
*   5    *            N1=NS)1*               *          N1*N2=NS)2*      *
*        *                                                               *
*        *****************************************************************
*        *                                                               *
*   6    *         N1*N2*N3=NS)3*            *     N1*N2*N3*N4=NS)4*     *
*        *                                                               *
*        *****************************************************************
*        *                                                               *
*   7    *    N1*N2*N3*N4*N5=NS)5*      *    N1*N2*N3*N4*N5*N6=NS)6*    *
*        *                                                               *
*        *****************************************************************
         SPACE 2
         USING *,15
SETUP    STM   R6,R12,44)13* SAVE REGISTERS
CHECK    NØP   GØ            BRANCH ARØUND IF NØT THE FIRST TIME THRØUGH
         L     R11,STABLE
         CHECK R11,Z,ERRS    SEE IF -WHERE- HAS BEEN ENTERED
         MVI   CHECK,1,X-F0-
GØ       L     R9,TABLE      PICK UP LØCATIØN FØR NEXT TABLE ENTRY
         L     R11,0)0,1*    PICK UP ADDRESS ØF VARIABLE NAME
         MVC   0)8,R9*,0)R11* MØVE NAME INTØ TABLE
         L     R11,4)0,1*
         L     R11,0)0,R11*  PICK UP LENGTH ØF VARIABLE
         STH   R11,12)0,R9*  INSERT IT INTØ TABLE
         L     R11,8)0,1*    PICK UP ADDRESS ØF VARIABLE
         ST    R11,8)0,R9*   INSERT IT INTØ TABLE
         SR    R12,R12
         CLI   8)1*,X-80-    LEAVE IF IT WAS THE LAST ARGUMENT
         BE    ALLDØNE
         LR    R7,1
         L     R11,ØNE
         LA    R6,14
LØØP     LA    R12,1)0,R12*
         CLI   12)R7*,X-80-  LEAVE IF THIS IS THE LAST ARGUMENT
         BE    ALLDØNE
         C     R12,C7        LEAVE IF SEVEN SUBSCRIPTS
         BE    ALLDØNE
         L     R8,12)0,R7*   PICK UP DIMENSIØN
         MH    R11,2)0,R8*   CØMPUTE DIMENSIØN FACTØR
         LA    R7,4)0,R7*
         LA    R6,2)0,R6*
         STH   R11,0)R6,R9*  INSERT DIMENSIØN FACTØR INTØ TABLE
         B     LØØP
ALLDØNE  STH   R12,14)0,R9*  INSERT NUMBER ØF SUBSCRIPTS INTØ TABLE
         LA    R9,28)0,R9*   ØBTAIN LØCATIØN ØF NEXT TABLE ENTRY
         ST    R9,TABLE
         LH    R11,NUMTAB    INCREMENT NUMBER ØF TABLE ENTRIES
         LA    R11,1)0,R11*
         STH   R11,NUMTAB
         LM    R6,R12,44)13* RESTØRE REGISTERS
         BR    14            RETURN
ERRS     L     12,AREAD      ERRØR IF WHERE NØT ENTERED BEFØRE SETUP
         L     RD,AREAD,4
```

```
         USING READIN,12                                                 EAD 1065
         B     ERRØR1                                                    EAD 1066
         DRØP  12                                                        EAD 1067
         DS    0F                                                        EAD 1068
AREAD    DC    A)READIN*                                                 EAD 1069
         DC    A)DATA*                                                   EAD 1070
         EJECT                                                           EAD 1071
*        REGISTER ASSIGNMENT                                             EAD 1072
         SPACE 2                                                         EAD 1073
RI       EQU   0                                                         EAD 1074
RA       EQU   1                                                         EAD 1075
RT       EQU   2                                                         EAD 1076
RW       EQU   3                                                         EAD 1077
RS       EQU   4                                                         EAD 1078
R5       EQU   5                                                         EAD 1079
RC       EQU   6                                                         EAD 1080
R6       EQU   6                                                         EAD 1081
R7       EQU   7                                                         EAD 1082
R8       EQU   8                                                         EAD 1083
RCMAX    EQU   9                                                         EAD 1084
R9       EQU   9                                                         EAD 1085
RD       EQU   10                                                        EAD 1086
R11      EQU   11                                                        EAD 1087
R12      EQU   12                                                        EAD 1088
         EJECT                                                           EAD 1089
*        BCD AND EBCDIC CHARACTERS                                       EAD 1090
         SPACE 2                                                         EAD 1091
BLANK    EQU   C- -                                                      EAD 1092
A        EQU   C-A-                                                      EAD 1093
D        EQU   C-D-                                                      EAD 1094
E        EQU   C-E-                                                      EAD 1095
F        EQU   C-F-                                                      EAD 1096
H        EQU   C-H-                                                      EAD 1097
N        EQU   C-N-                                                      EAD 1098
P        EQU   C-P-                                                      EAD 1099
T        EQU   C-T-                                                      EAD 1100
X        EQU   C-X-                                                      EAD 1101
Y        EQU   C-Y-                                                      EAD 1102
Z        EQU   C-Z-                                                      EAD 1103
DØLLAR   EQU   C-$-                                                      EAD 1104
STAR     EQU   C-*-                                                      EAD 1105
PLUS026  EQU   C-++-                                                     EAD 1106
PLUS029  EQU   C-.-                                                      EAD 1107
MINUS    EQU   C---                                                      EAD 1108
DECIMAL  EQU   C-.-                                                      EAD 1109
CØMMA    EQU   C-,-                                                      EAD 1110
ZERØ     EQU   C-0-                                                      EAD 1111
EQ026    EQU   C-#-                                                      EAD 1112
EQ029    EQU   C-=-                                                      EAD 1113
LP026    EQU   C-(-                                                      EAD 1114
LP029    EQU   C-)-                                                      EAD 1115
RP026    EQU   C-)-                                                      EAD 1116
RP029    EQU   C-*-                                                      EAD 1117
SLASH    EQU   C-/-                                                      EAD 1118
         EJECT                                                           EAD 1119
*        PRØGRAM DATA                                                    EAD 1120
         SPACE 2                                                         EAD 1121
DATA     DS    0D                                                        EAD 1122
CARD     DS    10D            CARD CØLUMNS 1 - 80                        EAD 1123
```

```
CFLØAT    DC    XL8-4E00000000000000- FØR INTEGER TØ FLØATING PØINT   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
DØUBLE    DS    D              THESE TWØ TEMPØRARIES MUST             EAD 1129
DB        DS    D              REMAIN CØNTIGUØUS TØ EACH ØTHER         EAD 1130
ØNE       DC    F-1-                                                   EAD 1131
TWØ       DC    F-2-                                                   EAD 1132
C7        DC    F-7-                                                   EAD 1133
NINE      DC    F-9-                                                   EAD 1134
C72       DC    F-72-                                                  EAD 1135
C8888     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
VIBCØM    DC    A)IBCØM**                                              EAD 1141
INCØL1    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
ADINCØL1  DS    F                                                      EAD 1146
STABLE    DC    F-0-                                                   EAD 1147
TABLE     DC    F-0-                                                   EAD 1148
NADD      DC    F-0-                                                   EAD 1149
CØL       DC    F-0-                                                   EAD 1150
CØLNØ     DC    F-0-                                                   EAD 1151
NUMSCR2   DC    F-0-                                                   EAD 1152
          LTØRG                                                        EAD 1153
CØNV      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
NØFLIP    DC    CL6-NØFLIP-                                            EAD 1164
FLIP      DC    CL4-FLIP-                                              EAD 1165
NØPRINT   DC    CL7-NØPRINT-                                           EAD 1166
PRINTS    DC    CL5-PRINT-                                             EAD 1167
          EJECT                                                        EAD 1168
*         FØRMAT 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-      FØRMAT FØR TIME ØF DAY FIELD            EAD 1179
HRS       DS    CL2            HØURS                                   EAD 1180
          DC    CL1-.-                                                 EAD 1181
MNS       DS    CL2            MINUTES                                 EAD 1182
```

```
         DC      CL1=.=                                                EAD 1183
SCS      DS      CL2           SECØNDS                                 EAD 1184
         DC      CL1=.=                                                EAD 1185
HSC      DS      CL2           HUNDREDTHS ØF A SECØND                  EAD 1186
         DC      CL5=     =                                            EAD 1187
         DC      XL2=1A08=     FØRMAT FØR DATE FIELD                   EAD 1188
MTH      DS      CL2           MØNTH                                   EAD 1189
         DC      CL1=/=                                                EAD 1190
DAY      DS      CL2           DAY                                     EAD 1191
         DC      CL1=/=                                                EAD 1192
YR       DS      CL2           YEAR                                    EAD 1193
         DC      XL1=22=       END ØF FØRMAT                           EAD 1194
F2       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
         DC      XL1=22=                                               EAD 1200
FE2      DC      XL3=021A1E=                                           EAD 1201
         DC      CL13=0 NØ SYMBØL ==                                   EAD 1202
NAME     DC      CL8=        =                                         EAD 1203
         DC      CL9=== LISTED.=                                       EAD 1204
         DC      XL1=22=                                               EAD 1205
FE1      DC      XL3=021A44=                                           EAD 1206
         DC      CL32=0 WHERE NØT ENTERED BEFØRE SETUP=                EAD 1207
         DC      CL36= ØR SETUP NØT ENTERED BEFØRE READIN.=            EAD 1208
         DC      XL1=22=                                               EAD 1209
FE       DC      XL3=021A1F=                                           EAD 1210
         DC      CL31=0 ERRØR IN FØRMAT ØN LAST CARD.='                EAD 1211
         DC      XL1=22=                                               EAD 1212
         END                                                           EAD 1213
```

```
UPRESET DATA                                                                    PRESE  1
     ERNRTB   0.  2.26D-6   1.7D-5  4.26D-5  1.112D-4  .00033                   PRESE  2
     .001068    .01171   .0804   1000000.                                       PRESE  3
     ERNUTB   0.   10.   14.  18.   22.    26.    30.    1000000.               PRESE  4
     ERNTBL                                                                     PRESE  5
     (1,01)1    100.  1.15D6 4.0D9   2.1D11 7.D11 1.3D12  2.D12  2.D12          PRESE  6
     (1,02)1    100.  1.15D6 4.0D9   2.1D11 7.D11 1.3D12  2.D12  2.D12          PRESE  7
     (1,03)1    100.  2.00D6 2.0D10 7.2D11 3.1D12 7.2D12 1.0D13 1.0D13          PRESE  8
     (1,04)1    100.  4.30D7 1.0D11 3.5D12 1.5D13 4.0D13 7.0D13 7.0D13          PRESE  9
     (1,05)1    100.  6.70D8 7.7D11 2.0D13 8.8D13 2.1D14 3.5D14 3.5D14          PRESE 10
     (1,06)1    100.  1.2D10 9.2D12 1.7D14 7.4D14 1.8D15 3.0D15 3.0D15          PRESE 11
     (1,07)1    100.  6.0D11 1.1D14 1.9D15 7.0D15 1.8D16 2.8D16 2.8D16          PRESE 12
     (1,08)1    100.  2.3D12 8.1D14 1.4D16 5.4D16 1.5D17 2.2D17 2.2D17          PRESE 13
     (1,09)1    100.  1.4D13 5.0D15 9.1D16 3.6D17 1.0D18 2.0D18 2.0D18          PRESE 14
     (1,10)1    100.  1.4D13 5.0D15 9.1D16 3.6D17 1.0D18 2.0D18 2.0D18          PRESE 15
     RSTABL .0 1.E-6 1.E-5  1.E-4  1.E-3  1.E-2  ,1 1.       100.               PRESE 16
    HSTABL 0. 30.(20.)450.  470.  1.E10                                         PRESE 17
     ERTABL                                                                     PRESE 18
 (1,9)1  0.0  0.72E9  0.25E14  0.20E15  0.78E16  0.35E17  0.13E18               PRESE 19
0.34E18  0.62E18  0.90E18  0.14E19  0.17E19  0.21E19  0.25E19  0.36E19          PRESE 20
0.39E19  0.47E19  0.56E19  0.63E19  0.70E19  0.86E19  0.20E19                   PRESE 21
0.12E2.  0.14E20  0.14E20                                                       PRESE 22
   (1,8)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
0.54E15  0.90E15  0.12E16  0.14E16  0.17E16  0.21E16  0.24E16  0.28E16          PRESE 30
0.37E16  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
0.46E14  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
0.35E13  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
0.59E14  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
0.18E12  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
0.39E13  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
0.19E11  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
0.78E10  0.90E10  0.12E11  0.14E11  0.17E11  0.24E11  0.29E11  0.33E11          PRESE 56
0.33E11                                                                         PRESE 57
      (2-30,1)  0.                                                              PRESE 58
      SHCTBL  1. 3.(3.)30.  1.E6                                                PRESE 59
      THTTBL  0. 2.(4.)30.  45.  1.E6                                           PRESE 60
```

```
 TABL                                                                    PRESE 61
(1,1)1  1.0  3.0  6.0  9.0  12.0  15.0  18.0  21.0  24.0  27.0           PRESE 62
30.  100.0                                                               PRESE 63
(1,2)1  1.13  3.11  6.30  9.64  13.1  16.8  20.6  24.6  28.8  33.4       PRESE 64
38.  100.0                                                               PRESE 65
(1,3)1  1.29  3.33  7.00  11.2  16.2  22.0  29.0  37.7  48.6  63.0       PRESE 66
81.0  100.0                                                              PRESE 67
(1,4)1  1.43  3.58  7.84  13.4  21.0  33.4  49.2  80.0  100.0            PRESE 68
  .0  100.0  100.0                                                       PRESE 69
(1,5)1  1.57  3.85  8.89  16.7  29.8  58.0  100.0  100.0  100.0          PRESE 70
  .0  100.0  100.0                                                       PRESE 71
(1,6)1  1.70  4.15  10.2  21.8  51.0  100.0  100.0  100.0  100.0         PRESE 72
  .   100.0  100.0                                                       PRESE 73
(1,7)1  1.84  4.49  12.0  31.6  100.0  100.0  100.0  100.0               PRESE 74
  .0  100.0  100.0  100.0                                                PRESE 75
(1,8)1  2.00  4.88  14.6  53.0  100.0  100.0  100.0  100.0  100.0        PRESE 76
  .   100.0  100.0                                                       PRESE 77
(1,9)1  2.13  5.32  18.4  100.0  100.0  100.0  100.0  100.0  100.0       PRESE 78
  .   100.0  100.0                                                       PRESE 79
(1,10)1  2.76  7.79  100.0  100.0  100.0  100.0  100.0  100.0  100.0     PRESE 80
  .   100.0  100.0                                                       PRESE 81
(1,11)1  100.0  100.0  100.0  100.0  100.0  100.0  100.0  100.0          PRESE 82
  .0  100.0  100.0  100.0                                                PRESE 83
     XDTABL   .0000001  .01   .03   .1   1.E6                            PRESE 84
     YDTABL   0.  15.  20.(20.)160.  1.E6                                PRESE 85
     ZDTABL   0.   .01   .1   1.                                         PRESE 86
 TABL                                                                    PRESE 87
(1,5,1)1  0.0  0.70E1  0.38E3  0.32E10  0.25E12  0.32E13  0.17E14        PRESE 88
 .  4E14  0.24E15  0.44E15  0.44E15                                      PRESE 89
(1,4,1)1  0.0  0.50E1  0.38E3  0.32E10  0.25E12  0.32E13  0.17E14        PRESE 90
 .  4E14  0.24E15  0.44E15  0.44E15                                      PRESE 91
(1,3,1)1  0.0  0.32E2  0.12E4  0.11E10  0.78E11  0.10E13  0.65E13        PRESE 92
 .   E14  0.87E14  0.16E15  0.16E15                                      PRESE 93
(1,2,1)1  0.   0.95E2  0.30E4  0.20E9  0.05E11  0.14E12  0.20E13         PRESE 94
 .   E1   0.22E14  0.38E14  0.38E14                                      PRESE 95
(1,1,1)1  0.   0.90E2  0.30E4  0.20E9  0.05E11  0.14E12  0.20E13         PRESE 96
 .   E1   0.22E14  0.38E14  0.38E14                                      PRESE 97
(1,5,2)1  0.0  1.0E5  3.5E7  2.0E11  1.3E12  2.0E12  4.5E12  1.4E13      PRESE 98
 .  E1   3.0E13  4.0E13                                                  PRESE 99
(1,4,2)1  0.0  1.0E5  3.5E7  2.0E11  1.3E12  2.0E12  4.5E12  1.4E13      PRESE100
2.  E13  3.0E13  4.0E13                                                  PRESE101
(1,3,2)1  0.0  5.5E4  1.9E7  1.1E11  2.3E11  4.5E11  2.0E12  5.0E12      PRESE102
 .  E1   1.5E13  2.5E13                                                  PRESE103
(1,2,2)1  0.0  2.0E4  7.0E6  4.0E10  1.6E11  2.0E11  6.0E11  1.4E12      PRESE104
 .  E1   3.0E12  4.0E12                                                  PRESE105
(1,1,2)1  0.0  2.0E4  7.0E6  4.0E10  1.6E11  2.0E11  6.0E11  1.4E12      PRESE106
2.  E12  3.0E12  4.0E12                                                  PRESE107
(1,5,3)1  0.0  2.4E5  1.5E8  7.6E11  5.6E11  1.3E13  1.6E13  2.1E13      PRESE108
3.  E13  4.0E13  4.5E13                                                  PRESE109
(1,4,3)1  0.0  2.4E5  1.5E8  7.6E11  5.6E12  1.3E13  1.6E13  2.1E13      PRESE110
3.  E13  4.0E13  4.5E13                                                  PRESE111
(1,3,3)1  0.0  1.2E5  7.1E7  3.7E11  2.4E12  5.8E12  6.2E12  7.7E12      PRESE112
1.1E1    1.5E13  2.0E13                                                  PRESE113
(1,2,3)1  0.0  4.5E4  2.7E7  1.4E11  7.1E11  1.7E12  1.7E12  2.0E12      PRESE114
2.7E12   3.5E12  4.0E12                                                  PRESE115
(1,1,3)1  0.0  4.5E4  2.7E7  1.0E11  7.1E11  1.7E12  1.7E12  2.0E12      PRESE116
2.7E12   3.5E12  4.0E12                                                  PRESE117
(1,5,4)1  0.   7.0E5  2.3E8  2.0E12  1.8E13  6.0E13  1.4E14  1.3E14      PRESE118
1.  E14  1.5E14  2.0E14                                                  PRESE119
```

```
(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      PRESF122
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
(9?)  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
ACON 1.                                                                  PRESE135
AKU  50.                                                                 PRESE136
BCON 1.                                                                  PRESE137
B21 1.                                                                   PRESE138
B22 .25                                                                  PRESE139
B23 0.0                                                                  PRESE140
CCON 1.                                                                  PRESE141
CRHOW .75                                                                PRESE142
  DELWT .01 DHCHEM 0.0                                                   PRESE143
RHOSL .08042                                                             PRESE144
RHOW 115.                                                                PRESE145
RT  8.475D5                                                              PRESE146
TABL 15.0.                                                               PRESE147
BZERO  5.8E-21 B2  4.0E-10 B3  2.0 BTWEN  1.0 B24  1.0E-26               PRESE148
DX  50.0 NSTWL  100 ZNUS  2.0E11 CNE  0.0 DSD  0.0 X2BOD  0.0 X3B  0.0 PRESE149
IND2  0 IWPRNT  0 IND  0 IWAKE  2 WKALT  500000.0  0.0 IDBL  4           PRESE150
?ETAZ  22.0  22.0 PHI1 6.0  6.0 PHI2  6.0  6.0 PHI3  6.0  6.0            PRESE151
FRQ1  4.25E8 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
??ALT  1800??.0 XCOM(7)  1.0 ICOM(4)  1  1  1 ACOE  1.0 PFD  0.03        PRESE154
 ?(1-9)  40.?                                                            PRESE155
   I?(22-9?) 0                                                           PRESE156
   I? (7) 136 135 134 XCOM(8) 1.0                                        PRESE157
   ? (7?) = 1                                                            PRESE158
?? ??D OF THE PRESET INPUT DECK
```

<u>APPENDIX II</u> - 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
INFCØF
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 IØP(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 ENDJØB, EZPLØT, FRAMEV, IDFRMV, and PLTND must be removed at AVCO.