AD NUMBER: AD0853819

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to DoD Only; Export Controlled; Jun 1969. Other requests shall be referred to Space and Missile Systems Organization, Norton AFB, CA 92409.

AUTHORITY

SAMSO ltr dtd 19 Jan 1972

ADVANCED DECOY TECHNOLOGY PROGRAM
ADTECH IV
FINAL REPORT (U)

APPENDIX I, PART I
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 I
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, and
3. Studies of the use of certain discrimination techniques for a hard point defense system.

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

TABLE OF CONTENTS

PART I

1.0 INTRODUCTION AND PRELIMINARY OPERATIONS

1.1 Introduction

The Optimum Decoy Design Program selects optimum decoy configurations which meet specified performance, weight, and geometric constraints. The technique used to determine the optimum values of the decoy design parameters is the sequential minimization of a penalty function formulated from the differences between certain characteristics of the reentry vehicle and those of the decoy as well as constraints on the decoy itself.

The four optimization techniques available for determining minima are:

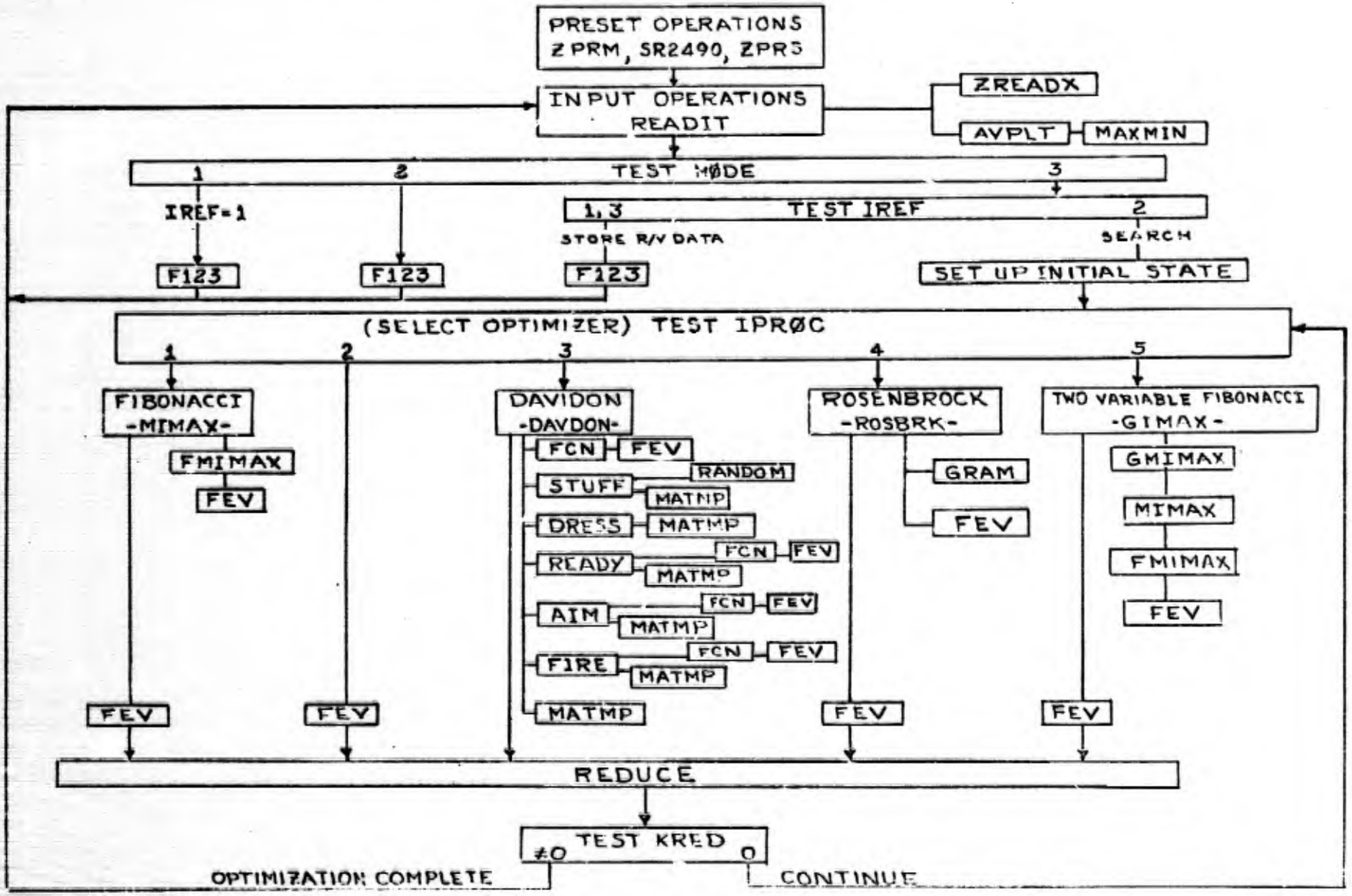
1. A Fibonacci one variable search.
2. A Fibonacci two variable search.
3. The Davidon variable metric method.
4. The Rosenbrock rotating coordinate method.

The function to be minimized is generated using the results of the trajectory calculations (which include the effects of mass loss, noseblunting, and angle of attack), the wake observables approximations, and the effectiveness operations. An option exists which enables the bypassing of the optimization process and the direct evaluation of the penalty function.

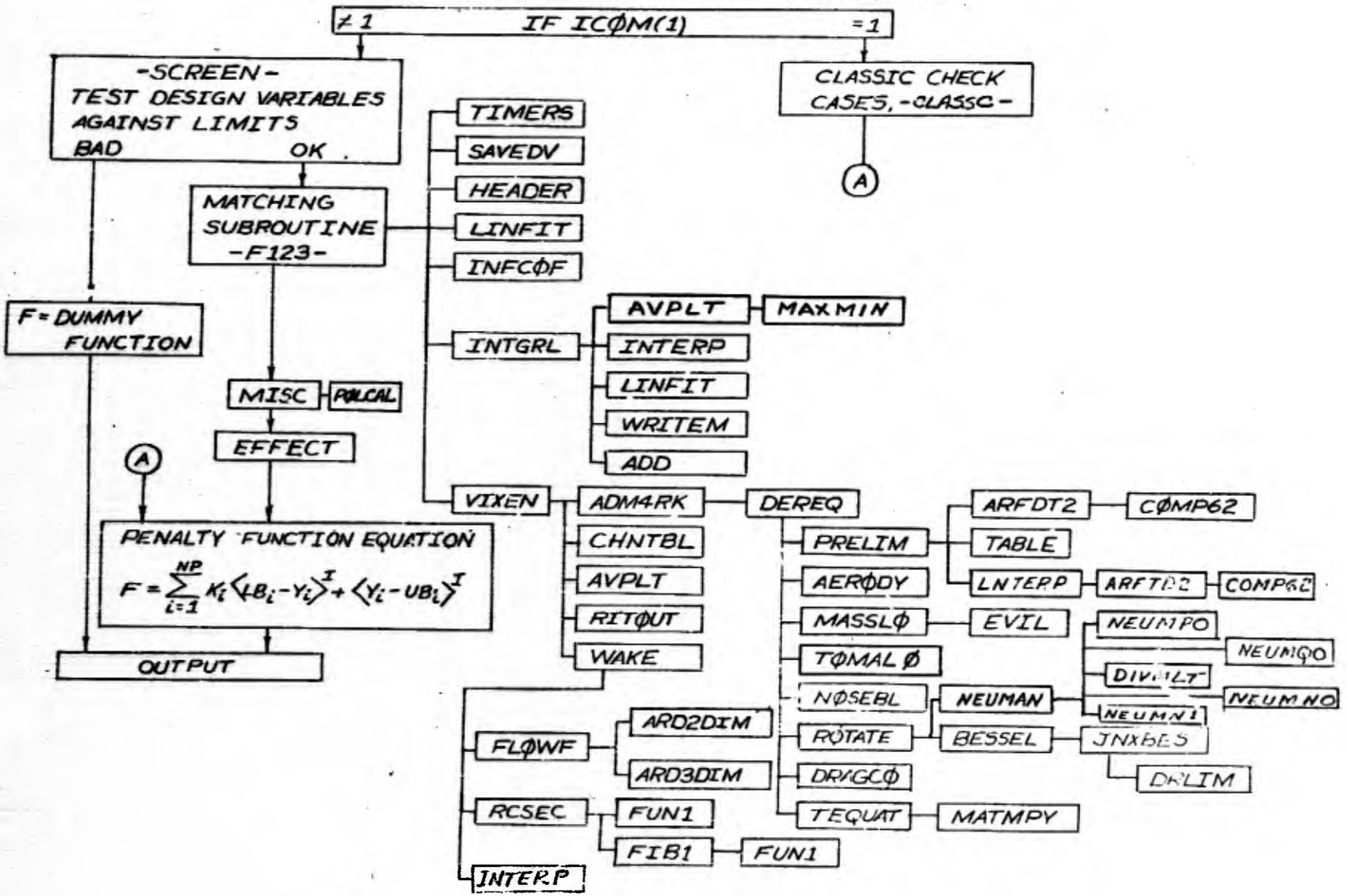
This document contains the description of the numerical methods employed, the correlation of program segments with their functional description, and a complete listing of the source symbolic program and preset deck. This report was prepared by R. A. MacFarlane and E. R. Nickerson with the assistance of J. F. Connors and R. E. Housman.

The organization of the program and the relationships between the various subroutines are shown in the following flow charts of the Main program and the optimization logic of the function evaluator routine.

Flow Chart Of Main Program And Optimization Logic



FLOW CHART OF FUNCTION EVALUATOR, FEV



1.2 MAIN PROGRAM

1. Purpose

MAIN directs the calculation of optimum values for the design parameters of a decoy. The technique used is to minimize the differences between certain characteristics of a reentry vehicle and the same characteristics for a decoy. In this version, the following nine characteristics may be used. Velocity, deceleration, ballistic coefficient, wake length at each of three radar frequencies and wake cross section at each of these frequencies.

Four methods of finding these minima are available.

1. A Fibonacci one variable search.
2. A Fibonacci two variable search.
3. The Davidon method
4. The Rosenbrock rotating coordinate method.

A quantity PD, the probability that a decoy will be discriminated, can also be calculated.

2. Input

*indicates integer quantity

Name	Source	Common Block	Preset Value	Description
ALØW, 20	ZREADX	MIN	0.0	vector of lower bounds
CTP, 20	ZREADX	MIN	0.0	vector of upper bounds
ID1, 50*	ZREADX	IDNØS	0	identification of design variable
ID2, 50*	ZREADX	IDNØS	0	identification of constraints
IMPLØT*	SR2490	IØCCUR(309)	0	test parameter for AVPLT
IN *	ZREADX	IØPT	1	number of parameters being obtained
IØP, 90 *	ZREADX	IØCCUR(1-90)	1	option parameter
IPRØC *	ZREADX	IØPT	1	test parameter for choosing optimizer
IREF *	ZREADX	IØCCUR(301)	1	test parameter for type of vehicle
KRED *	REDUCE	-		test parameter
LIMIT*	ZREADX	IØPT	30	iteration counter
LRED *	ZREADX	ØWL	0	iteration upper limit
MØDE *	ZREADX	IØCCUR(303)	3	test parameter
NPRINT *	READIT	NØCCUR(14)	1	print option
ØCCUR, 4000		ØCCUR		common array
OVECT, 20	ZREADX	ØWL	5.0	initial values of design variables
WRF	ZREADX	ØWL	0.9	reduction factor
X1SAVE, 40	GMIMAX	XXSAVE		optimum values of XX(1)
X2SAVE, 40	GMIMAX	XXSAVE		all values of XX(2)

3. Output

* indicates integer quantity

Name	Common Block	Preset Value	Description
ALB, 20	MINSK	-	vector of lower bounds
D	CCRN	0.0	see SCREEN
DELTA	FØPT	1.0	see DAVDØN output
FAC	FØPT	1.0	if greater than 0.0, the value given to diagonal elements of H matrix used in DAVDØN.
ILL11 *	-	1	test parameter for READIT
IMPLØT *	IØCCUR, 309	0	see READIT
IRED *	-	-	iteration counter
K *	MINSK	0	not used
KØUNT *	XXSAVE	-	iteration counter
MØDE *	IØCCUR, 303	3	test parameter for F123
UB, 20	MINSK	-	vector of upper bounds
P	BLKØ	0.0	see DAVDØN output
UP, 20	MIN	-	vector of upper bounds
WRF	ØWL	-	reduction factor
X, 40	BLKØ	-	see DAVDØN output
XX, 20	MINSK	-	see GMIMAX output

4. Numerical Procedures

As a first step, SUBROUTINE WHERE is called to allocate space needed for the input. Then subroutines SR2490 and ZPRM are called to preset the values of certain quantities, primarily input quantities. Next, the statements between CALL ZPRM and statement 2 preset additional quantities.

At statement 2, SUBROUTINE READIT is called to obtain the input for one case. The ILL111, used by READIT, is changed to two and MØDE is tested. Control is then transferred to 100, 110 or 120 when MØDE is 1, 2 or 3 respectively.

If statement 100 is reached, a single trajectory will be calculated, but no optimization will be attempted. First, the value of IREF, which is needed by F123, is saved as IREFS. IØP(74) is set equal to zero so that no calculations relating to wake length or wake cross section will be performed. Then F123 is called to secure the trajectory data, the original value of IREF is restored, then control goes to statement 2 to read the input for the next case.

At statement 110, F123 is again called, but this time the number of trajectories calculated will depend on the input. In addition, two types of integrals may be calculated and plots may be produced. No optimization of decoy parameters is attempted. After F123 is called, control returns to statement 2 to obtain input for the next case.

At statement 120, the integer code IREF is tested. If IREF equals 2, statement 5 is executed next. If IREF equals 1 or 3, F123 is called to obtain data for a reference reentry vehicle trajectory and control goes to statement 2 to transmit the input for the next case.

If statement 5 is reached, optimization of decoy design parameters will be attempted. IRED, an iteration counter, is set equal to -1. The next three statements save quantities which may be changed in the optimization process so that they can be restored later. Then, in the DØ loop ending at statement 10, the values of the first IN elements of the input array ØVECT are stored in the ØCCUR array, the X array, and the XX array for use by other subroutines. Then statement 20 is reached.

At statement 20, control is transferred to statement 500, 600, 700, 800 or 850 when IPRØC is respectively 1, 2, 3, 4 or 5.

If statement 500 is reached, the Fibonacci one variable procedure will be used. SUBROUTINE MIMAX is called to control the calculations and then ITERM is tested. If ITERM equals 1, statement 200 is executed next. Otherwise, NPRINT is saved as NPSAVE, then NPRINT is set equal to 1 and SUBROUTINE FEV is called. SUBROUTINE FEV calls SUBROUTINE F123 to get the trajectory corresponding to the solution and print extra output relating to this trajectory. Then the original value of NPRINT is restored and control is transferred to statement 900.

At 600, NPRINT is stored as NPSAVE, then reset to 1 and FEV is called to obtain trajectory data. NPRINT is then restored and control goes to statement 900.

At 700, SUBROUTINE DAVDØN is called to use the Davidon optimization method. Then ITERM is tested. If ITERM equals 1, control passes to statement 200. Otherwise, control passes to 900.

At 800, SUBROUTINE RØSBRK is called to use the Rosenbrock rotating coordinate method. The sequence of statements executed after the call to RØSBRK is the same as that following the call to MIMAX explained above.

At 850, the Fibonacci two variable method is used. The values of the lower bound vector ALB and the upper bound vector UB to be used by SUBROUTINE GIMAX are defined in the DØ loop ending at 860. KØUNT, an iteration counter used by SUBROUTINE GIMAX is set equal to zero. Then GIMAX is called to do the optimizing. Then, if ITERM equals 1, control goes to 200. Otherwise, in the DØ loop ending at 666, the elements of the XX array are tested to see if XX(I) equals X2SAVE(I) for any I. If an equality is found, XX(1) is set equal to X1SAVE(I) and control passes to 870. If no equality is found, the WRITE statement following 666 causes an error message to be written and statement 870 is reached.

At 870, NPRINT is saved as NPSAVE and reset to 1 to provide extra print out. FEV is called to get trajectory data corresponding to the two values in the solution. Then control passes to statement 900.

At statement 900, SUBROUTINE REDUCE is called to tighten constraints. Then the values of KRED and ITERM are printed. Next, KRED is tested. If KRED equals zero, go to statement 20 to iterate. Otherwise, statement 200 is reached, the quantities saved earlier are restored and control goes to statement 2 to get input for the next case.

5. Other Information

A. MAIN is not called by any other routine.

B. MAIN calls the following subroutines:

1. SUBROUTINE DAVDQN
2. SUBROUTINE F123
3. SUBROUTINE FEV
4. SUBROUTINE GIMAX
5. SUBROUTINE MIMAX
6. SUBROUTINE RQSBRK
7. SUBROUTINE SR2490
8. SUBROUTINE ZPRM
9. the library subroutine WHERE

1.3 Input Subroutines

The input for the Optimum Decoy Design Program is handled by two subroutines - READIT and ZREADX. SUBROUTINE READIT deals primarily with inputs needed for the trajectory calculation. Inputs for the optimization, wake, and effectiveness operations appear in ZREADX.

SUBROUTINE READIT (ILL111, IMPLØT)

1. Purpose

SUBROUTINE READIT reads in the input for one case and tests certain values in the IØP array.

2. Input

NOTE

(1) The source of all inputs to READIT except IØP is SUBROUTINE SETUP. IØP is from SUBROUTINE ZREADX.

(2) *indicates an integer quantity and an NØCCUR location unless otherwise designated.

(3) Numbers in the COMMON LOCATION column refer to locations in the ØCCUR array unless otherwise designated.

Name	Symbol	Common Location	Preset	Description
A, 514	A_i	301-814		curve fit coefficients
AE	A_e	214		thrust nozzle exit area, ft^2
ALPTAB, 75	α(table)	3646-3720		input angle of attack table, degrees
ALST	α_{ST}	122	0.2	stopping angle of attack, degrees
AWREF	$A_{W_{ref}}$	188		reference area of the WCDTAB drag coefficient, ft^2
B, 21	B_i	823-843		curve fit coefficients
BETA11	β_{11}	152		sublimation rate coefficient for initial h/s material, ft/sec / ^oR

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
BETA12	β_{12}	171		sublimation rate coefficient for h/s material after shape change, ft/sec/^{o}R
BETA21	β_{21}	153		same as for BETA11, $\frac{ft}{sec(^{o}R)^{\beta_3}}$
BETA22	β_{22}	172		same as for BETA12, $\frac{ft}{sec(^{o}R)^{\beta_3}}$
BETA31	β_{31}	154		order of reaction for initial configuration h/s material
BETA32	β_{32}	173		order of reaction for h/s material after shape change
BETA41	β_{41}	155		activation temperature for initial configuration, ^{o}R
BETA42	β_{42}	174		activation temperature for h/s material after shape change, ^{o}R
C	C	115	1.0	multiplier on stagnation point heating, used to simulate nose cap of a different material than heatshield
CAPG	G	19	32.21852	gravitational acceleration, ft/sec^2
CASE		128		case number
CDØWN, 16		3549-3564	1.0D-5	lower limit on accuracy of integrated variables, see VIXEN writeup
CDTAB, 75	C_D(table)	3383-3457		tabular input total drag coefficient

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
CHIGH, 16		3533-3549	1.0D-4	upper bound on accuracy of integrated variable, see VIXEN
CMQIN1	C_{m_q}	124		input C_{m_q} for initial configuration
CMQIN2	C_{m_q}	125		input C_{m_q} for configuration after shape change
CP21	$C_{p_{21}}$	160		specific heat of solid for initial configuration h/s material, $\frac{Btu}{lbm\,^{\circ}R}$
CP22	$C_{p_{22}}$	179		specific heat of solid for h/s material after shape change, $\frac{Btu}{lbm\,^{\circ}R}$
CPG1	$C_{p_{g_1}}$	161		specific heat of gas for initial configuration h/s material, $\frac{Btu}{lbm\,^{\circ}R}$
CPG2	$C_{p_{g_2}}$	180		specific heat of gas for configu- after shape change, $\frac{Btu}{lbm\,^{\circ}R}$
DATE		127		date
DELHC1	ΔH_{C_1}	166		heat of decomposition for initial h/s material, Btu/lbm
DELCH2	ΔH_{C_2}	185		heat of decomposition for h/s material after shape change Btu/lbm

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
DELIN		187	-2000.0	maximum allowable delta of integration, feet
DELRH1	$\Delta \rho_1$	159		difference between the virgin and char density of initial h/s material, lbm/ft^3
DELRH2	$\Delta \rho_2$	178		difference between the virgin and char density of h/s material for configuration after shape change, lbm/ft^3
DELY	ΔY	219		linear component of thrust offset, inches
DELZ	ΔZ	220		linear component of thrust off set, inches
DNBNDZ		248		lower altitude boundary on use of tabular input atmosphere, feet
EMO		129		memo number
EPSIL1	ϵ_1	167		coefficient of emission for initial h/s material
EPSIL2	ϵ_2	186		coefficient of emission for h/s material after shape change
F1	f_1	157		heat of ablation for the initial h/s material, Btu/lbm
F2	f_2	176		heat of ablation for the h/s material after shape change, Btu/lbm
G	g	27	32.174	conversion factor for changing slugs to lbm, lbm/slug

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
GAMF0	γ_{f_o}	105		initial flight path angle, degrees (negative number)
GAMMA	γ	28	1.4	ratio of specific heats
HREF1	H_{ref_1}	156		constant which equals zero for no combustion ablation
HREF2	H_{ref_2}	175		constant which equals zero for no combustion ablation
HTAB, 75	Z (table)	3233-3307		tabular altitude used with CDTAB and ALPTAB, feet
IATMØS		08 *		input atmosphere option code
IKCMQ		09 *		input C_{m_q} option code
INALPH		30 *		input angle of attack option
IØP, 90		IØCCUR (1-90)	1.0	control codes which call for various plots, influence coefficients, corridor printouts
ISP	I_{SP}	222	1.0	specific impulse
ITAPE		29 *		option for V, β, Z output tape
ITHRST		23 *		number of values in the thrust table
LA1	La_1	138		axial length of the initial vehicle configuration, inches
LA2	La_2	144		axial length of the vehicle immediately after shape change at ZTURN, in.

2. Input (Cont'd)

Name	Symbol	Common Location		Preset	Description
LAMDA1	λ_1	137			bluntness ratio of initial vehicle configuration
LAMDA2	λ_2	143			bluntness ratio of vehicle immediately after shape change at ZTURN
LØPT		07	*	1	trajectory option code; see other information
MATLN1		20	*	1	heatshield material option code for initial configuration
MATLN2		21	*	1	heatshield material option code for vehicle after shape change
MAXCD		18	*		the number of values in the CDTAB table
MAXVAL		06	*		the number of values in the input trajectory or wind tunnel conditions table
MAXWCD		19	*		the number of values in the WCDTAB table
MHEAT		10	*	0	option code which controls mass loss calculation
MØPT		03	*	0	option code which controls calculation of aerodynamic heating
MW	M_w	117		28.9	molecular weight of air, 28.9 gram/mole
MXTAB1		16	*	1	number of values in $X_{c.g}/D$, I, and I_x tables for first configuration

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
MXTAB2		17 *	1	number of values in $X_{cg.}/D$, I, and I_x tables for configuration after shape change
NGEØM		15 *	1	geometry input option code, indicates which geometric parameters are being input
NGL1	η_{GL_1}	164		laminar transpiration factor of gas for initial h/s material
NGL2	η_{GL_2}	183		laminar transpiration factor of gas for h/s material after shape change
NGT1	η_{GT_1}	165		turbulent transpiration factor of gas for initial h/s material
NGT2	η_{GT_2}	184		turbulent transpiration factor of gas for h/s material after shape change
NØSEØP		05 *		noseblunting option code
NPLØT, 5		24-28 *		plotting option codes for trajectory parameters
NPRINT		14 *	1	trajectory printout option code
NSL1	η_{SL_1}	162		laminar transpiration factor of solid for initial h/s material
NSL2	η_{SL_2}	181		laminar transpiration factor of solid for h/s material after shape change

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
NST1	η_{ST_1}	163		turbulent transpiration factor of solid for initial h/s material
NST2	η_{ST_2}	182		turbulent transpiration factor of solid for h/s material after shape change
NTHRUST		22 *	0	thrusting option code; see other information
ØCCUR, 4000		1-4000		array containing all variables relating to trajectory calculation
P0	P_o	109		initial angular rate, rad/sec
PHI0	ϕ_o	112		initial value of Euler angle, Φ degrees
PSI0	ψ_o	114		initial value of Euler angle, Ψ degrees
PSIZET	ψ_s	223		thrust offset angle, degrees
Q0	Q_o	110		initial angular rate, rad/sec
R	R	057	53.5	gas constant for air, $\frac{\text{ft-lb}}{\text{lbm-}^\circ\text{R}}$
RB1	Rb_1	136		base radius for initial vehicle configuration, in.
RB2	Rb_2	142		base radius of vehicle immediately after shape change at ZTURN, in.
RE	R_e	063	2.090299D+7	earth's radius, feet
RHØ21	ρ_{21}	158		char density for the initial h/s material, lbm/ft^3

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
RHØ22	ρ_{22}	177		char density for the h/s material after shape change, lbm/ft^3
RN1	Rn_1	135		nose radius of initial vehicle configuration, in.
RN2	Rn_2	141		nose radius of vehicle configuration after shape change, in.
SIG	σ	116	3.5	collision cross section for air, angstroms
SMR0	R_o	111		angular rate R, rad/sec
T0	T_o	102		initial time, sec.
TABI1, 50	I_1(table)	2933-2982	1.0	moment of inertia I = I_{yy} or I_{zz} for initial configuration, slug-ft^2
TABI2, 50	I_2(table)	2983-3032	1.0	moment of inertia, I, for vehicle configuration after shape change, slug-ft^2
TABIX1, 50	I_{x_1} (table)	3033-3082	1.0	the transverse moment of inertia I_{xx} for initial configuration, slug-ft^2
TABIX2, 50	I_{x_2} (table)	3083-3132	1.0	the transverse moment of inertia I_{xx} for configuration after shape change, slug-ft^2
TABRHØ, 50	ρ_∞(table)	3771-3820		tabular input free stream density, lbm/ft^3
TABSND, 50	a_∞(table)	3821-3870		tabular input free stream speed of sound, ft/sec

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
TABZ1, 50	Z_1(table)	3133-3182		tabular altitude for use with input $X_{cg.}/D$, I, and I_x tables of first configuration, feet
TABZ2, 50	Z_2(table)	3183-3232		tabular altitude for use with input $X_{cg.}/D$, I, and I_x tables for the configuration after shape change, feet
TBATMZ, 50	Z(table)	3721-3770		tabular input altitude for use with ρ_∞ and a_∞ tables, feet
TCRIT	t_{crit}	077		angle of attack cycle time test parameter, sec.
TECØN	t_{econ}	078		angle of attack cycle time test parameter, sec.
TH0	Th_o	207		reference thrust level, lb.
THDELT, 25	$t-t_{on}$	3618-3642		the time from thrust onset, an abscissa of THTH0 table
THDELZ, 25	$z - z_{on}$	3593-3617		change in altitude from thrust onset, an abscissa of THTH0 table
THEAL0	θ_{a_o}	113		initial Euler angle Θ_a, degrees
THETA1	θ_1	134		cone half angle of initial vehicle configuration, degrees
THETA2	θ_2	140		cone half angle of vehicle configuration after shape change, degrees
THEZET	θ_ζ	224		thrust misalignment angle, degrees

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
THTH0, 25	Th/Th_o	3568 -3592		non-dimensional thrust input table
TINIT	T_{init}	132	500.	initial internal body temperature, ^oR
TØFF	t_{off}	209		time of thrusting shut off, sec.
TØN	t_{on}	208		time of thrusting onset, sec.
TRAJRN, 75	Rn(table)	1644-1718		nose radius table for input trajectory option, must be input in addition to RN1, in.
TRAJT, 75	t(table)	1344-1418		time table for the input trajectory option, sec.
TRAJV, 75	V(table)	1494-1568		velocity table for the input trajectory option, ft/sec.
TRAJW, 75	W(table)	1569-1643		vehicle total weight table for input trajectory option, lb.
TRAJZ, 75	Z(table)	1419-1493		altitude table for input trajectory option, feet
TRJALP, 75	α(table)	1719-1793		angle of attack table for input trajectory option, degrees
TRZTR	Z_{tr}	243		input transition altitude, feet
TST	t_{st}	123	100.	trajectory stopping time, seconds
TW1	T_{W_1}	149	1200.	initial wall temperature for first vehicle configuration, ^oR

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
TW2	T_{W_2}	168	1200.	initial wall temperature for vehicle configuration after shape change
TWST	$T_{W_{ST}}$	148	580.	effective wall temperature used in free molecule drag calculation
TXCGD1,50	$(X_{cg}/D)_1$ (table)	2833-2882		$X_{cg.}/D$ table for initial vehicle configuration
TXCGD2,50	(X_{cg}/D_2) (table)	2883-2932		$X_{cg.}/D$ table for configuration after shape change
UPBNDZ		247		upper altitude boundary on use of tabular input atmosphere, feet
V0	V_o	106		initial velocity, ft/sec
W1	W_1	133		initial weight of the vehicle first configuration, lb.
W2	W_2	139		initial weight of the vehicle configuration after shape change, lb.
WCDTAB,75	C_D(table)	3458-3532		input total drag coefficient table
WHTAB,75	Z(table)	3308-3382		input altitude table abscissa of WCDTAB table
WTMINF,75	M (table)	1119-1193		input Mach number table for wind tunnel conditions options

2. Input (Cont'd)

Name	Symbol	Common Location	Preset	Description
WPTØT, 75	P_{total}(table)	1269-1343		input total pressure table for wind tunnel conditions option, lb/ft^2
WTRINF, 75	R_e /ft(table)	1194-1268		input Reynolds number per inch table for wind tunnel conditions option
WTZ, 75	Z(table)	1044-1118		input altitude table abscissa of WTMINF, WPTØT, WTRINF and ALPTAB used in wind tunnel conditions option
X1LØW	$\chi_{1_{LOW}}$	240	0.2	value of the rarefaction parameter which is the lower boundary of fairing region between free molecule and strong interaction flow regimes
X1UP	$\chi_{1_{UP}}$	239	0.4	value of the rarefaction parameter which is the upper boundary of fairing region between free molecule and strong interaction flow regimes
XLØW	χ_{LOW}	238	4.0	the value of the interaction parameter which is the lower boundary of fairing region between strong interaction and continuum flow regimes
XR0	X_{r_o}	107	0.0	initial range, ft.

2. Input (Concl'd)

Name	Symbol	Common Location	Preset	Description
XUP	χ_{UP}	237	6.0	the value of the interaction parameter which is the upper boundary of fairing region between strong interaction and continuum flow regimes
Z0	Z_o	108		initial altitude, feet
ZBAR	$\bar{Z}$	120	-10000.	altitude at which printout altitude increment changes, feet
ZETA	ζ	093	0.9	accomodation coefficient
ZØFF	Z_{off}	206		altitude for thrust shut off, feet
ZØN	Z_{on}	205		altitude for thrust onset, feet
ZPR1		118	10000.	initial altitude printout increment, feet
ZPR2		119		second altitude printout increment, feet
ZST	Z_{ST}	121	0.0	trajectory stopping altitude, feet
ZTURN	Z_{turn}	145	-1.0	altitude at which vehicle configuration changes discontinuously, feet

NOTE: IMPLØT and ILL111 enter from MAIN through the subroutine argument list and are control codes.

3. Output

The output from READIT is identical to the input with the single exception that the input NTHRUST is given the output designation NTHRST.

4. Numerical Procedures

READIT tests ILL111 when it is entered. If ILL111 equals one, statement 1 is executed next and if ILL111 equals two, statement 2 is executed next. The transfer to 1 will occur only the first time READIT is called.

The call to ZREADX and the calls to SETUP which follow statement 1 provide the reading subroutine READIN with information about the length and dimension of all input quantities.

At statement 2, SUBROUTINE READIN is called to read in the data for one case. E is found in column 1 of the first card read, statement 99 is executed next. If not, the statements between the sequence of statements starting after 2 and ending at 88 determines what the values of IØP(73) and IØP(74) should be. Then subroutine HEDING is called and the control goes to 98.

At 99, if IMPLØT equals one, AVPLT is called to close the plot file. The next two statements put an end of file mark on and rewind tape 8 if ITAPE is not equal to zero. Then EXIT is called to terminate the computer run.

At 98, control is returned to the main program.

5. Other Information

A. SUBROUTINE READIT is called by the main program only.

B. SUBROUTINE READIT calls

1. SUBROUTINE AVPLT
2. SUBROUTINE ZREADX
3. SUBROUTINE HEDING from the AVCO library
4. SUBROUTINE READIN from the AVCO library
5. SUBROUTINE SETUP from the AVCO library
6. the IBM system subroutine EXIT

C. The values of the control codes LØPT and NTHRST (NTHRUST input) have the following significance:

LØPT = 0 3 degree of freedom in rotation trajectory
= 1 particle trajectory
= 2 simplified angle of attack trajectory
= 3 input trajectory for purpose of calculating drag coefficient for specified conditions
= 4 input wind tunnel conditions table to calculate drag coefficient

NTHRST = 0 no thrust
= 1 non-dimensional thrust, THTH0, vs. change in altitude from ZØN
= 2 non-dimensional thrust, THTH0, vs. change in time from TØN

D. The matrix of the IØP option codes is given in the following table:

TABLE MATRIX OF OPTION CODES, IØP

	DIFFERENCE CORRIDOR PLOTS VERSUS ALTITUDE	CALCULATION OF EFFECTIVENESS INTEGRALS	CALCULATION OF CORRIDOR FUNCTIONS	SLOPE PRINTOUT	DIFFERENCE TABLES PRINTOUT	INFLUENCE COEFFICIENT CALCULATIONS AND PRINTOUT	INFLUENCE COEFFICIENT PLOTS	QUANTITY PLOTS VERSUS ALTITUDE
VELOCITY	IØP (1)	4	7	10	13	16	19	SEE NPLOT
DECELERATION	2	5	8	11	14	17	20	SEE NPLOT
BALLISTIC COEFF.	3	6	9	12	15	18	21	SEE NPLOT
WAKE LENGTH FOR 1ST FREQ.	22	28	34	40	46	52	58	77
WAKE LENGTH FOR 2nd FREQ.	23	29	35	41	47	53	59	78
WAKE LENGTH FOR 3rd FREQ.	24	30	36	42	48	54	60	79
WAKE RCS FOR 1ST FREQ.	25	31	37	43	49	55	61	80
WAKE RCS FOR 2nd FREQ.	26	32	38	44	50	56	62	81
WAKE RCS FOR 3rd FREQ.	27	33	39	45	51	57	63	82

IØP(76) SHOULD BE 1 AT AVCO AND 0 AT AEROSPACE
IØP(64-75; 83-90) ARE NOT INPUTS

SUBROUTINE ZREADX

1. Purpose

SUBROUTINE ZREADX provides information about the size and dimension of each input variable it refers to. This information is required by the input subroutine READIN.

2. Input

*indicates integer quantities

NOTE: All input variables are from SUBROUTINE SETUP

Name	Common Location	Description	Preset Value
AA, 27	PCCUR(11571-11597)	coefficients used by SUBROUTINE ADD under the designation E; preset in SR2490	AA(I) = 1.0 if I is 3, 6, 9, ... 27; otherwise AA(I) = 0.0
ACØE, 140	PCCUR(5881-6020)	coefficients used in SUBROUTINE MISC to define the free space radar cross section of the decoy	0.0
ACØN	NIMPUT	exponent for scale factor, $(CC\text{Ø}N)^{AC\text{Ø}N}$ on transition electron density n_{et}	1.0
AKW	NIMPUT	heatshield conductivity for wake calculations in Btu/(ft-^{o}R-hr)	

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
ALØW, 20	MIN	lower limits for independent variables	0.0
AMULT, 20	MIN	multipliers for each term in the penalty equations in FEV	1.0
BCB, 40	PCCUR(6181-6220)	lower corridor limits for ballistic coefficients	0.0
BCØN	NIMPUT	exponent for scale factor, (CCØN), $^{BC\emptyset N}$ on decoy rate B1	1.0
BCD, 40	PCCUR(6101-6140)	lower corridor limits for deceleration	0.0
BCV, 40	PCCUR(6021-6060)	lower corridor limits for velocity	0.0
BCWL1, 40	PCCUR(6501-6541)	lower corridor limits for wake length at first radar frequency	0.0
BCWL2, 40	PCCUR(6581-6620)	lower corridor limits for wake length at the second radar frequency	0.0
BCWL3, 40	PCCUR(6661-6700)	lower corridor limits for wake length at the second radar frequency	0.0
BCWR1, 40	PCCUR(6261-6300)	lower corridor limits for wake cross section at first radar frequency	0.0
BCWR2, 40	PCCUR(6341-6380)	lower corridor limits for wake cross section at second radar frequency	0.0

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
BCWR3, 40	PCCUR(6421-6460)	lower corridor limits for wake cross section at third radar frequency	0.0
BETAPL, 160	PCCUR(641-800)	ballistic coefficients input for reference reentry vehcile	0.0
BETAZ, 10	CWAKE	atmospheric density scale height for wake calculations--units of 1000 feet.	
BTWEN	DRCSEC	scaling constant in FLØWF	-
BZERØ	DRCSEC	scaling constant in RCSEC	-
B2	DRCSEC	scaling constant in RCSEC	-
B3	DRCSEC	scaling constant in RCSEC	-
B21	NIMPUT	scaling constant in FLØWF	-
B22	NIMPUT	scaling constant in FLØWF	-
B23	NIMPUT	scaling constant in FLØWF	-
B24	DRCSEC	scaling constant in RCSEC	-
CALØW, 20	MIN	lower bounds for constrained items in penalty equation	0.0
CCØN	ØCCUR(3963)	base of scale factor for decoy rate and transition electron density	1.0
CNE	DRCSEC	transition electron density when non-linear production terms are considered in turbulent wake, used in RCSEC	-

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
CNUMB, 169	NIMPUT	preset constant used in FLØWF and RCSEC, set in preset deck	-
CRHØW	NIMPUT	heatshield specific heat used in the wake calculations of FLØWF, Btu/(lb-^{o}R)	-
CTP, 20	MIN	upper bounds for constrained items in the penalty equation	0.0
DELTA	FØPT	estimate of the determinant of the H matrix of DAVDØN	-
DELWH	NIMPUT	heatshield thickness in inches used in wake calculations of FLØWF	-
DELX, 20	DØPT	in DAVDON, finite difference increments; in RØSBRK, the initial step sizes.	-
DHCHEM	NIMPUT	chemical enthalpy of heatshield in FLØWF (ft^2/sec^2)	-
DSB	DRCSEC	additional radar cross section due to consideration of non-linear production terms in turbulent wake in RCSEC	-
DTABL, 220	TBLS12	electron density as a function of normalized enthalpy, ratio of ablation to boundary layer air, and air density for 1000 PPM sodium seed	-

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
DVH, 50	PCCUR(11607-11656)	input values of design variables for second perturbation of comparison decoys	0.0
DVL, 50	PCCUR(11657-11706)	input values of design variables for first perturbation of comparison decoys	0.0
DX	DRCSEC	numerical step size used in finding wake length	-
EMCTBL, 12	TBLS12	cone Mach number array, one of the coordinates of the table of Mach number, M, as a function of cone half angle THETAC and cone Mach number MC in FLØWF	-
ENTABL, 225	TBLS12	table of n_c, electron density in e/cc, as a function of h/RT_o, normalized enthalpy, and ρ/ρ_o, air density	-
ERNRTB, 10	TBLS12	array of air densities, ρ, in lbm/ft^3 in table of equilibrium normal shock electron density table as a function of ρ and velocity	-
ERNTBL, 80	TBLS12	equilibrium normal shock electron density table in e/cc. whose coordinates are density, ERNRTB, and velocity, ERNUTB	-
ERNUTB, 8	TBLS12	array of velocities in 1000 ft/sec which are coordinate of ERNTBL	-
ERR	FØPT	if IPRØC = 3 (Davidon's method), stopping tolerance on transformed gradient; if IPRØC = 1 or 5 and LIMIT = 0, the accuracy requirement for the Fibonacci search in the physical units of the independent variables	0.01

2. Input (Cont'd)

Name	Common Location	Description	Preset
ETABL, 132	TBLS12	table of Mach number, M, as a function of cone Mach number EMCTBL and cone half angle THTBL.	
FAC	FØPT	if non-zero, the value which is given to the diagonal elements of the H matrix in DAVDON while non-diagonal elements are zeroed.	1.0
FGSM	NALTFG	multiplier on the step size limit, FGSM*(f/g_s), in the SUBRØUTINE READY in the Davidon method	4.0
FRQ1	CWAKE	first radar frequency	cycles/sec
FRQ2	CWAKE	second radar frequency	cycles/sec.
FRQ3	CWAKE	third radar frequency	cycles/sec.
H, 40	PCCUR(5841-5880)	altitudes for the corridor tables and for the radar measurement errors	0.0
HH, 1600	BLKØ	upper right triangular input of the initial elements of H matrix in DAVDØN	
HSTABL, 25	TBLS12	array of h/RT_o which is a coordinate of the electron density table, ENTABL	
ICØM, 200	IXCØM*	multi-purpose input array, see user manual	
IDBL	IØCCUR(314)*	interger code which determines units of radar cross section; if = 3, RCS in decibels, if = 4, RCS in square meters	4

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
IDC, 50	IDNØS*	identification array, the values are the indices of the constraints in the ØCCUR array	0
IDNØ, 50	IDNØS*	identification array, the values are the indices of the design parameters in the ØCCUR array	0
IEX	IØPT*	the exponent of the penalty function	2
IGDH, 20	IGDHL*	locations in the ØCCUR array of the first terms used in obtaining the general differences in MISC	0
IGDL, 20	IGDHL*	locations in the ØCCUR array of the second terms used in obtaining the general differences in MISC	0
IN	IØPT*	the number of design variables	1
IND	CWAKE*	printout option control in FLØWF	
IND2	DRCSEC*	if IND 2 = 0, no output is generated by RCSEC. If IND2 = 1, intermediate steps are printed out	
IØP, 90	IØCCUR(1-90)*	input integer code, see user manual	
IPNT	IØPT*	not used currently	
IPRØC	IØPT*	optimizer selection code, see user manual	1

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
IRAND	IØPT*	number of random starting points to be used in Davidon method	0
IREF	IØCCUR(301)	trajectory processing option code; value of 1, calculate R/V trajectory or other miscellaneous calculations; value of 2, calculate and compare decoy trajectory; value of 3, input an R/V trajectory	1
ISEN1	SENSE *	DAVDØN printout control	0
ISEN2	SENSE *	not used currently	0
IWAKE	CWAKE *	number of entries in the wake-altitude table, WKALT	
IWPRNT	CWAKE *	print option in WAKE	
K	MINSK *	not used	
LIMIT	IØPT *	in the one variable Fibonacci, number of times the function will be calculated unless = 0, then the value of ERR determines the number of times. In Davidon method, maximum number of iterations. In Rosenbrock's method, limit on the number of successful steps taken on each variable. In the two variable Fibonacci method, $(LIMIT)^2 + 1$ is the number of times the function will be calculated.	30

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
LPLØT	IØCCUR(302)*	number of trajectory points in the input R/V trajectory for MØDE = 3	1
LRED	ØWL *	the maximum number of times that the factor WRF can be applied, see REDUCE	0
MØDE	IØCCUR(303)*	fundamental option code; if = 1 single trajectory calculation, input trajectory, or wind tunnel conditions used to find drag coefficients; if = 2, R/V decoy comparison and influence coefficient calculations are performed; if = 3, optimization calculations are performed.	3
NALT	NALTFG *	alternate logic for step size in SUBROUTINE READY	0
NCØMDV, 50	IØCCUR(91-140)*	identification code numbers (indices in ØCCUR array) of design variables to be perturbed under MØDE = 2 influence coefficient calculations	133
NCØNS	IØPT*	number of entries in constraint table	1

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
NCP	IØCCUR(304)*	number of corridor points	1
NDECØY	IØCCUR(305)*	vehicle type identification code; if = 1, R/V or one basic decoy; if = 2, one perturbation of each design variable; if = 3, two perturbations on each design variable	1
NDVCH	IØCCUR(306)*	number of entries in NCØMDV table, i.e., number of design variables	1
NPA	IØCCUR(307)*	number of entries in NPV table for influence coefficient plots	1
NPV, 160	IØCCUR(141-300)*	index of altitudes for influence coefficient plots	1
NSTWL	DRCSEC*	maximum number of steps used to compute wake length	
OVECT, 20	ØWL	initial values of the design variables	5.0

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
PFD	PCCUR(11757)	probability of false dismissal of an R/V, see EFFECT	0.0
PHI1, 10	CWAKE	look angle for radar of the first frequency	
PHI2, 10	CWAKE	look angle for radar of the second frequency	
PHI3, 10	CWAKE	look angle for radar of the third frequency	
PRAND	FØPT	random step size control for DAVDØN	0.0
RHØSL	NIMPUT	sea level density in lbm/ft^3 in FLØWF	
RHØW	NIMPUT	heatshield density in lbm/ft^3 in FLØWF	
RSTABL, 9	TBLS12	array of ρ/ρ_o which is a coordinate of the electron density table ENTABL	
RT0	NIMPUT	the reference enthalpy in FLØWF in ft^2/sec^2	
SB, 40	PCCUR(6821-6860)	the standard deviation of radar measurement errors for ballistic coefficient	0.0
SD, 40	PCCUR(6781-6820)	standard deviation of radar measurement errors for deceleration	0.0
SIGNL1	CWAKE	noise level for wake length definition at first frequency	

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
SIGNL2	CWAKE	noise level for wake length definition at second frequency	
SIGNL3	CWAKE	noise level for wake length definition at third frequency	
SMULT, 25	MULT	multipliers of special penalty terms in SUBROUTINE SCREEN	1.0
SRS, 9	PCCUR(11598-11606)	number of smooth radar samples	0.0
SV, 40	PCCUR(6741-6780)	standard deviation of radar measurement errors for velocity	0.0
SWL1, 40	PCCUR(6981-7020)	standard deviation of radar measurement errors for wake length at first radar frequency	0.0
SWL2, 40	PCCUR(7021-7060)	standard deviation of radar measurement errors for wake length at second radar frequency	0.0
SWL3, 40	PCCUR(7061-7100)	standard deviation of radar measurement errors for wake length at third radar frequency	0.0
SWR1, 40	PCCUR(6861-6900)	standard deviation of radar measurement errors for wake cross section at first radar frequency	0.0
SWR2, 40	PCCUR(6901-6940)	standard deviation of radar measurement errors for wake cross section at second radar frequency	0.0
SWR3, 40	PCCUR(6941-6980)	standard deviation of radar measurement errors for wake cross section at third radar frequency	0.0

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
TABL	NIMPUT	ablation temperature in ^{o}K of the heatshield used in FLØWF	
TAU1	CWAKE	pulse length for radar of first frequency in μ sec.	
TAU2	CWAKE	pulse length for radar of second frequency in μ sec.	
TAU3	CWAKE	pulse length for radar of third frequency in μ sec.	
TCB, 40	PCCUR(6221-6260)	upper corridor array for ballistic coefficient	0.0
TCD, 40	PCCUR(6141-6180)	upper corridor array for deceleration	0.0
TCV, 40	PCCUR(6061-6100)	upper corridor array for velocity	0.0
TCWL1, 40	PCCUR(6541-6580)	upper corridor array for wake length at first radar frequency	0.0
TCWL2, 40	PCCUR(6621-6660)	upper corridor array for wake length at second radar frequency	0.0
TCWL3, 40	PCCUR(6701-6740)	upper corridor array for wake length at third radar frequency	0.0
TCWR1, 40	PCCUR(6301-6340)	upper corridor array for wake cross section at first radar frequency	0.0
TCWR2, 40	PCCUR(6381-6420)	upper corridor array for wake cross section at second radar frequency	0.0

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
TCWR3, 40	PCCUR(6461-6500)	upper corridor array for wake cross section at third radar frequency	0.0
THTTBL, 11	TBLS12	array of cone half angle which is a coordinate of the M vs. THETAC and MC table, ETABL	
TPLØT, 160	PCCUR(1-160)	tab[illegible] of times for the input re[illegible]y vehicle trajectory	0.0
UP, 20	MIN	vector of upper bounds on design variables	0.0
V0GPLT, 160	PCCUR(481-640)	table of deceleration ($\dot{V}/g$) for input R/V trajectory in g's	0.0
VPLØT, 160	PCCUR(321-480)	table of velocities for input R/V trajectory in ft/sec	0.0
WKALT, 10	CWAKE	altitudes correspond to the scale height and look angle input tables used in SUBROUTINE WAKE	
WL1P, 160	PCCUR(1281-1440)	input wake length at the first radar frequency for the input R/V trajectory	0.0
WL2P, 160	PCCUR(1441-1600)	input wake length at the second radar frequency for the input R/V trajectory	0.0
WL3P, 160	PCCUR(1601-1760)	input wake length at the third radar frequency for the input R/V trajectory	0.0

2. Input (Cont'd)

Name	Common Location	Description	Preset Value
WRF	ØWL	generalized reduction factor in REDUCE	
WR1P, 160	PCCUR(801-960)	input wake cross section at the first radar frequency for the input R/V trajectory	0.0
WR2P, 160	PCCUR(961-1120)	input wake cross section at the second radar frequency for the input R/V trajectory	0.0
WR3P, 160	PCCUR(1121-1280)	input wake cross section at the third radar frequency for the input R/V trajectory	0.0
WSTALT	CWAKE	starting altitude for wake calculation	-
XCØM, 200	IXCØM	multi-purpose input array, see user manual	-
XDTABL, 11	TBLS12	array of air density ratios, ρ/ρ_0, which is one of three coordinates of the electron density table D	-
X2BØD	DRCSEC	two body overdense length in RCSEC	-
X3B	DRCSEC	station where linear production terms first dominate the non-linear production terms in RCSEC	-
YDTABL, 11	TBLS12	the array of h/RT_o values which is one of three coordinates of the electron density table D.	

2. Input (Concl'd)

Name	Common Location	Description	Preset Value
ZDTABL, 11	TBLS12	the array of ratios of ablation to boundary layer air, M_{rat}, one of three coordinates of the electron density table D	-
ZNUS	TBLS12	sea level collision frequency in cps used in FLØWF	
ZPLØT, 160	PCCUR(161-320)	altitudes for the input R/V trajectory	0.0

3. Output

*indicates integer quantity

Name	Common Location	Description
D, 220	TBLS12	see input for DTABL
ID1, 50	IDNØS*	see input for IDNØ
ID2, 50	IDNØS*	see input for IDC
XYZTBL(11, 3)	TBLS12	single three dimensional table which incorporates the three independent variables of the D table; the second integer indicates the related independent variable in the following manner: 1 indicates XDTABL values 2 indicates YDTABL values 3 indicates ZDTABL values

In addition, all input quantities not mentioned above are output with no change in name.

4. Numerical Procedure

SUBROUTINE ZREADX calls SUBROUTINE SETUP, and entry point for SUBROUTINE READIN, once for each input variable it refers to. These calls provide READIN with the length and dimension of each of these input variables. All of the calls to SETUP could not be included in one subroutine because it would then have been too large for the compiler to handle. The remaining calls to SETUP are made in SUBROUTINE READIT.

5. Other Information

A. SUBROUTINE ZREADX is called by SUBROUTINE READIT only.

B. SUBROUTINE ZREADX calls SUBROUTINE SETUP only. SETUP is an entry point to READIN.

1.4 Presetting Operations

Preset values are defined by the three subroutines - SR2490, ZPRM, and ZPRS. The preset values for the curve fit coefficients A_i and B_i are assigned in SUBROUTINE ZPRS. SUBROUTINE SR2490 presets primarily quantities used in the trajectory computations, while the quantities of ZPRM are related for the most part to the optimization calculations.

SUBROUTINE SR2490

1. Purpose

SUBROUTINE SR2490 presets the values of many of the input variables.

2. Input

Name	Common Location	Source of Input	Description	Preset Value
A, 514	ØCCUR(301-814)	ZPRS	curve fit coefficients	see ZPRS
B, 21	ØCCUR(823-843)	ZPRS	cure fit coefficints	see ZPRS

3. Output

Name	Common Location	Description	Preset Value
A, 514	ØCCUR(301-814)	curve fit coefficients	see ZPRS
AA, 27	PCCUR(11571-11597)	coefficients used in FUNCTION ADD	1.0 for index which is a multiplier of 3, otherwise zero
ALST	ØCCUR(122)	stopping angle of attack	0.2 degrees
B, 21	ØCCUR(823-843)	curve fit coefficients	see ZPRS

3. Output (Cont'd)

Name	Common Location	Description	Preset Value
C	ØCCUR(115)	multiplier on stagnation heating to perturb mass loss	1.0
CAPG	ØCCUR(19)	gravitational acceleration, preset to 32.21852 ft/sec^2	
CDØWN, 16	ØCCUR(3549-3564)	array of lower bounds on integration accuracy	0.00001
CHIGH, 16	ØCCUR(3533-3548)	array of upper bounds on integration accuracy	0.0001
DELIN	ØCCUR(187)	maximum allowable delta of integration in ADM4RK	-2000.0 ft.
FACTR1	ØCCUR(189)	numerical factor, preset to (.002375 slug)$^{0.8}$	
G	ØCCUR(27)	conversion factor for changing slugs to lbm.	32.174 lbm/slug
GAMMA	ØCCUR(28)	ratio of specific heats for air	1.4
IMPLØT*	IØCCUR(309)	test parameter for opening plot file	0
IØP, 90*	IØCCUR(1-90)	option codes for decoy optimization	1
IREF*	IØCCUR(301)	integer control code which must be 1 or 2 for the reentry vehicle and 2 for decoy optimization calculations (program changes this to 2 after R/V case)	1
ISP	ØCCUR(222)	specific impulse	1.0 sec.

3. Output (Cont'd)

Name	Common Location	Description	Preset Value
LØPT*	NØCCUR(07)	trajectory option control code	1
LP*	ØCCUR(4000)	error control code	1
LPLØT*	IØCCUR(302)	the number of altitudes for which information is output from VIXEN	1
MATLN1*	NØCCUR(20)	material option control code for initial configuration	1
MATLN2*	NØCCUR(21)	material option control for configuration after shape change at ZTURN	1
MW	ØCCUR(117)	molecular weight of air	28.9 gram/ mole
MXTAB1*	NØCCUR(16)	number of values in $X_{c.g.}/D$, I, I_x table for initial configuration	1
MXTAB2*	NØCCUR(17)	number of values in $X_{c.g}/D$, I, I_x table for configuration after shape change	1
NCØMDV, 50*	IØCCUR(91-140)	identification codes of variables to be perturbed in the optimization	133
NCP*	IØCCUR(304)	number of elements in the array of input corridor limits	1
NDECØY	IØCCUR(305)	test parameter for influence coefficient plots	1

3. Output (Cont'd)

Name	Common Location	Description	Preset Value
NDVCH*	IØCCUR(306)	input control code for matching subroutine	1
NGEØM*	NØCCUR(15)	geometry input option code	1
NPA*	IØCCUR(307)	the number of altitudes which influence coefficients will be plotted	1
NPRINT*	NØCCUR(14)	print code option	1
NPV, 160*	IØCCUR(141-300)	array containing the indices of altitudes at which influence coefficients will be produced	1
PI	ØCCUR(42)	mathematical constant, preset to 3.141592653589793	
R	ØCCUR(57)	gas constant for air, preset to 53.5 $\frac{\text{ft-lb}}{\text{lbm-}^{\circ}\text{R}}$	
RE	ØCCUR(63)	radius of the earth	20902290. ft
SIG	ØCCUR(116)	collision cross section of air	3.5 A°
TABIX1, 50	ØCCUR(3033-3082)	transverse moment of inertia I_{xx} for initial configuration	1.0 slug-ft^2
TABIX2, 50	ØCCUR(3083-3132)	transverse moment of inertia I_{xx} for configuration after shape change at ZTURN	1.0slug-ft^2
TABI1, 50	ØCCUR(2933-2982)	moment of inertia $I = I_{yy} = I_{zz}$ for initial configuration	1.0 slug-ft^2

3. Output (Cont'd)

Name	Common Location	Description	Preset Value
TABI2, 50	ØCCUR(2983-3032)	moment of inertia $I = I_{yy} = I_{zz}$ for configuration after shape change at ZTURN	1.0 slug-ft^2
TECØN	ØCCUR(78)	limit on time for one cycle in angle of attack	2.0 sec.
TINIT	ØCCUR(132)	internal temperature of vehicle used in iterative mass loss calculation to find temperature gradient	500.0°R
TST	ØCCUR(123)	stopping time	100. sec.
TWST	ØCCUR(143)	effective wall temperature used in rarefied flow region	580.$^\circ$R
TW1	ØCCUR(149)	input initial wall temperature for the initial configuration	1200.$^\circ$R
TW2	ØCCUR(168)	input initial wall temperature for the configuration after shape change	1200.$^\circ$R
XLØW	ØCCUR(238)	the value of the interaction parameter $\bar{\chi}$ which marks the beginning of the fully laminar flow regime and the end of fairing between continuum and strong interaction	4.0
XUP	ØCCUR(237)	the value of the interaction parameter $\bar{\chi}$ which marks the end of the strong interaction and the beginning of fairing region between strong interaction and continuum flow	6.0

3. Output (Concl'd)

Name	Common Location	Description	Preset Value
X1LØW	ØCCUR(240)	the value of the rarefaction parameter $\bar{\chi}_1$ which marks the end of fairing region between the transitional free molecule and the strong interaction and the beginning of strong interaction regime	0.2
X1UP	ØCCUR(239)	the value of the rarefaction parameter $\bar{\chi}_2$ which marks the beginning of fairing region between transitional free molecule and strong interaction and the end of the free molecule transitional flow regime	0.4
ZBAR	ØCCUR(120)	altitude at which altitude print increment changes	-10000.0 ft.
ZETA	ØCCUR(93)	accommodation coefficient	0.9
ZFR1	ØCCUR(118)	initial altitude print increment	10000. ft.
ZTURN	ØCCUR(145)	altitude at which a discontinuous shape change occurs	-1.0 ft.

4. Numerical Procedures

As a first step SR2490 sets all of the elements in the four common arrays IØCCUR, NØCCUR, ØCCUR and PCCUR equal to zero. Then the values of those elements which should not be zero are reset to their proper values. At the end of SR2490, SUBROUTINE ZPRS is called to accomplish the resetting of the A and B arrays which are stored in the ØCCUR common block.

5. Other Information

A. SUBROUTINE SR2490 is called by MAIN only.

B. SUBROUTINE SR2490 calls SUBROUTINE ZPRS.

SUBROUTINE ZPRM

1. Purpose

SUBROUTINE ZPRM presets the values of some of the variables used in the program.

2. Input

None

3. Output

*indicates an integer quantity

Name	Common Block	Preset Value	Description
ALØW, 20	MIN	0.0	lower limits on independent variables
ALPHA	XCØM(1)	3.0	multiplier used to obtain new step size from previous step size for a successful step in RØSBRK
AMULT, 20	MIN	1.0	multipliers for each term in the penalty equations
BETA	XCØM(2)	.5	multipler whose negative is used to obtain a new step size from the previous step size after an unsuccessful step in RØSBRK.
CALØW, 20	MIN	0.0	lower bounds for constrained items in the penalty equation
CTP, 20	MIN	0.0	upper bounds for constrained items in the penalty equation

3. Output

Name	Common Block	Preset Value	Description
DEL	XCØM(4)	.01	multiplier used in obtaining the limiting value on function U in RØSBRK
DELX, 20	DØPT	.001	in DAVDØN, finite difference increments; in RØSBRK, the initial step sizes.
ERR	FØPT	0.01	See input description for ZREADX
FGSM	NALTFG	4.0	multiplier on step size limit, FGSM*(f/g_s), in SUBROUTINE READY
GAMMA	XCØM(3)	0.5	multiplier used to redefine step size after a failure
ICØM,*200	IXCØM	all 0	multi-purpose input array, see user manual
IEX*	IØPT	2	the exponent of the penalty function
IGDH*, 20	IGDHL	0	locations in the ØCCUR array of the first terms used in obtaining the general differences in MISC
IGDL*, 20	IGDHL	0	locations in the ØCCUR array of the second terms used in obtaining the general differences in MISC
IN*	IØPT	1	the number of design variables
IPRØC*	IØPT	1	optimizer selection code, see user manual
IRAND*	IØPT	0	number of random starting points to be used in the Davidon method
ISEN1*	SENSE	0	DAVDØN printout control

3. Output

Name	Common Block	Preset Value	Description
ISEN2*	SENSE	0	not used currently
LIMIT*	IØPT	30	see input description of ZREADX
NALT*	NALTFG	0	code for alternate logic for step size in SUBROUTINE READY
NCØNS*	IØPT	1	number of constraints
ØVECT, 20	ØWL	5.0	initial values of the design variables
PRAND	FØPT	0.0	random step size control for DAVDØN
RATU	XCØM(5)	0.5	input, tolerance on quantity URAT, described in text of RØSBRK
SMULT, 25	MULT	1.0	multiplier on penalty function used in SUBROUTINE SCREEN
TØL	XCØM(6)	0.0001	input accuracy test parameter
UP, 20	MIN	0.0	vector of upper bounds on design variables
WRF	ØWL	0.9	generalized reduction factor in SUBROUTINE REDUCE

4. Numerical Procedures

No special numerical methods are required. The setting of values is accomplished in the usual FORTRAN arithmetic type statements

5. Other Information

A. ZPRM is called by MAIN only.

B. ZPRM does not call or reference any subprogram.

SUBROUTINE ZPRS (A, B)

1. Purpose

SUBROUTINE ZPRS defines the elements of the A and B arrays which are coefficients of the curve fits contained in various other subroutines of the program.

2. Input

There are no input quantities to SUBRØUTINE ZPRS.

3. Output

B(1) through B(21)	Coefficients for the probability distribution between free molecule and continuum flow regimes. These are used in SUBROUTINE DRAGCØ.
A(11) through A(37)	Coefficients used in finding maxima and minima in SUBRØUTINE VIXEN.
A(41) through A(58)	Coefficients used for calculating blunt cone pressure distributions for cone half angles greater than or equal to 20 degrees in SUBRØUTINE AERØDY.
A(90) through A(104)	Coefficients used for calculating edge temperatures in SUBRØUTINE PRELIM.
A(105) through A(110)	Coefficients used for calculating specific heat at constant pressure when temperature is in the range 700 to 5000 degrees Rankine inclusive. They are used in SUBRØUTINE PRELIM.

A(111) and A(112)	Coefficients used for calculating specific heats at constant pressure when temperature is greater than 5000 degrees Rankine. They are used in SUBRØUTINE PRELIM.
A(135) through A(161)	Coefficients used for blunt cone pressure distributions for cone half angles less than 20 degrees. They are used in SUBRØUTINE AERØDY.
A(162) through A(173)	Coefficients used for calculating TSTAR in SUBRØUTINE DRAGCØ.
A(174) through A(193)	Coefficients used for calculating free molecule drag coefficients on the spherical nose. They are used in SUBRØUTINE DRAGCØ.
A(200) through A(207)	Coefficients used in calculating the strong interaction drag coefficients for cone half angles less than 15 degrees. They are used in SUBRØUTINE DRAGCØ.
A(211) through A(246)	Coefficients used to determine blunt cone forebody pressure drag coefficients for cone half angles 4 to 10 degrees inclusive. They are used in SUBRØUTINE DRAGCØ.
A(247) through A(282)	Coefficients used to determine the blunt cone forebody pressure drag coefficient for cone half angles greater than 10 and less than or equal to 20 degrees. They are used in SUBRØUTINE DRAGCØ.
A(283) through A(290)	Coefficients used to determine the shape factor for calculating laminar induced drag coefficients. They are used in SUBRØUTINE DRAGCØ.

A(300) through A(311) Coefficients used to calculate the transition altitude. They are used in SUBRØUTINE PRELIM.

A(312) through A(347) Coefficients used to determine the blunt cone forebody pressure drag coefficient for cone half angles greater than 20 and less than or equal to 40 degrees. They are used in SUBRØUTINE DRAGCØ.

A(348) through A(383) Coefficients used to determine the ratio (C_{d_p} with angle of attack effects)/(C_{d_p} for angles of attack of zero), where the absolute value of the angle of attack is greater than 4 and less than or equal to 40 degrees. If the absolute value of alpha is greater than 40 degrees then alpha is set equal to 40 degrees. They are used in SUBRØUTINE DRAGCØ.

A(384) through A(399) Coefficients used in calculating the strong interaction drag coefficient for cone half angles greater than or equal to 15 degrees. They are used in SUBRØUTINE DRAGCØ.

A(400) through A(420) Coefficients used to determine A(118), A(119) and A(120) in SUBRØUTINE DRAGCØ to calculate turbulent skin friction drag coefficient.

A(421) through A(456) Coefficients used to determine the ratio (C_{d_p} with angle of attack effects)/(C_{d_p} for angle of attack of zero), where the absolute value of alpha is less than 4 degrees and greater than zero. They are used in SUBRØUTINE DRAGCØ.

A(457) through A(471)	Coefficients used to determine the bluntness correction to the skin friction drag coefficient for a bluntness ratio greater than or equal to 0.2. They are used in SUBRØUTINE DRAGCØ.
A(472) through A(486)	Coefficients used for the same purpose as A(457) through A(471) when the bluntness ratio is less than 0.2. They are used in SUBRØUTINE DRAGCØ.
A(487) through A(498)	Coefficients used to calculate base drag coefficient in SUBRØUTINE DRAGCØ for free stream Mach numbers less than 7.
A(499) through A(510)	Coefficients used for the same purpose as A(487) through A(498) when free stream Mach number is greater than or equal to 7.

4. Numerical Procedure

All the elements of the A and B arrays are set equal to zero by SUBRØUTINE SR2490, which then calls SUBRØUTINE ZPRS only once for each computer run to reset those elements which should not be zero. Array A has 514 elements and B has 21 elements. Those elements being reset here are set equal to constant numerical quantities.

5. Other Information

A. SUBRØUTINE ZPRS calls in no other subroutines or functions.

B. SUBRØUTINE ZPRS is called by SUBRØUTINE SR2490.

2.0 OPTIMIZATION TECHNIQUES

Optimization Techniques

The following sections deal with the subroutines involved in the optimization process. In addition to the subroutines directly related to the four optimization techniques - the one and two variable Fibonacci searches, the Davidon variable metric method, and the Rosenbrock rotating coordinate method, those related to the penalty function transformation - FEV, SCREEN, REDUCE - and the subroutine which calculates the gradient of the penalty function - FCN are described.

2.1 Penalty Function Transformation

The subroutines described in this section are FEV which defines the penalty function equation, SCREEN which tests the design variables against limitations imposed, and REDUCE which controls the sequence of solution-finding operations in order to produce an optimum solution.

SUBROUTINE FEV(N, X, VAL)

1. Purpose

SUBROUTINE FEV evaluates the penalty function which is being optimized for the point described by the N values of the X array.

2. Input

*indicates integer quantity

Name	Common Block	Source of Input	Description
AMULT, 20	MIN	ZREADX or ZPRM	multipliers for each constraint penalty term
CALØW, 20	MIN	ZPRM or ZREADX	lower allowable bounds for each constraint
CUP, 20	MIN	ZPRM or ZREADX	upper allowable bounds for each constraint; CTP is the name in ZREADX
D	CCRN	MAIN	penalty function value
ICØM, 200	IXCØM *	ZREADX or ZPRM	input option
ID1, 50	IDNØS *	ZREADX	identification numbers for the independent (design) variables; read in as IDNØ

2. Input (Concl'd)

Name	Common Block	Source of Input	Description
ID2, 50	IDNØS *	ZREADX	identification numbers for the constraints; read in as IDC
IEX	IØPT *	ZPRM	exponent in the penalty equation; should be 1 or 2 for Fibonacci method; should be 2 for DAVDØN method.
ISUC	- *	SCREEN	integer control code; if ≥ 0, function is calculated; if < 0, dummy function is used
LP	ØCCUR(4000)*	F123	error code
N	- *	FMIMAX or FCN	the number of independent variables which make up the X array
NCØNS	IØPT*	ZREADX or ZPRM	the number of constraints in the ID2 (input IDC) table
VAL	-	SCREEN, F123, or CLASSC	the value of the penalty function
X, 20	-	FMIMAX or FCN	array containing the independent variables

3. Output

Name	Common Block	Description
D	CCRN	see input
ITERM	END *	integer control code: zero value indicates a non-zero, defined penalty function; non - zero value indicates a zero or undefined function
N	- *	see input
ØCCUR, 4000	-	common block
VAL	-	see input
X, 20	-	see input

4. Numerical Procedure

SUBROUTINE FEV begins calculations by zeroing the quantities ITERM and VAL. The DØ loop ending with statement 10 for values of I from 1 to N sets the IZth location in the ØCCUR array equal to the Ith X value, where IZ has the value ID1(I).

If the integer code ICØM(1) is equals 1, SUBROUTINE CLASSIC is called in to provide the optimizer with classic check case functions, then control passes to statement 22. If ICØM(1) is not equal to 1, control passes to statement 15 and the calling of SUBROUTINE SCREEN to test the design variables against the limits imposed. If the variables are outside the limits SCREEN defines a dummy penalty function and sets ISUC to a negative number. If the variables are within the limits, ISUC is set equal to zero in SCREEN.

ISUC is then tested in FEV. If less than zero, control passes to statement 100, where the penalty function, VAL, and the X array are printed out, before the return to the calling subroutine is executed. If ISUC is greater than or equal to zero, SUBROUTINE F123 is called to determine the reentry vehicle and decoy trajectory data and perform the appropriate matching calculations. If the error code LP is greater than or equal to 6, indicating a program failure in the trajectory calculations, ITERM is set equal to 1, indicating an undefined function, and control passes to statement 100. If LP is less than 6, control passes to statement 20, which calls SUBROUTINE MISC to perform miscellaneous calculations. SUBROUTINE EFFECT is then called to calculate the probability that a decoy will be discriminated.

Statement 22 tests the number of constraints, integer code NCØNS; if NCØNS equals zero, control passes to statement 100. If NCØNS is non-zero, the WRITE statement is executed which prints out the following titles: IZ, LOWER BOUND, UPPER BOUND, OCCUR(IZ), PENALTY. Next a DØ loop ending with the statement 50 is executed for values of I from 1 to NCØNS. At each pass through the DØ loop, the Ith value of PNLTY is zeroed and the index IZ set equal to the Ith value in the ID2 array. If ØCCUR(IZ) is less than or equal to the lower allowable bound CALØW(I), the following definition is made

$$PNLTY(I) = AMULT(I)\left(\left|OCCUR(IZ) - CALØW(I)\right|\right)^{IEX}$$

and the results IZ, CALØW(I), CUP(I), ØCCUR(IZ), PNLTY(I) printed out before control passes to 50.

If ØCCUR(IZ) is greater than CALØW(I), control passes to statement 25. If ØCCUR(IZ) is greater than CUP(I), PNLTY(I) is defined to be

$$PNLTY(I) = AMULT(I)\left(\left|ØCCUR(IZ) - CUP(I)\right|\right)^{IEX}$$

then statement 27 is executed.

If ØCCUR (IZ) is less than or equal to CUP(I), control passes to statement 27. Statement 27 causes the printout of the same quantities indicated in the previous paragraph, before control reaches statement 50. Statement 50 defines the value of the penalty function, VAL, to be the sum of the NCØNS values of PNLTY. D is then equated to VAL, before VAL is tested. If VAL is greater than zero, control passes to statement 100. If VAL is less than or equal to zero, ITERM is set equal to -1, indicating a zero function, before 100 is reached.

5. Other Information

A. SUBROUTINE FEV is called by the MAIN program and by the following subroutines:

1. SUBROUTINE FCN
2. SUBROUTINE RØSBRK
3. SUBROUTINE GIMAX
4. SUBROUTINE MIMAX
5. SUBROUTINE FMIMAX

B. SUBROUTINE FEV calls in

1. SUBROUTINE CLASSC
2. SUBROUTINE SCREEN
3. SUBROUTINE F123
4. SUBROUTINE MISC
5. SUBROUTINE EFFECT

C. SUBROUTINE FEV calls in the library function FDXPI.

SUBROUTINE SCREEN (N, X, ISUC, VAL, D)

1. Purpose

SUBROUTINE SCREEN tests the values of up to twenty-five elements of the ØCCUR array to determine whether these values lie within prescribed limits. If any values lie outside the prescribed limits, a quantity E is calculated as a measure of the error.

2. Input

All numbers in the COMMON BLOCK column refer to positions in the ØCCUR array unless otherwise indicated.

*indicates integer quantity and an NØCCUR location

Name	Source	Common Block	
D	MAIN or FEV	CCRN	penal
ISP	READIT	222	speci
LAMDA1	READIT	137	RN1/ config
LAMDA2	READIT	143	RN2/ config
LA1	READIT	138	axial
LA2	READIT	144	axial (after
NGEØM	READIT	15 *	vehic
NTHRST	READIT	22 *	corre 2 indi and th
ØCCUR, 4000	-	-	comm
PI	SR2490	42	π
RB1	READIT	136	base
RB2	READIT	142	base (after
RN1	READIT	135	nose
RN2	READIT	141	nose
SMULT, 25	ZPRM	MULT	multi
TH0	READIT	207	input
THETA1	READIT	134	cone in deg
THETA2	READIT	140	cone degre
TØFF	READIT	209	input input
TØN	READIT	208	time input

the

Description	Units
penalty value	-
specific impulse	seconds
RN1/RB1, bluntness ratio of the vehicle initial configuration	-
RN2/RB2, bluntness ratio of the vehicle second configuration (after shape change)	-
axial length of vehicle initial configuration	feet
axial length of vehicle second configuration (after shape change)	feet
vehicle geometry input code	-
corresponds to input quantity NTHRUST; values 0, 1, 2 indicate respectively no thrust, thrust vs. altitude, and thrust vs. time	-
common block	-
TT	-
base radius of the vehicle initial configuration	feet
base radius of the vehicle second configuration (after shape change)	feet
nose radius of the vehicle initial configuration	feet
nose radius of the vehicle after shape change	feet
multiplier on penalty function, preset to 1.0	
input reference thrust level	lb.
cone half angle of vehicle initial configuration in degrees	
cone half angle of vehicle second configuration in degrees (after shape change)	
input time of thrust shut off used when NTHRUST input is 2	seconds
time of thrust onset used when NTHRUST input is 2, input quantity	seconds

B

2. Input (Concl'd)

Name	Source	Common Block	
TST	READIT or SR2490	123	stopping t
T0	READIT	102	input initi
W1	READIT	133	initial we
W2	READIT	139	initial we (after sha
ZØFF	READIT	206	input altit input is 1
ZØN	READIT	205	input altit
ZST	READIT	121	input stop
ZTURN	READIT	145	altitude a

Description	Units
stopping time	seconds
input initial time	seconds
initial weight of vehicle first configuration	lb.
initial weight of the vehicle second configuration (after shape change)	lb.
input altitude of thrust shut off used when NTHRUST input is 1 (program code is NTHRST)	feet
input altitude of thrust onset used when NTHRUST is 1	feet
input stopping altitude	feet
altitude at which shape change occurs	feet

B

3. Output

Name	Description
ISUC *	integer control code
LAMDA1	See input
LAMDA2	See input
LA1	See input
LA2	See input
ØCCUR, 4000	See input
RB1	See input
RB2	See input
RN1	See input
RN2	See input
THETA1	cone half angle of vehicle initial configuration in degrees
THETA2	cone half angle of vehicle second configuration in degrees (after shape change)
VAL	penalty value for FEV

4. Numerical Procedure

As a first step. SUBROUTINE SCREEN sets each element of the A array and each element of the PENLTY array equal to zero. This is done in the DØ loop ending at statement 2. The following sixteen statements then initialize some elements of the A, DLL and DUL arrays. Next, ZTURN is tested. If ZTURN is less than zero, statement 10 is executed next. Otherwise, the values of five more A's are set equal to one. This is done in the DØ loop ending at 8, then statement 10 follows.

At 10, NTHRST is tested. If NTHRST equals zero, control passes to 15. Otherwise, A(11) is set equal to one and NTHRST is tested again. If NTHRST equals 2, control passes to 12. Otherwise, two more A's are set equal to one and control is transferred to statement 15.

At 12, two additional A's are set equal to one and statement 15 follows.

At 15, all of the A's required in calculating PENLTY array have been defined and NGEØM is tested. Of the six quantities W1 (weight), THETA1 (cone half angle), RN1 (nose radius), RB1 (base radius), LAMDA1 (RN1/RB1) and LA1 (axial length of vehicle), some are input and others must be calculated. These describe the initial geometry of the vehicle. The corresponding quantities W2, THETA2, RN2, RB2, LAMDA2 and LA2 describe the geometry immediately after shape change. These must be either input or calculated unless ZTURN is less than zero. The value of NGEØM specifies the calculations as follows.

If NGEØM equals one, W1, THETA1, RN1, and RB1 have been input, then LAMDA1 and LA1 must be calculated. W2, THETA2, RN2 and RB2 are also input, if ZTURN is greater than or equal to zero; similarly in this case LAMDA2 and LA2 must be calculated. These calculations are performed starting at statement 18.

If NGEØM equals 2, W1, THETA1, RB1, LAMDA1 and possibly W2, THETA2, RB2 and LAMDA2 have been input,but RN1, LA1, and possibly RN2 and LAMDA2 must be calculated. These calculations are performed starting at statement 30.

If NGEØM equals three, W1, RN1, RB1 and LA1 and if necessary W2, RN2, RB2 and LA2 have been input, but THETA1 and LAMDA1 and possibly THETA2 and LAMDA2 must be calculated. These calculations are performed starting at statement 40. After performing the necessary calculations, control always reaches statement 200 where the accumulator E is set equal to zero.

The DØ loop ending at statement 500 is then executed twenty-five times. ØCCUR(IZ) is tested to see if it lies between DLL(I) and DUL(I); statement 500 is executed next. If not, the value of PENLTY(I) is calculated and added to E at 500.

Next E is tested. If E still equals zero, control passes to statement 600. If not, ISUC is set equal to zero and control passes to 700.

At 600, ISUC is set equal to -1, a table of values useful for checking the error is printed,VAL is set equal to E + D, and control passes to statement 700.

At 700 control is returned to SUBROUTINE FEV.

5. Other Information

A. SUBROUTINE SCREEN is called by SUBROUTINE FEV only.

B. SUBROUTINE SCREEN references the IBM routines DCØS, DSIN and DSQRT and the Avco library function ASINR.

SUBROUTINE REDUCE (CUP, KRED, IRED, LRED, WRF)

1. Purpose

SUBROUTINE REDUCE can multiply any specified element of the ØCCUR array by the reduction factor WRF and set the first element of the CUP array equal to any specified element of the ØCCUR array multiplied by WRF.

2. Input

Name	Source	Common Block	Description
CUP, 20	FEV	-	vector of upper bounds of constraints
ICØM(3)	ZREADX	IXCØM	test parameter
ID1, 50	ZREADX	IDNØS	array containing ØCCUR locations of the design variables; read in as IDNØ
ID2, 50	ZREADX	IDNØS	array containing ØCCUR locations of the constraints; read in as IDC
IRED	FEV	-	iteration counter
ITERM	FEV	END	test parameter; non-zero value indicates a penalty function which is undefined or zero; zero value indicates a non-zero, defined penalty function.
LRED	FEV	-	iteration upper limit
ØCCUR, 4000	-	ØCCUR	common array
WRF	FEV	-	reduction factor

3. Output

Name	Common Block	Description
CUP, 20	-	See Input
IRED	-	See Input
KRED	-	test parameter
ØCCUR, 4000	ØCCUR	See Input

4. Numerical Procedure

SUBROUTINE REDUCE first defines IZ as the location in the ØCCUR array of the value which may be multiplied by WRF. The value of the iteration counter IRED is then increased by one and compared with LRED, the iteration upper limit. If IRED equals LRED, statement 100 is executed next. If IRED is less than IRED, ITERM is tested; control is transferred to 10, 60, and 50 when ITERM is respectively negative, positive and zero.

In the DØ loop starting at 10, the first IN elements of the ID1 array are compared with IZ. If one of these if found equal to IZ, statement 40 is executed next. If no equality is found, the statement following 20 will test ICØM(3). If ICØM(3) equals one, statement 40 is executed. If ICØM(3) equals zero, ØCCUR(IZ) is set equal to WRF*ØCCUR(IZ), KRED is set equal to zero and control is transferred to 200.

At 40 CUP(1) is set equal to WRF*ØCCUR(IZ). KRED is set equal to zero and control goes to 200.

At 50 KRED is set equal to 1 and statement 200 is executed. This is intended for possible future implementation.

At 60 KRED is set equal to 1 and control is transferred to statement 200.

At 100 KRED is set equal to -1 then statement 200 is executed.

At 200 control is returned to the main program where KRED is tested.

If KRED equals zero, MAIN sends control to the appropriate optimization technique for another iteration. Otherwise, data for a new case is read.

5. Other Information

A. SUBROUTINE REDUCE is called by the main program, MAIN, only.

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

2.2 Gradient of the Penalty Function

The subroutine FCN calculates the gradient of the penalty function defined by FEV. The partial derivatives are calculated with respect to the design variables using finite difference techniques.

SUBROUTINE FCN(N, G, F, X, M1)

1. Purpose

SUBROUTINE FCN must compute the function F and the gradient vector G (i.e., the vector whose elements are the first partial derivatives of F with respect to the parameters), given the N coordinates of the point X.

2. Input

*indicates integer quantity

Name	Common Location	Source of Input	Description
DELX, 20	DØPT	ZREADX or ZPRM	finite difference increment for each variable (should be smaller than anticipated step size)
F	-	FEV	penalty function value
FM		FEV	penalty function value
IØP, 90	IØCCUR(1-90)*	ZREADX or SR2490	input option code
ITERM	END *	FEV	integer control code; a zero value indicates that function F is non-zero and defined; a non-zero value indicates that F is either zero or undefined
M1	- *	DAVDØN, READY, AIM, FIRE	integer control code; is set to 1 for first call of FCN, to 2 for each subsequent call up to the list, to 3 for last entry to FCN.
MS	BLKI *	DAVDON or STUFF	integer counter for the number of random steps

2. Input (Concl'd)

Name	Common Location	Source of Input	Description
N	- *	DAVDØN	number of independent variables
NPLØT, 5	NØCCUR(24-28)	READIT	plotting option code
X, 40	-	DAVDØN, READY, AIM, FIRE	independent variable array

3. Output

Name	Common Location	Description
F	-	See Input
G, 40	-	the gradient of F at point X
IØP, 90	IØCCUR(1-90)*	See Input
N	- *	See input
NPLØT, 5	NØCCUR(24-28) *	plotting option code
NPRINT	NØCCUR(14) *	option code controlling detailed printout of trajectory information
X, 40	-	See Input

4. Numerical Procedure

SUBROUTINE FCN begins calculations with a DØ loop, ending at statement 416, which defines the 14 elements of the NPSAVE array as follows:

NPSAVE(1-6) = IØP(64-69) respectively

NPSAVE(7-11) = NPLØT(1-5) respectively

NPSAVE(12-14) = IØP(1-3) respectively

Then, if M1 equals 3, indicating the last call of FCN, the print code NPRINT is set equal to 1 to allow detailed trajectory printout, FEV is called to calculate the final value of F, NPRINT is reset to zero, and control passes to statement 106. If M1 is not equal to 3, control passes to statement 5.

Statement 5 calls SUBROUTINE FEV, then the quantity ITERM is tested. If ITERM is not equal to zero indicating a zero or undefined F, control passes to 106. If ITERM equals zero, the DØ loop ending with statement 417 is executed. Here the values of IØP (1-3, 64-69) and NPLØT (1-5) are zeroed. A second DØ loop ending with statement 105 is then employed to define the N elements of the G array. This is accomplished by increasing the Ith element of the X array by the Ith element of the DELX array to define the Ith term of new X array, then calling FEV for this element of X to define FM. ITERM is tested; if non-zero, control passes to 106; otherwise, the Ith term of the G array is defined as $G(I) = \frac{FM-F}{DELX(I)}$. The Ith element of DELX is then subtracted from the Ith term of the new X array to restore the original value to the X array before statement 105 is reached. Statement 106 is executed next.

Statement 106 is the beginning of a DØ loop ending at statement 418, which restores the values of IØP(1-3, 64-69) and NPLØT (1-5) which were stored in the NPSAVE array. Following statement 418, a test on M1 sends control to statement 1000, if M1 is not equal to 1. If M1 equals 1, first the internal quantities IRC and IC are set equal to 1, then statement 1000 causes

the quantities IC, MS, F, and the elements of the X array to be printed out. The WRITE statement following causes the elements of the G array to be printed out. Then IC is increased by 1 before the return to the calling program is executed.

5. Other Information

A. SUBROUTINE FCN is called by the following subroutines:

1. SUBROUTINE DAVDON
2. SUBROUTINE AIM
3. SUBROUTINE READY
4. SUBROUTINE FIRE

B. SUBROUTINE FCN calls in SUBROUTINE FEV.

2.3 Search Logic

The search logic for the four optimization techniques is explained in the following sections.

2.3.1 Davidon Variable Metric Optimization Method

The Davidon variable metric method is programed for machine computation in the six subroutines - DAVDON, READY, AIM, FIRE, DRESS, and STUFF - with specialized procedures of matrix multiplication and random number generation performed by MATMP and RANDOM respectively.

SUBROUTINE DAVDON is the controlling routine for the calculations of the Davidon method. The basic procedure is as follows: In order to calculate the initial value of the penalty function and its gradient, DAVDON calls SUBROUTINE FCN which in turn calls FEV. Next, SUBROUTINE READY is called to establish the direction along which the search for the relative minimum is to be carried out and to box off the interval in this direction within which the relative minimum is located. SUBROUTINE AIM estimates the location of the relative minimum within the selected interval (interpolated point) and compares this with the result of taking a step perpendicular to the direction chosen by READY. If the perpendicular step is an improvement, SUBROUTINE DRESS is called to revise the metric and the process repeated. If not, SUBROUTINE FIRE is called to evaluate the function at the interpolated point and to determine if the minimum has been sufficiently well located. If so, the rate of change of the gradient is evaluated by interpolating from its values at the initial, interpolated, and end points before SUBROUTINE DRESS is summoned. If not, SUBROUTINE AIM is recalled. SUBROUTINE DRESS modifies the metric on the basis of information obtained about the function in previous iterations. When the minimum has been sufficiently well located,

SUBROUTINE STUFF is called to test how well the function has been minimized and how well the metric approximates the matrix of second partial derivatives of the penalty function at the minimum. This is done by displacing the evaluation point from the location of the minimum in a random direction. Currently the random number generator is not being used and instead, at this point in the procedure, the current search is terminated and tests for the convergence of the sequential procedures are made in SUBROUTINE REDUCE.

SUBROUTINE DAVDØN

1. Purpose

SUBROUTINE DAVDØN is the controlling subroutine for the calculations of the Davidon variable metric method of minimization, which determines numerically the local minima of differentiable functions of several variables. In the process of locating each minimum, a matrix, $h^{\mu\nu}$, which characterizes the behavior of the function about the minimum is determined.

2. Input

*indicates integer quantity

Name	Symbol		Block Common Name	Source of Input	
C, 400	C_i		BLKØ		constr
DELTA1	Δ		FØPT	ZREADX	input e quanti
ERR	ϵ		FØPT	ZREADX	the pr preset
F	f		BLKØ	FCN or DRESS	the val
FAC1			FØPT	ZREADX	corres
G, 40	g_μ		BLKØ	FCN or DRESS	the val
GS	g_s		BLKØ	READY	the co
H, 1600	$h^{\mu\nu}$		BLKØ	ZREADX or DRESS	the no the sp
IN		*	IØPT	ZREADX	the pr
IRAND		*	IØPT	ZREADX	randor
ISEN1		*	SENSE	ZREADX	printou
ITERM		*	END	FEV	integer functio
LIMIT		*	IØPT	ZREADX	the ma total p direct
P			BLKØ	MAIN	multip
T, 40	t^μ		BLKØ	MATMP	interm in this
TØ	t_o		BLKØ	MATMP	interm in this

of	Description
	constraints on the matrix $h^{\mu\nu}$, not currently being used
X	input estimated initial value for determinant of H matrix input quantity is DELTA preset to 1.0
X	the program input stopping tolerance on the transformed gradient; preset to .01
DRESS	the value of the penalty function at the point x^{μ}
X	corresponds to program input FAC; see text
DRESS	the value of the gradient of the function evaluated at x^{μ}
	the component in the s direction of g_{μ}
X or DRESS	the non-negative symmetric matrix which willbe used as a metric in the space of the variables x^{μ}
X	the program input for the number of independent variables; preset to 1
X	random step size control; preset to 0
X	printout code which should be set to 0; preset to 0
	integer code: if < 0, indicates that the function is zero; if = 0, functions defined and non-zero; if > 0, function is undefined
X	the maximum number of iterations allowable; where an iteration is the total process of selecting a direction, bracketing the minimum in that direction and locating the minimum; preset to 30
	multiplier; preset to 0.0
	intermediate parameter in defining constrained $h^{\mu\nu}$ matrix, never used in this subroutine
	intermediate parameter in defining constrained $h^{\mu\nu}$ matrix, never used in this subroutine

B

3. Output

*indicates integer quantity

Name	Symbol	Block Common Name	Description
C, 400		BLKØ	see input
DELTA	Δ	BLKØ	the determinant of the $h^{(\mu,\nu)}$ matrix
E	ϵ	BLKØ	tolerance test parameter
F	f	BLKØ	the penalty function evaluated at $x^{(\mu)}$
FAC		BLKØ	see text
G, 40	g_μ	BLKØ	see input
GS	g_s	BLKØ	see input
H, 1600	$h^{(\mu,\nu)}$	BLKØ	see input
IT	*	BLKI	integer iteration counter
K	*	BLKI	random step control
L	*	BLKI	internal control code
M1	*	BLKI	integer control which is set equal to 1 for the first entry into FCN to 2 for subsequent entries up to the final entry, and to 3 for the final entry integer counter for the number of random steps
MS	*	BLKI	integer counter for the number of random steps
N	*	BLKI	number of independent variables

3. Output (Concl'd)

Name	Symbol	Block Common Name	Description
NC	*	BLKI	number of constraints
P		BLKØ	multiplier used in SUBROUTINE STUFF
T, 40	t_μ	BLKØ	see input
X, 40	x_μ	BLKØ	the set of N independent variables describing the point from which a step is taken to begin the search for a minimum

4. Numerical Procedure

SUBROUTINE DAVDØN begins with the setting of internal program parameters to the corresponding input values and the setting of certain initial values. Then, if the input quantity FAC is zero, control passes to statement 1066. This is the beginning of a DØ loop ending with statement 1068 which completes the formation of the symmetric H matrix by defining the lower left triangular portion from the input upper right triangular portion of H. If FAC is not zero, control passes to statement 106, the beginning of a DØ loop ending at 1064 which zeroes the non-diagonal elements of the H matrix and sets the diagonal elements equal to FAC. Printout of the initial H matrix and of the initial estimate of the independent variables, X(I), is executed. The counter M1 is set to 1 at the first entry into FCN; (it is set to 2 for subsequent entries up to the final entry and it is set to 3 for the final entry).

In statement 1165, the function F which is being minimized is set equal to zero, SUBROUTINE FCN is called to compute the function F and the gradient G, whose elements are the first partial derivatives of F with respect to the parameters. If the integer code ITERM is not equal to zero, which indicates that function F is either zero or undefined control passes to statement 149. If ITERM = 0 indicating that function F is non-zero and defined, then F itself is tested. If F is negative or zero, control passes to statement 1393; but, if F is positive, control passes to statement 118 where the integer code L is set equal to 1. The iteration counter integer IT is set equal to zero, before WRITE statements are

executed. In statement 1201 the quantity NC, the number of linear constraints on the parameters, is tested. Presently control always passes to statement 121, since NC is defined as zero at the beginning of DAVDØN. Thus statements 1202 through 1209, which allow for the future application of constraints on the parameters, are bypassed.

SUBROUTINE READY is called in by statement 121 to determine the search direction, s, the gradient in the search direction, g_S; if $|g_S| > \epsilon$, to estimate the step size in x to bracket the minimum along S; and to calculate f^+, g^+, and g_S^+, which are respectively f, g and g_S at the incremented x. If ITERM is not equal to zero for f^+, control passes to statement 149; otherwise the value of L is tested in statement 123 for ITERM = 0. If L equals 1, control passes to statement 139; if L equals 2, control passes to statement 159 where L is set equal to 4 and then to statement 133; if L equals 3, control passes to statement 137 where L is set equal to 3 and then to statement 133; if L equals 4, control passes to statement 126. Statement 126 calls SUBROUTINE AIM, which determines whether the minimum has been bracketed by the steps and, if it has, compares the minimum's location with the value which would be expected from a perpendicular step. After the return from SUBROUTINE AIM, ITERM is tested and, if non-zero, causes control to be sent to statement 149. If ITERM = 0, the integer code L is used in the directed GØ TØ of statement 128; if L = 1, control passes to 129; if L = 2 or 3, control passes to statement 133. Statement 129 calls SUBROUTINE FIRE to interpolate for the minimum along the search direction and calculate the function at the interpolated point. If the value of the function causes ITERM to be non-zero, upon return from FIRE, control passes to statement 149. If ITERM is zero, the directed GØ TØ passes control to statement 135, if L =1, there L is set to 2 before control passes to statement 133.

Similarly, if L = 2, control passes to statement 132 where L is set equal to 1 and then to statement 133. If L = 3, control passes to statement 126 which recalls SUBROUTINE AIM.

Statement 133, calls SUBROUTINE DRESS to modify the H matrix, increase the quantity IT by 1, and set F equal to FB, $\bar{f}$. Then, if $F \leq 0.0$ or IT > the limiting value ITLIM, control passes to statement 142. The directed GØ TØ of statement 134 causes control to pass to statement 124 if L = 1 or 162 if L = 2. Statement 124 sets L = 2 and passes to statement 1201 and the SUBROUTINE READY is recalled. If L = 2, control is returned to MAIN by statement 162.

Statement 139 tests the parameter ISEN1, a printout code (which should be set to zero), which, if ≤ 0 passes, control to statement 1395 and, if > 0, to statement 1393. Statement 1393 causes the X array to be printed out. Statement 1395, calls in SUBROUTINE STUFF which tests how well the function has been minimized and how well the matrix H approximates $\left\| \frac{\partial^2 f}{\partial x^{\mu} \partial x^{\nu}} \right\|$ at the minimum. Then in statement 141, the directed GØ TØ sends control to statement 1165 for the next iteration if L = 1 or to statement 142 if L = 2. Statement 142 causes the title "FINAL VALUES" to be written. The the interation counter IT is tested if IT $\leq$ the limiting value control passes to 143, otherwise, a message that the program has not converged is written before 143 is reached. Statements numbered 143 through 148 write the final output quantities. Statement 149 sets M1=3 before the calling of SUBROUTINE FCN for the final values of F and G. The test of ISEN1 in statement 151 is performed, if ISEN1 $\leq$ 0 control passes to the CONTINUE statement 152 and immediately reaches a GØ TØ 157 statement. If ISEN1 is > 0, control passes directly to 157, a CONTINUE statement preceding the RETURN statement 162.

5. Other Information

A. SUBROUTINE DAVDØN is called by the main program.

B. SUBROUTINE DAVDØN calls in the following subroutines:

1. SUBROUTINE FCN
2. SUBROUTINE MATMP
3. SUBROUTINE READY
4. SUBROUTINE AIM
5. SUBROUTINE FIRE
6. SUBROUTINE DRESS
7. SUBROUTINE STUFF

SUBROUTINE READY

1. Purpose

SUBROUTINE READY performs the calculations of the Davidon minimization method which establish a direction along which to search for a relative minimum and box off an interval in this direction within which a relative minimum is located. In addition, the criterion for terminating the iterative procedure is evaluated.

2. Input

Name	Symbol	Common Block	Source of Input	Description
DELTA	Δ	BLKØ	MAIN or ZREADX	the determinant of the matrix $h^{(\mu)}$
E	ϵ	BLKØ	DAVDØN	input stopping tolerance on the transformed gradient
F	f	BLKØ	DAVDØN or DRESS	the penalty function evaluated at the point $x^{(\mu)}$
FGSM		NALTFG	ZREADX	input mutliplier used in defining λ.
FP	f^+	BLKØ	FCN or FIRE	the value of the function at the point $x^{+(\mu)}$
G, 40	$g^{(\mu)}$	BLKØ	DAVDØN or DRESS	the gradient of the function at the point $x^{(\mu)}$
GP, 40	$g^{+(\mu)}$	BLKØ	FCN	the gradient of the function at the point $x^{+(\mu)}$

2. Input (Concl'd)

Name	Symbol	Common Block	Source of Input	Description
GS	g_s	BLKØ	MATMP	the component of g_μ in the s μ direction
GSP	g^+_s	BLKØ	MATMP	the component of g^+_μ in the S direction
H, 1600	$h^{\mu\nu}$	BLKØ	DAVDØN or DRESS	the non-negative symmetric matrix which will be used as a metric in the space of the variables x^μ
ITERM	-	END	FEV	integer code: if < 0, indicates that the function is zero; if = 0, indicates function is defined and non-zero; if > 0, function is undefined
L	-	BLKI	DAVDØN or DRESS	internal control code
N	-	BLKI	DAVDØN	number of independent variables
NALT	-	NALTFG	ZREADX	program input option; if = 0, the normal logic proceeds, if = 1 the procedure which describes the step size is used
S, 40	s^μ	BLKØ	MATMP	the direction along which the search for relative minimum is to proceed from x^μ
SL	l	BLKØ	AIM	the squared length of s^μ
X, 40	x^μ	BLKØ	MAIN, STUFF, or DRESS	the set of N independent variables describing the point from which the search proceeds

3. Output

Name	Symbol	Common Block	Description
DELTA	Δ	BLKØ	see input
FB	$\bar{f}$	BLKØ	the actual value of the penalty function at the interpolated point
GB, 40	$\bar{g}_{\mu}$	BLKØ	the value of the gradient of the function evaluated at the interpolated point
GP, 40	g^{+}_{μ}	BLKØ	see input
IT	-	BLKI	iteration counter
L		BLKI	see input
M1	-	BLKI	the integer control which is set equal to 1 for the first entry to FCN, to 2 for subsequent entries up to the final entry, and to 3 for the final entry
N	-	BLKI	see input
S, 40	s^{μ}	BLKØ	see input
SL	λ	BLKØ	see input
T, 40	t^{μ}	BLKØ	takes on the value of XP
TØ	t_o	BLKØ	1/SL
X, 40	x^{μ}	BLKØ	see input
XP 40	$x^{+\mu}$	BLKØ	the point described by the independent variables after a step of length λ along s from x^{μ}

4. Numerical Procedure

SUBROUTINE READY begins by testing the control integer L. If L = 1, the iteration counter IT is set equal to one; indicating the first iteration, before control passes to statement 201. If L = 2, control passes directly to statement 201 which calls SUBROUTINE MATMP and utilizes statements 202 and 203 to define the search direction, s, as

$$s^{\mu} = - h^{\mu\nu} g_{\mu}$$

SUBROUTINE MATMP is then called to compute g_s from $g_s = s^{\mu} g_{\mu}$. From the equations for s^{μ} and g_s, one sees that $- g_s$ is the squared length of g, thus the improvement to be expected in the function is $- 1/2\, g_s$. The positive definiteness of $h^{\mu\nu}$ insures that g_s is negative, so that the step is in a direction which, at least initially, decreases the function. If the decrease is written the accuracy desired, i.e., $g_s + \epsilon \geq 0.0$ in test of statement 206, then the minimum has been determined and control passes to statement 227 where L is set equal to 1 and control returns to DAVDON. If $g_s + \epsilon < 0$, the procedure continues by passing to statement 207 where TP1 is defined using the input FGSM as

$$TP1 = - FGSM(\frac{f}{g_s})$$

The quantities g_s, ϵ, TP1, and IT are then printed out, before λ, EL, is set equal to the smaller value of the quantities 2.0 and TP1 and λ, SL, is defined as $- g_s$. Then $x^{+\mu}$, the value of x after the step in the s direction, is evaluated from $x^{+\mu} = x^{\mu} + \lambda\, s^{\mu}$, M1 is set equal to 2, and SUBROUTINE FCN is called to compute the function

and its gradient to $x^{+\mu}$, f^{+} and g^{+}_{μ} respectively. If the function is zero or undefined, the code ITERM $\neq 0$, causing the X array to be equated to the XP array, i.e., $x^{\mu} = x^{+\mu}$, before executing a return to DAVDØN. If the function is defined and non-zero, ITERM = 0 and control passes to statement 214, where SUBROUTINE MATMP is called to yield the projection of the gradient at $x^{+\mu}$ in the direction of the step, g^{+}_{s}, which equals $s^{\mu} g^{+}_{\mu}$. If $g^{+}_{s} \geq 0.0$ or $f^{+} > f$, then there is a relative minimum along the direction s between x^{μ} and $x^{+\mu}$ and control passes to statement 229 where L is set equal to 4 and executes the return to DAVDON which then calls SUBROUTINE AIM to interpolate for the point. On the other hand if $g^{+}_{s} < 0.0$ and $f^{+} < f$, control passes to statement 218 which writes the message "UNDERSHOT" indicating that the step taken was too small. The input option code NALT is then tested. If NALT ≤ 0, the normal logic is pursued and control passes to statement 231. If NALT > 0, control passes to statement 10 where the step size λ, EL, is doubled and then to statement 210 to repeat the calculations with the new step.

Starting with statement 231, the following quantities are defined $\bar{f} = f^{+}$, $\bar{g} = g^{+}_{\mu}$, and $t^{\mu} = x^{\mu}$. If the step size λ is less than 2.0, control passes to statement 221 where L is set equal to 3 before the return to DAVDØN which then calls in SUBROUTINE DRESS to alter the $h^{\mu\nu}$ matrix.

If the step size $\lambda \geq 2.0$, control passes to statement 223 where $\Delta = 2.0\,\Delta$, $t_o = 1.0/\lambda$, and L = 2 are defined before the return to DAVDØN which redefines L as 4 and calls in SUBROUTINE DRESS.

5. Other Information

A. SUBRØUTINE READY is called by SUBRØUTINE DAVDØN.

B. SUBRØUTINE READY calls in SUBRØUTINE MATMP and SUBROUTINE FCN.

SUBROUTINE AIM

1. Purpose

SUBROUTINE AIM, part of the Davidon minimization method, estimates the location of the relative minimum within the interval selected by SUBROUTINE READY. This location is compared with the value that would be expected from a "perpendicular step".

2. Input

*indicates an integer quantity

Name	Symbol	Block Common Name	Source of Input	
EL	λ	BLKØ	READY, FIRE, or STUFF	the step
F	f	BLKØ	DAVDØN, DRESS, FIRE	the valu
FB	$\bar{f}$	BLKØ	FCN	the actu
FP	f^+	BLKØ	FIRE or READY	the valu
GB, 40	g	BLKØ	FCN	the grad
GP, 40	g^+	BLKØ	FIRE or READY	the valu
GS	g_s	BLKØ	FIRE or READY	the com
GSP	g_s^+	BLKØ	FIRE or READY	the com
GTP	g_t^+	BLKØ	MATMP	the comp
GTT	g_{tt}	BLKØ	MATMP	the com
H, 1600	$h^{\mu\nu}$	BLKØ	DAVDØN or DRESS	the non- the spac
ITERM	*	END	FCN	error co
N	*	BLKI	DAVDØN	the num
S, 40	s^μ	BLKØ	READY	the chos is to pro
SL	ℓ	BLKØ	READY	squared
T, 40	t	BLKØ	MATMP or READY	the step to s^μ th

	Description
RE, or STUFF	the step length along s from x^{μ} to $x^{+\mu}$
DRESS, FIRE	the value of the penalty function at the point x^{μ}
	the actual value of the penalty function at the interpolated point
ADY	the value of the penalty function at the point $x^{+\mu}$
	the gradient of the function at the interpolated point
ADY	the value of the gradient of the function at the point $x^{+\mu}$
ADY	the component in the S direction of the gradient at x^{μ}
ADY	the component in the S direction of the gradient at $x^{+\mu}$
	the component of g^{+} in the direction of the perpendicular step t^{μ}
	the component in the S direction of the gradient at the interpolated point
DRESS	the non-negative symmetric matrix which will be used as a metric in the space of the variables x^{μ}
	error code
	the number of independent variables
	the chosen direction along which the search for the relative minimum is to proceed
	squared length of s^{μ}
READY	the step to the optimum point in the N-1 dimensional surface perpendicular to s^{μ} through $x^{+\mu}$

B

2. Input (concl'd)

Name	Symbol	Block Common Name	Source of Input	
X, 40	x^{-1L}	BLKØ	MAIN, STUFF, DRESS, or FIRE	the set of N step is take
XP, 40	x^{+1L}	BLKØ	FIRE or READY	the set of N reached by

	Description
DRESS,	the set of N independent variables describing the point from which a step is taken to begin the search for a minimum
Y	the set of N independent variables describing the point which is reached by taking a step of length λ in the S direction from point $x^{(k)}$

B

3. Output

Name	Symbol	Block Common Name	Description
A	a	BLKØ	the percentage of λ by which the minimum in f is expected to precede $x^{+\mu}$
EL	λ	BLKØ	see input
FØ	f_o	BLKØ	the expected value of f at the minimum
GB, 40	g_{μ}	BLKØ	see input
GP, 40	g^{+}_{μ}	BLKØ	see input
GSS	g_{ss}	BLKØ	the component in the S direction of the quadratically interpolated value for the gradient at the interpolated point
GTT	g_{tt}	BLKØ	see input
H, 1600	$h^{\mu\nu}$	BLKØ	ses input
L	*	BLKI	internal control code
M	*	BLKI	an integer code output to MATMP to define the number of columns in MATMP input matrix and the number of elements in the resulting vector
M1	*	BLKI	integer control which is set equal to 1 for the first entry into FCN, to 2 for subsequent entries up to the final entry, and to 3 for the final entry.
N	*	BLKI	see input

3. Output (concl'd)

Name	Symbol	Block Common Name	Description
Q	Q	BLKØ	see text
S, 40	s^{μ}	BLKØ	see input
SL	l	BLKØ	see input
T, 40	t^{μ}	BLKØ	see input
TØ	t_o	BLKØ	the amount by which the minimum in f is expected to fall below f^+
X, 40	x^{μ}	BLKØ	see input
Z	z	BLKØ	see text

4. Numerical Procedure

SUBROUTINE AIM begins with the calculation of preliminary quantities by evaluating the following equations:

$$Z = g_s + g_s^+ + \frac{3(f - f^+)}{\ell}$$

$$t_o = g_s / Z$$

$$t_i = g_s^+ / Z$$

$$Q = \left| \sqrt{1.0 - t_o t_i * Z'} \right|$$

$$a = (g_s^+ + Q - Z) / (g_s^+ - g_s + 2.0Q)$$

For a step in the S direction, the amount to which the minimum in f is expected to fall below f^+ is t_o

$$t_o = \frac{\lambda a^2}{3} (g_s^+ + Z + 2.0Q)$$

Thus the expected value of f at the minimum then is $f_o = f^+ - t_o$.

SUBROUTINE MATMP is called in to determine the t^μ from $t^\mu = h^{\mu\nu} g_\nu^+$. The values of t^μ, the step to the optimum point in the N-1 dimensional surface perpendicular to s^μ through x^+_μ are redefined so that $t^\mu = -h^{\mu\nu} g_\nu + \frac{g_s^+ S}{\ell}$. SUBROUTINE MATMP is then called to evaluate the expression $g_t^+ = t^\mu g_\mu^+$, twice the change in f to be expected from step t_μ.

If the sum $2.0\, t_o + g_t^+$ is less than zero, control passes to statement 317; if greater than or equal to zero, control passes to statement 312. Statement 312 begins the calculations which redefine the t^μ to be

$$t^\mu = a\, x^\mu + (1.0 - a)\, x^{+\mu},$$

then set the control code L =1 before returning to DAVDØN which subsequently calls in FIRE to perform calculations at the interpolated point for the minimum along S.

Statement 317 tests the sum $f + g_t^+ / 2.0$; if this is less than zero, control passes to statement 312 and the procedure described above; if this is greater than or equal to zero control passes to statement 318. Statement 318 evaluates the function for the perpendicular step t by redefining $t^\mu = t^\mu + x^{+\mu}$, by setting the integer code M1 = 2, and by calling SUBROUTINE FCN at point t^μ to obtain the value of the function $\bar{f}$ and its gradient $\bar{g}_\mu$. If the error code ITERM ≠ 0; indicating that $\bar{f}$ is either zero or undefined, the x^μ are set equal to t^μ and control returns to DAVDØN. If ITERM = 0, indicating that $\bar{f}$ is neither undefined nor zero, control passes to statement 322.

Statement 322 compares the values of the function at the minimum point, $\bar{f}$ and f_o, as determined by the perpendicular step and the step along s respectively. If $\bar{f}$ is greater than f_o control passes to statement 312 and the procedure described previously for the use of the interpolated point is utilized. If $\bar{f}$ is smaller than f_o, meaning that the perpendicular point is to be used, control passes to statement 323.

Statement 323 writes the message "RICOCHET". The s^{μ} is redefined as $s^{\mu} = t^{\mu} - x^{+,\mu}$, whereupon SUBROUTINE MATMP is called to evaluate the expression $g_{tt} = s^{\mu} \bar{g}_{\mu}$, where g_{tt} is the component in the s direction of the gradient at the interpolated point. Then modifying g_{tt}, we have $g_{tt} = g_{tt} - \beta_t^+$. If this value of $g_{tt} < 0$, control passes to statement 335 where L is set to 3 and then to DAVDØN which calls in SUBROUTINE DRESS. If $g_{tt} \geq 0.0$, then g_{ss} is set equal to g_{tt}, ℓ is defined as $-\beta_t^+$, λ is set equal to 1.0 and L = 2 before the return to DAVDØN which then calls in DRESS.

5. Other Information

A. SUBROUTINE AIM is called by DAVDØN.

B. SUBROUTINE AIM calls in

1. SUBROUTINE FCN
2. SUBROUTINE MATMP

C. SUBROUTINE AIM calls in the library functions

1. DSQRT
2. DABS

SUBROUTINE FIRE

1. Purpose

SUBROUTINE FIRE, part of the calculations for the Davidon minimization method, evaluates $\bar{f}$ and $\bar{g}$, respectively the penalty function equation and its gradient at the interpolated point, and determines if the local minimum has been sufficiently well located. If so, then the rate of change of gradient is evaluated by interpolating from its values at the end points of the interval being considered and at the interpolated point.

2. Input

Name	Symbol	Block Common Name	Source of Input
A	a	BLKØ	AIM
E	ϵ	BLKØ	DAVDØN
EL	λ	BLKØ	READY, AIM, or STUFF
F	f	BLKØ	DAVDØN or DRESS
FB	$\bar{f}$	BLKØ	FCN
FP	f^{+}	BLKØ	READY
G, 40	g_{μ}	BLKØ	DAVDØN or DRESS
GB, 40	$\bar{g}_{\mu}$	BLKØ	FCN
GSB	$\bar{g}_{s}$	BLKØ	MATMP
ITERM	*	END	FCN
N	*	BLKI	DAVDØN
Q	Q	BLKØ	AIM
S, 40	s_{μ}	BLKØ	READY
T, 40	t_{μ}	BLKØ	AIM

rce of nput	Description
	the percentange of λ by which the minimum in f is expected to precede the $x^{+\mu}$ point
VDØN	input stopping tolerance on the transformed gradient
ADY, , or STUFF	the length of step along s from x^{μ} to $x^{+\mu}$
VDØN or DRESS	the value of the penalty function at the point x^{μ}
N	the actual value of the penalty function at the interpolated point
ADY	the value of the penalty function at the point $x^{+\mu}$
VDØN or DRESS	the value of the gradient of the function at x^{μ}
N	the value of the gradient of the function at the interpolated point
TMP	the component in the s direction of GB
N	error code
VDØN	the number of independent variables
	see text of AIM
ADY	the direction chosen by READY along which the search for the relative minimum is to proceed.
	the step to the optimum point in the N-1 dimensional surface perpendicular to s^{μ} through $x^{+\mu}$

B

3. Output

Name	Symbol	Block Common Name	Description
EL	λ	BLKØ	see input
FP	f^+	BLKØ	see input
G, 40	g_μ	BLKØ	see input
GB, 40	$\bar{g}_\mu$	BLKØ	see input
GP	g^+_μ	BLKØ	the value of the gradient of the function at the point x^+_μ
GS	g_s	BLKO	the component in the s direction of the gradient of the function at x_μ
GSP	g^+_s	BLKØ	the component in the s direction of the gradient of the function at x^+_μ
GSS	g_{ss}	BLKØ	the component in the s direction of the quadratically interpolated value for the gradient at the interpolated point
L	*	BLKI	internal control code
M	*	BLKI	an integer code output to MATMP to define the number of columns in the MATMP input matrix and the number of elements in the resulting vector
M1	*	BLKI	the integer counter; which = 1 for first entry to FCN, = 2 for subsequent entries up to the final entry, and 3 for the final entry
N	*	BLKI	see input
S, 40	s_μ	BLKØ	see input

3. Output (Concl'd)

Name	Symbol	Block Common Name	Description
T, 40	t	BLKØ	see irput
TØ	t_o	BLKØ	$g_s \left(\frac{a}{1-a} - \frac{1-a}{a}\right)$
X, 40	x	BLKØ	the set of N independent variables describing the point from which a step is taken to begin the search for a minimum

4. Numerical Procedure

SUBROUTINE FIRE begins by setting the integer code M1 = 2 and then calls SUBROUTINE FCN to evaluate the penalty function and its gradient at the interpolated point, FB and GB respectively. If the function at the interpolated point is defined and non-zero, ITERM equals zero and control passes to statement 402. If the function is zero or undefined, the X array is given the value of the T array.

Statement 402 sets integer M = 1, then SUBROUTINE MATMP is called to multiply the components of the gradient GB by the components vector S to obtain GSB, the component of the gradient GB in the s direction. The smaller value of f and f^+, F and FP respectively, is defined as TP1. According to statement 405, should FB, $\bar{f}$, be greater than the sum of TP1 + ϵ, the interpolation is not considered satisfactory, and control passes to statement 418. At 418, f is tested against f^+ to determine that part of the original interval for which the function at the end point is smaller, then a new interpolation is made within that part. If f^+, FP, is smaller than or equal to f, F, control passes to statement 428; if f is smaller than f^+, control passes to statement 419.

Should FB, $\bar{f}$, be less than or equal to the sum TP1 + ϵ, control passes to statement 406. There, in the succeeding statements, the equation $t_o = \bar{g}_s \left(\frac{a}{1-a} - \frac{1-a}{n} \right)$ is evaluated and the t value compared with Q. If $|t_o| \geq Q$, g_{ss}, the component in the s direction of rate of change of the gradient at the interpolated point times λ, is defined as 2.0 * Q, since the interpolated point is a minimum and $\bar{g}_s = 0$, then L is set equal to 1 before

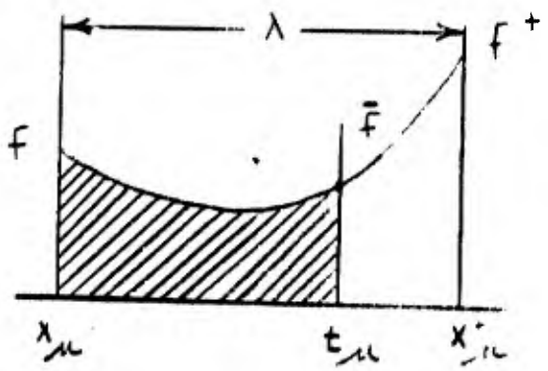
returning to DAVDØN which then calls in SUBROUTINE DRESS. If $|t_o| < Q$, we define

$$g_{ss} = \bar{g}_s \left(\frac{a}{1-a} - \frac{1-a}{a}\right) + 2.0Q$$

$$g_{\mu\mu} = \left(\bar{g}_{\mu\mu} - g_{\mu\mu}\right)\left(\frac{a}{1-a}\right) + \left(g^{+}_{\mu} - \bar{g}_{\mu\mu}\right)\left(\frac{1-a}{a}\right)$$

and L = 2 before returning to DAVDØN which then calls in SUBROUTINE DRESS.

At statement 419, the message "MOVE LEFT" is written to indicate that the new interpolation is to be made at a point to the left of the current interpolation point.



Shaded region is area in which the new interpolation will be made.

The quantity EL, λ, the length of the step, is redefined as $(1-a)\lambda$; f^+ and g^+_s, FP and GSP are given respectively the values of $\bar{f}$ and $\bar{g}_s$, FB and GSB; x^+_μ and g^+_μ, XP and GP, are given respectively the values of t_μ and $\bar{g}_\mu$ T and GB; and L is set equal to 3 before the return to DAVDØN which then calls in SUBROUTINE AIM to find a new interpolated point.

Similarly, at statement 428, the message "MOVE RIGHT" is written to indicate that the new interpolation is to be made at a point to the right of the current interpolation point.

The quantity EL, λ, is redefined as ($a\lambda$); f and g_s, F and GS, are given respectively the values of $\bar{f}$ and $\bar{g}_s$, FB and GSB; x_μ and g_{μ}, X and G, are given respectively the values of t_μ and $\bar{g}_{\mu}$, T and GB; then L is set equal to 3 and control returns to DAVDØN which calls in AIM to interpolate for the new point.

5. Other Information

A. SUBROUTINE FIRE is called by SUBROUTINE DAVDØN.

B. SUBROUTINE FIRE calls in SUBROUTINE FCN and SUBROUTINE MATMP.

C. SUBROUTINE FIRE calls in IBM function DABS.

SUBROUTINE DRESS

1. Purpose

SUBROUTINE DRESS, part of the calculations for the Davidon minimization technique, modifies the metric $h^{\mu\nu}$ on the basis of information obtained about the function along the direction under consideration, s. The new $h^{\mu\nu}$ is to have the property that $(h^{\mu\nu})' g_{\nu s} = \lambda s$, and must retain the information which the preceding iterations had given about the function.

2. Input

Name	Symbol	Block Common Name	Source of Input	
DELTA	Δ	BLKØ	MAIN or ZREADX	the c
E	ϵ	BLKØ	DAVDØN	inpu
EL	λ	BLKØ	READY, AIM, or FIRE	the l
FB	$\bar{f}$	BLKØ	READY, AIM, or FIRE	the a
G, 40	g_{μ}	BLKØ	DAVDØN or FIRE	the v
GB, 40	$\bar{g}_{\mu}$	BLKØ	READY, AIM, or FIRE	the v
GSS	g_{ss}	BLKØ	AIM or FIRE	the c for t
H, 1600	$h_{\mu s}$	BLKØ	DAVDØN or DRESS	the n in th
ISEN1	-	SENSE	ZREADX	a pri
IT	-	BLKI	READY, DAVDØN, or DRESS	itera
L	-	BLKI	READY, AIM, or FIRE	inter
N -		BLKI	DAVDON	the n
S, 40	s_{μ}	BLKØ	MATMP	the c is to
SL	l	BLKØ	READY or AIM	squa
T, 40	t_{μ}	BLKØ	READY or AIM	eithe perp
TØ	t_o	BLKØ	MATMP	the p
X, 40	x_{μ}	BLKØ	MATMP	the s step

	Description
	the determinant of the $h^{\mu\nu}$ matrix
	input stopping tolerance on the transformed gradient
	the length of step along s from x^{μ} to $x^{+\mu}$
	the actual value of the penalty function at the interpolated point
	the value of the gradient of the function at x^{μ}
	the value of the gradient of the function evaluated at the interpolated point
	the component in the S direction of the quadratically interpolated value for the gradient at the interpolated point
	the non-negative symmetric matrix which will be used as a metric in the space of the variables x^{μ}
	a printout code which should be set equal to 0; preset to 0.
RESS	iteration counter
	internal control code
	the number of independent variables
	the chosen direction along which the search for the relative minimum is to proceed
	squared length of s^{μ}
	either the step to the optimum point in the N-1 dimensional surface perpendicular to s^{μ} through x^{μ} or the interpolated step along s^{μ}
	the product $x^{\mu} g_{\mu}$
	the set of independent variables describing the point from which a step is taken to begin the search for a minimum

B

3. Output

Name	Symbol	Block Common Name	Description
DELTA	Δ	BLKØ	see input
F	f	BLKØ	the penalty function evaluated at point x^{μ}
G, 40	g_{μ}	BLKØ	the gradient of the penalty function evaluated at point x^{μ}
H, 1600	$h^{\mu\nu}$	BLKØ	see input
IT	-	BLKØ	see input
L		BLKI	see input
M		BLKI	an integer code output to MATMP to define the number of columns in MATMP input matrix and the number of elements in the resulting vector
N		BLKI	see input
TØ	t_o	BLKØ	see input
X, 40	x_{μ}	BLKØ	see input

4. Numerical Procedure

Upon entry into SUBROUTINE DRESS, the control code L is tested and directs the flow of the subroutine. If L has the value 1, control passes to statement 500; for a value of 2, to statement 525; for a value of 3, to statement 519; for a value of 4, to statement 510.

Statement 500 calls SUBROUTINE MATMP first to evaluate the expression

$$x^{\mu} = h^{\mu\nu} g_{\mu}$$

and then to evaluate $t_o = x^{\mu} g_{\mu}$. The latter is used to define TP1

$$TP1 = \ell - \frac{g_{ss}^2}{t_o} - \epsilon$$

which is tested and directs control to statement 505, if ≥ 0.0, or to statement 524, if < 0.0. Statement 505 begins a double DØ loop ending with 507 which determines an intermediate value of $h^{\mu\nu}$ from

$$(h^{\mu\nu})_{int.} = (h^{\mu\nu})_{previous} - \frac{x_{\mu} x_{\nu}}{t_o}$$

This is followed by computing a new Δ and a new t_o from

$$\Delta = \Delta \lambda \quad g_{ss}/t_o$$

$$t_o = \lambda / g_{ss}$$

before passing to statement 510.

Statement 524 writes the message "COLINEAR", then starting with statement 525 the following are defined

$$\Delta = \Delta\lambda \, l \, / \, g_{ss}$$

$$t_o = \left(\frac{\lambda \, l}{g_{ss}} - 1.0 \right) / l$$

before passing control to statement 510.

Statement 510 is the beginning of a double DØ loop ending with 512 which computes the new $h^{\mu\nu}$ matrix

$$h^{\mu\nu} = (h^{\mu\nu})_{\text{int.}} + t_o \, s^{\mu} \, s_{\nu} .$$

Statement 519, then gives the new function f at x^{μ} the value of $\bar{f}$, the value of function f at the interpolated point. Similarly, g_{μ} is given the value of $\bar{g}_{\mu}$ and x_{μ} the value of t_{μ}.

A test for overflow is made. If an overflow exists control passes to statement 531, where appropriate error messages and program quantities are written out. L is set equal to 2, and control returns to DAVDØN which ends the case by returning to MAIN. If no overflow exists, control passes to statement 513 where appropriate output messages are written, the iteration counter, IT, is then increased by 1, and L is set to 1 before control returns to DAVDØN, where L is reset to 1 and the iteration continues with the calling of SUBROUTINE READY.

5. Other Information

A. SUBROUTINE DRESS is called by DAVDON

B. SUBROUTINE DRESS calls in SUBROUTINE MATMP.

C. SUBROUTINE DRESS calls in the library routines

1. OVERFL
2. FDXPI

SUBROUTINE STUFF

1. Purpose

SUBROUTINE STUFF, part of the Davidon minimization method, tests how well the function has been minimized and how well the matrix $h^{\mu\nu}$ approximates $\left\| \frac{\partial^2 f}{\partial x^\mu \partial x^\nu} \right\|^{-1}$ at the minimum. This is done by displacing the point x from the location of the minimum in a random direction. The displacement of point x is chosen to be a unit length in terms of $h^{\mu\nu}$ as a metric. When $h^{\mu\nu} = \left\| \frac{\partial^2 f}{\partial x^\mu \partial x^\nu} \right\|^{-1}$, such a step will increase f by half the square of the step length.

2. Input

*indicates integer quantity

Name	Symbol	Common Block	Source of Input	Description
GS	g_s	BLKØ	READY	the component in the S direction of g^μ
H, 1600	$h^{\mu\nu}$	BLKØ	DAVDØN or DRESS	the non-negative symmetric matrix which will be used as a matrix in the space of the variable x^μ
K	- *	BLKI	MAIN	random step control
MS	- *	BLKI	DAVDØN or STUFF	integer counter for the number of random steps
N	- *	BLKI	DAVDØN	number of independent variables

2. Input (Concl'd)

Name	Symbol	Common Block	Source of Input	
P	-	BLKØ	MAIN	multiplier used in finding EL
S, 40	s^{μ}	BLKØ	MATMP or READY	the random direction or the direction chosen by READY for the search for minimum
TP1	-	BLKØ	MATMP	the product $s^{\mu} t^{\mu}$
X, 40	x^{μ}	BLKØ	MAIN, DRESS, READY	the set of N independent variables describing the point from which a step is taken to begin the search for a minimum

3. Output

*indicates integer quantity

Name	Symbol	Common Block	
EL	λ	BLKØ	the length of step along S from x^{μ} to $x^{+\mu}$
H, 1600	$h^{\mu\nu}$	BLKØ	the non-negative symmetric matrix which will be used as a metric in the space of the variables x^{μ}
K	- *	BLKI	random step control
L	- *	BLKI	internal control code
M	- *	BLKI	an integer code output to MATMP to define the number of columns in the matrix input to MATMP and the number of elements in the resulting vector

3. Output (Concl'd)

Name	Symbol	Common Block	Description
M1	- *	BLKI	the integer control code which is = 1 for the first entry to FCN; = 2 for all subsequent entries up to the final entry; = 3 for the final entry
MS	- *	BLKI	the integer counter for the number of random steps
N	- *	BLKI	number of independent variables
S, 40	$s^{(M)}$	BLKØ	the product of $h^{(M)}$ $t^{(M)}$
T, 40	$t^{(M)}$	BLKØ	the step in random direction
X, 40	$x^{(M)}$	BLKØ	new initial point described by N variables from which search for minimum will be initiated

4. Numerical Procedure

SUBROUTINE STUFF redefines K, the number of random steps to be taken, as K-1, then tests this quantity. If K is negative control passes to statement 617; if K is zero or positive, which indicates that the specified number of random steps to be taken has not been reached, control passes to statement 602.

In statement 617, since the specified number of random steps has been taken, the control code L is set equal to 2, then MS is set to zero before control is returned to DAVDØN which calls in FCN to determine the final values of f and $g^{(M)}$.

In statement 602 MS is increased by 1, then the values MS and GS are written out. A DØ loop then defines the N values of t^{μ} as the function RANDØM - 0.5. SUBRØUTINE MATMP is call in first to evaluate the expression $s^{\mu} = h^{\mu\nu} t^{\nu}$ and then to define TP1 = $s^{\mu} t^{\mu}$. The quantity TP1 is redefined as $\sqrt{s^{\mu} t^{\mu}}$ and used in the expression for λ, EL, which is $\lambda = \frac{P}{\sqrt{s^{\mu}, t^{\mu}}}$. The N values of x are computer from the expression

$$x^{\mu} = x^{\mu} + \lambda \; s^{\mu}$$

after which M1 is set to 2 and L to 1 before control returns to DAVDØN where the iteration continues.

5. Other Information

A. SUBRØUTINE STUFF is usually used to trigger the shut down of the optimization technique by setting the program input IRAND (which becomes K) to zero.

B. SUBRØUTINE STUFF is called by SUBRØUTINE DAVDØN.

C. SUBRØUTINE STUFF calls in SUBRØUTINE MATMP and FUNCTION RANDØM (which is not being used)

D. SUBRØUTINE STUFF calls in the library function DSQRT.

FUNCTION RANDØM(X)

1. Purpose

At present, this is a dummy function which merely sets RANDØM equal to the input argument X. It may be implemented later to actually calculate a random number.

2. Input

Name	Source of Input	Description
X	STUFF	dummy argument

3. Output

Name	Description
RANDØM	dummy variable

4. Numerical Procedure

FUNCTION RANDØM sets RANDØM equal to the dummy input variable X and returns control to SUBROUTINE MISC. This function may be implemented later to actually calculate a random number.

5. Other Information

A. SUBROUTINE RANDOM is called by SUBROUTINE STUFF only.

B. SUBROUTINE RANDOM does not call or reference any other subprograms.

SUBROUTINE MATMP(M, N, H, G, S)

1. Purpose

The vector G is multiplied on the left by the matrix H to produce a new vector S.

2. Input

Name	Source	Description
M	calling program	number of columns in matrix H and number of elements in vector S
N	calling program	number of rows in matrix H and number of elements in vector G
C	calling program	vector with N elements
H	calling program	matrix with N rows and M columns

3. Output

Name	Description
S	Vector product of matrix H times vector G

4. Numerical Procedure

The nth element of the product vector S is produced by scalar multiplication of the nth column of matrix H by the vector G.

5. Other Information

A. SUBROUTINE MATMP is called by each of the six subroutines:

AIM

DAVDØN

DRESS

FIRE

READY

STUFF

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

2.3.2 Rosenbrock Method

The following section contains the subroutines ROSBRK and GRAM for the Rosenbrock Rotating Coordinate Minimization Technique, the most powerful of the multidimensional direct search techniques.

The technique first searches for an approximate local minimum along a direction parallel to the first design variable axis. After a set of trials has been completed in one direction and an approximate local minimum has been established, the program searches from that point along the next orthogonal direction. This process is continued until all N directions have been treated. A new set of directions is then calculated by SUBROUTINE GRAM. All of the trials along the N directions and the subsequent calculation of a new set of directions is called a "stage."

SUBROUTINE ROSBRK

1. Purpose

SUBROUTINE ROSBRK is intended to minimize a function $U(P_1, P_2, \ldots, P_{NMAX})$ where the P's are adjustable design parameters.

2. Input

*indicates integer quantity

Name	Common Location	Source of Input	Description
ALPHA	XCØM(1)	ZREADX	multiplier used to obtain new step size from previous step size for a successful step; preset to 3.0
BETA	XCOM(2)	ZREADX	multiplier whose negative is used to obtain new step size from previous step size after an unsuccessful step, preset to 0.5
C,400	BLKØ	GRAM	matrix used in defining new coordinate system
DEL	XCØM(4)	ZREADX	multiplier used in obtaining ULIM, the limiting value of the function U; preset to 0.01
E0,20	DØPT	ZREADX	initial step sizes in units of corresponding parameters
GAMMA	XCØM(3)	ZREADX	multiplier used to redefine step size after a failure; preset to 0.5

2. Input (Concl'd)

Name	Common Location	Source of Input	Description
ITERM*	END	FEV	integer code; a value of -1 indicates that U is zero, 1 that U is undefined, and 0 that U is defined and non-zero; preset to 1.
LIMIT*	IØPT	ZREADX	the maximum number of times that a parameter can be changed successively; preset to 30.
NMAX*	IØPT	ZREADX	the number of parameters (design variables) of which U is a function; preset to 1
P, 40	BLKØ	MAIN	independent variables
RATU	XCØM(5)	ZREADX	input tolerance on quantity URAT, see text; preset to 0.5
TØL	XCØM(6)	ZREADX	input accuracy test parameter; preset to .0001
U	-	FEV	the function of P's being minimized

3. Output

Name	Common Location	
C, 400	BLKØ	unit matrix defined in RØSBRK
D, 40	BLKØ	array of the change from the initial value of each variable necessary to reach variable value for minimum in U
P, 40	BLKØ	array of design variable values which define minimum in U

4. Numerical Procedures

U is a function of the parameters $P_1, P_2, \ldots, P_{NMAX}$. Initial values for these P's are received through the BLKØ common block. A vector E0 of initial step sizes is received through the DØPT common block. The elements of a vector E are set equal to the corresponding elements of the E0 vector. This is necessary because the values of the E's may be changed by RØSBRK in the minimization process and the E0 vector may be needed as input to the next case.

RØSBRK tries to minimize the value of U by varying the P's one at a time and applying test criteria to the successive values of U. The value of U for a given set of P's is calculated by the function evaluator FEV. Each time FEV is called the variable ITERM, received from FEV through the IØPT common block, is tested. If ITERM equals -1, U= 0, and the minimization is complete since we know that U is always greater than or equal to zero. If ITERM equals 1, FEV has been unable to define U, and the case is a failure. In either case, control is returned to the main program. If ITERM equals zero, NTRIA, the number of times the same parameter has been changed successively, is tested. If NTRIA is greater than the input quantity LIMIT, the case is considered a failure and control is returned to the main program.

The general procedure is as follows. As a first step, FEV is called to obtain an initial value of U. If U is not zero, its value is saved as ULAST and the value of the first parameter is increased by the value of the first step size E_1. FEV is called again to compute a new value of U.

If the difference between the new U and the previous U, U-ULAST, is less than or equal to an input test parameter, TØL, times ULAST the trial is called a success and control passes to 109. If not, the trial is a failure and control passes to 110. If the trial is successful, the step size E_1 is multiplied by ALPHA to increase its value. The latest value of P_1 is then increased by the product of the new E_1 and the unit matrix C, then FEV is called again. If the trial is a failure, the value of E_1 is multiplied by minus BETA, the value of P_1 is increased by the product of the new E_1 and the unit matrix C, and FEV is called again.

If each of twenty successive trials with different values of the same parameter result in success, the advance criterion is said to be satisfied. Also, if a success if followed by a failure, the advance criterion is satisfied. Whenever the advance criterion is satisfied, RØSBRK stops changing the parameter it has been using and starts to change the next parameter. If the criterion has been satisfied for each of the NMAX parameters, the end of a stage has been reached. At the end of the second and subsequent stages, stopping criteria are applied. If these are all satisfied, the entire case is called a success and control is returned to the main program. If any one of these criteria is not satisfied, ROSBRK starts the whole procedure over again by varying the first parameter.

When statement 115 has been reached, UPREV, USTAG and ULAST are the values of U at the ends of three successive stages. UPREV is for the first of the three stages completed. USTAG is for the second, and ULAST is for the latest. The following requirements must all be satisfied if the latest value of U is to be accepted as a satisfactory minimum.

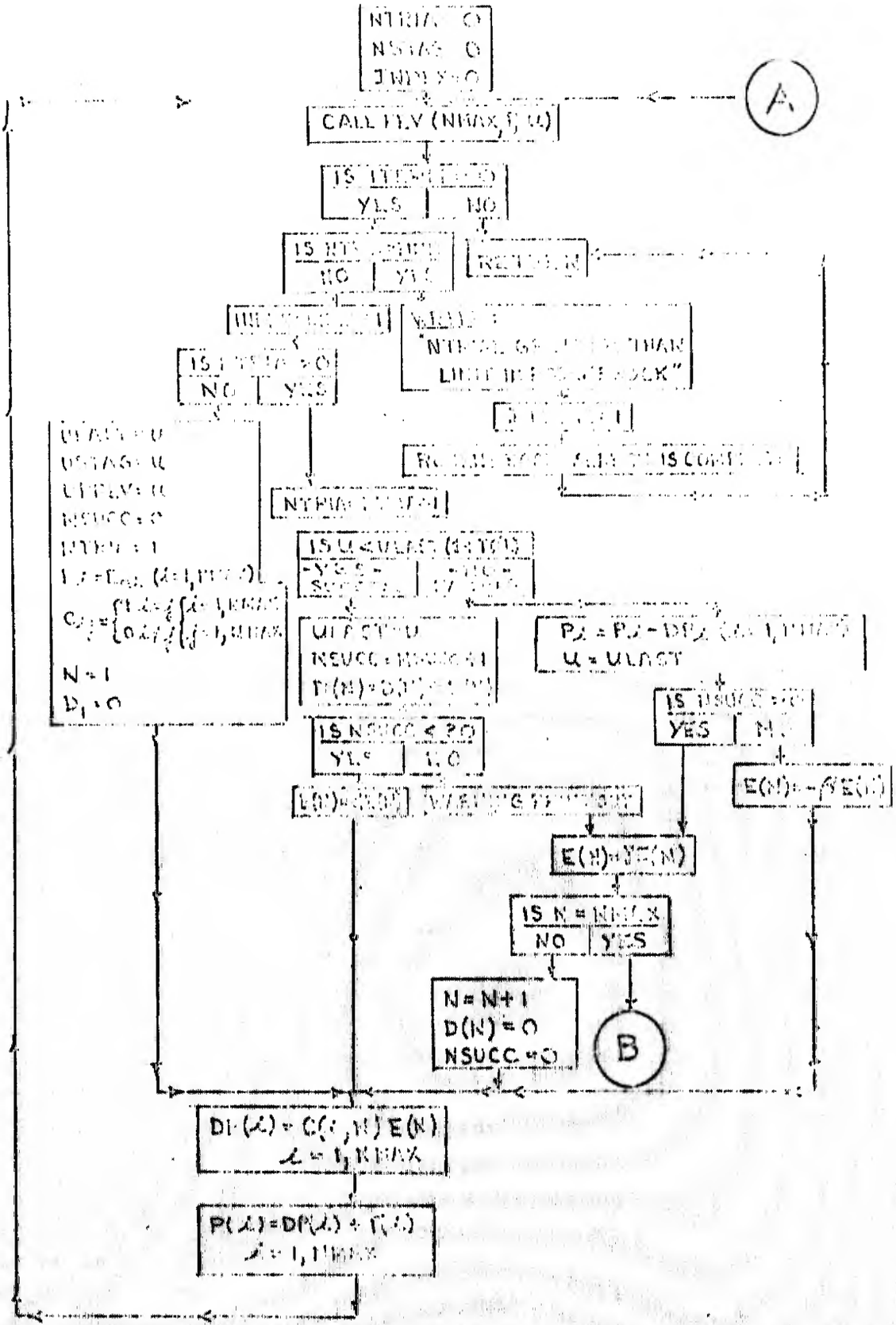
UPREV must not equal USTAG because URAT which equals (USTAG-ULAST) / (UPREV-USTAG) cannot be defined. ULAST and USTAG must both be greater than ULIM where ULIM is defined as UPREV-DEL*UPREV. Finally, URAT must be less than or equal to the input tolerance RATU.

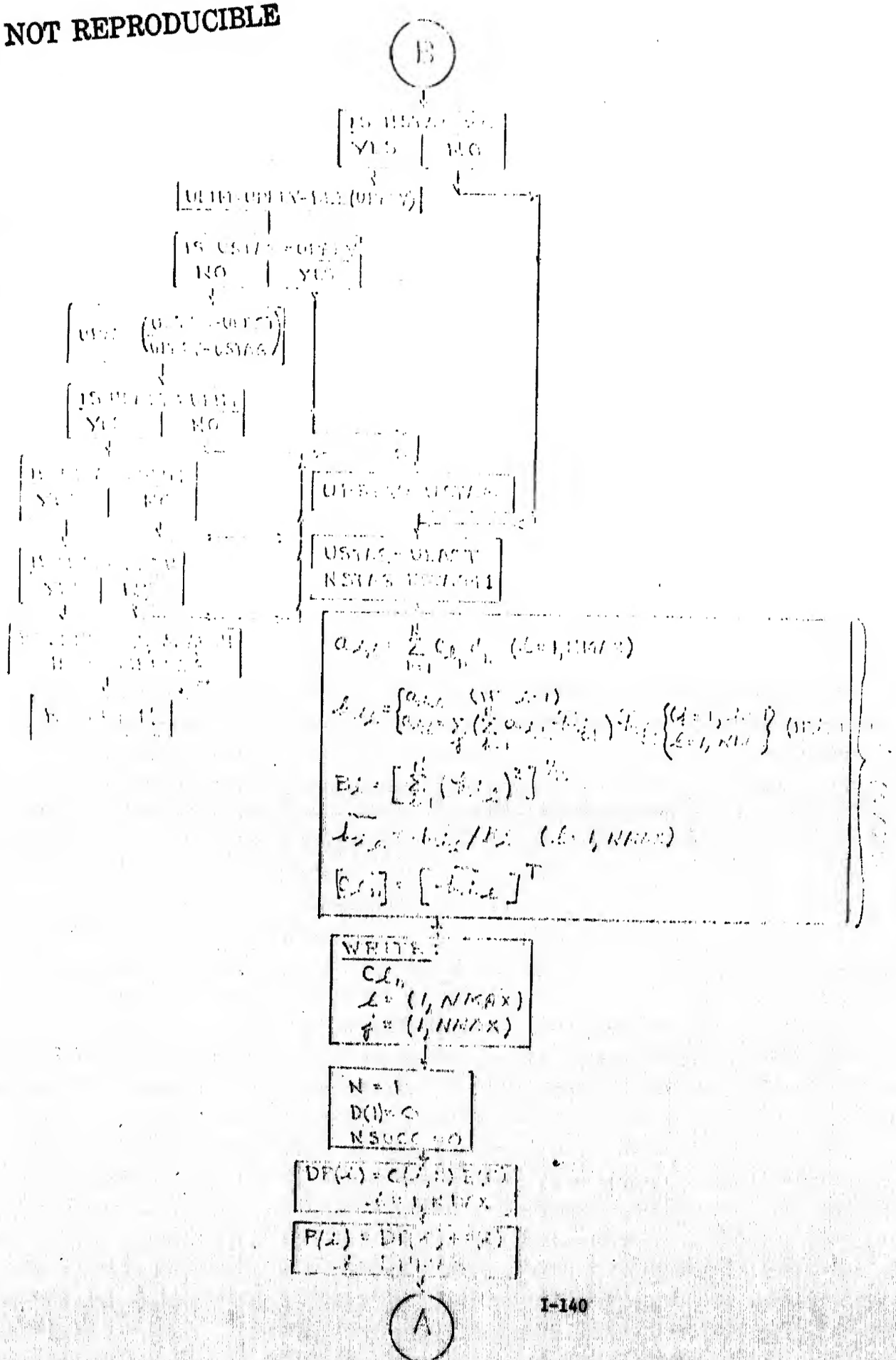
At statement 121, the value of USTAG is saved as UPREV and statement 116 is reached. At statement 116, ULAST is saved as USTAG. NSTAG, the number of stages, is then increased by one and GRAM is called to define a new set of coordinates. The C matrix, containing these coordinates as rows, is printed. Then NSUCC is set to zero, D is set to zero and statement 108 is reached to prepare for a new stage. At 108, printed output is produced before statement 125 is reached. At 125, new DP's are defined and these are used to redefine the P's. Then printed output is produced and control is returned to statement 123 for a new trial.

5. Other Information

A. SUBROUTINE RØSBRK is called by the main program only.

B. SUBROUTINE RØSBRK calls subroutines GRAM and FEV.





SUBROUTINE GRAM

1. Purpose

Given a matrix C and a column vector D, SUBROUTINE GRAM defines a set of vectors, A, which are used to calculate an orthogonal set of vectors, B, to be used as the row vectors of a matrix BB. The input matrix C is then redefined as the inverse (transpose) of matrix BB.

2. Input

*indicates integer quantity

Name	Common Block	Source of Input	Description
C, 400	BLKØ	RØSBRK	orthonormal matrix
D, 20	BLKØ	RØSBRK	vector of distances
NMAX*	IØPT	ZREADX	number of adjustable parameters

3. Output

Name	Common Block	Description
C, 400	BLKØ	redefined C matrix

4. Numerical Procedure

Given a matrix C, of order NMAX, and a vector D, of length NMAX, a set of NMAX vectors, A, each of length NMAX, is defined using C and D. Then a new set of orthonormal vectors is produced from these A vectors using the Gram-Schmidt equations to transform the A's. Finally, the matrix C is redefined as the inverse (transpose) of this orthonormal matrix.

The Gram-Schmidt equations are as follows:

1. Ui = Bi / (norm of Bi)
2. $B_1 = A_1$
3. $B_{I+1} = A_{I+1} - \sum_{K=1}^{I} (B_K \text{ dot } A_{I+1}) \times B$

The quantity in parentheses is the scalar product of the Kth B vector and the (I + 1) A vector.

As a preliminary step, the elements of the A vector, the B vector and the BB matrix are all set equal to zero.

Next, the DØ loop ending at 102 is entered to define the row vectors of the BB matrix. First, the A vector must be defined. This is done in the DØ loop ending at 101. The Kth element of the first A vector is the scalar product of Kth row vector of the C matrix with the column vector D. When I is greater than 1, a diminished C matrix and a diminished D vector are used. The first I-1 columns of the C matrix are ignored, leaving a matrix with NMAX rows but only (NMAX - (I+1)) columns. Similarly, the first I-1 elements of the D vector are ignored.

Now, the Kth element of the Ith A vector is the scalar product of the Kth row of the diminished C matrix and the diminished D vector.

After the A vector has been completely defined, the statement following 101 is reached and I is tested. If I equals 1, statement 111 is executed next. If I isgreater than 1, statement 112 is executed next.

At 111, the Lth element of the first B vector is set equal to the Lth element of the first A vector as required by the second of the Gram-Schmidt equations. Control then passes to statment 102.

The sequence of statements beginning at 112 and ending at 103 define the Lth element of the Ith B vector as required by the third of the Gram-Schmidt equations.

After the Ith B vector has been completely defined, the five statements following 102 define BMAGS as the norm of this B vector. Then the immediately following DØ loop is executed NMAX times to define the Ith row of the BB matrix as required by the first of the Gram-Schmidt equations.

Finally, when the BB matrix has been completely defined, the DØ loop ending at 108 is used to redefine the C matrix as the inverse (transpose) of the orthonormal BB matrix. Then control returns to SUBROUTINE RØSBRK.

5. Other Information

A. SUBROUTINE GRAM is called by SUBROUTINE RØSBRK.

B. SUBROUTINE GRAM calls no other subprograms.

2.3.3 One Variable Fibonacci Search

The subroutine MIMAX and the function FMIMAX, described in the following section, are utilized in the one variable Fibonacci search. SUBROUTINE MIMAX controls the search for the minimum and calls in FUNCTION FMIMAX at each point in the search to evaluate the function being optimized.

SUBROUTINE MIMAX (AA, BB, NF, NMIMAX, ACCUR, NFUNC, XMIMAX, YMIMAX)

1. Purpose

SUBROUTINE MIMAX utilizes a Fibonacci search technique to find the maximum or minimum of a one variable unimodal function within a defined region (AA, BB).

2. Input

*indicates an integer quantity

Name	Common Location	Source of Input	Description
AA	-	MAIN or GMIMAX	one boundary of the defined region within which search will take place
ACCUR	-	MAIN or GMIMAX	desired accuracy
BB	-	MAIN or GMIMAX	one boundary of the defined region within which search will take place
FMIMAX	-	FMIMAX	the value of the function being optimized at the current evaluation point within the search interval
ITERM	END *	FEV	integer code; non-zero value indicates a zero or undefined penalty function; zero value indicates a non-zero defined function
NF	*	MAIN or GMIMAX	code number of the function to be optimized

2. Input (Concl'd)

Name	Common Location	Source of Input	Description
NFUNC	- *	MAIN or GMIMAX	the number of values of the function to be utilized in the optimization
NMIMAX	- *	MAIN or GMIMAX	integer code; value of + 1 calls for maximizing calculation; value of -1 calls for minimization

3. Output

Name	Description
NF *	See Input
XL	the smaller of the two evaluation points within the current search interval (A, B)
XMIMAX	the optimum value of the independent variable
XR	the larger of the two evaluation points within the current search interval (A, B)
YMIMAX	the value of the function corresponding to the XMIMAX value of the independent variable

4. Numerical Procedure

The Fibonacci search technique is an optimal sequential search scheme for finding the maximum or minimum of a one variable unimodal function within a defined region (AA, BB). The code JMIMAX is set to 1 for a minimum and to 2 for a maximum. The quantity AA may be the lower bound of the region and BB the upper bound or vice versa. If AA is input as equal to BB, an error of type 1 exists and control returns to calling program. The initial interval R is defined from the upper bound, B, and the lower bound, A, as R = B-A in statement 23.

Either the number of function evaluations, NFUNC, to be made during the search or an end of search accuracy limit ACCUR, defined in terms of a number of independent variable units away from the actual maximum or minimum has to be given. If NFUNC is provided, control passes to statement 35 where the appropriate trial Fibonacci number, RØACC, is defined before passing to 45. If instead ACCUR has been given, then NFUNC is zero, control passes to statement 40 where RØACC, the trial Fibonacci number, is defined as R/ACC before control passes to 45. If NFUNC is negative and ACCUR is negative or zero, an error of type 2 exists and control returns to the calling program.

At statement 45, if RØACC is less than or equal to 2, control passes to 50 where XMIMAX is set equal to the midpoint of the interval (A, B) and YMIMAX evaluated for this value before the return to the calling program. If RØACC is greater than 2, control passes to statement 51, where it is further directed to either statement 60 if RØACC $\leq$ 3 or statement 61 if RØACC > 3.

Statement 60 begins a sequence of calculations where two evaluation points within the interval (A, B) are redefined. The left hand value of the independent variable, XL, is located one-third of the distance between A and B; the right hand value, XR, is located two-thirds of the distance between A and B. The Function FMIMAX is utilized to define YL, the function being optimized evaluated at point XL. If ITERM is non-zero, control then passes to 160; otherwise, FMIMAX is used to define YR at point XR. If ITERM now has a non-zero value, control passes to statement 150. If not, the difference, YL-YR, is tested. If this difference is zero or negative, XMIMAX is set equal to XR and YMIMAX to YR before the return to the calling subroutine. If this difference is positive, XMIMAX is set equal to XL and YMIMAX to YL before the RETURN is executed.

Statement 61 begins the calculation of the actual Fibonacci number which is obtained through an iteration process where

$$E_1 = 1$$

$$E_2 = 1$$

$$E_K = E_{K-1} + E_{K-2} \qquad K = 3, 4, 5 \ldots .$$

The iteration process is continued until the first K is found where $E_K \geq$ RØACC or until K reaches the maximum of 40. If K reaches 40 before $E_K \geq$ RØACC, then control passes to 76 where the message "ERROR OF TYPE 3" is printed out before the return to the calling program. Otherwise, N is set equal to the value of K for which E_K first becomes greater than or equal to RØACC in statement 80. Then the two evaluation points in the first interval are located as follows

4. Numerical Procedure

The Fibonacci search technique is an optimal sequential search scheme for finding the maximum or minimum of a one variable unimodal function within a defined region (AA, BB). The code JMIMAX is set to 1 for a minimum and to 2 for a maximum. The quantity AA may be the lower bound of the region and BB the upper bound or vice versa. If AA is input as equal to BB, an error of type 1 exists and control returns to calling program. The initial interval R is defined from the upper bound, B, and the lower bound, A, as R = B-A in statement 23.

Either the number of function evaluations, NFUNC, to be made during the search or an end of search accuracy limit ACCUR, defined in terms of a number of independent variable units away from the actual maximum or minimum has to be given. If NFUNC is provided, control passes to statement 35 where the appropriate trial Fibonacci number, RØACC, is defined before passing to 45. If instead ACCUR has been given, then NFUNC is zero, control passes to statement 40 where RØACC, the trial Fibonacci number, is defined as R/ACC before control passes to 45. If NFUNC is negative and ACCUR is negative or zero, an error of type 2 exists and control returns to the calling program.

At statement 45, if RØACC is less than or equal to 2, control passes to 50 where XMIMAX is set equal to the midpoint of the interval (A, B) and YMIMAX evaluated for this value before the return to the calling program. If RØACC is greater than 2, control passes to statement 51, where it is further directed to either statement 60 if RØACC $\leq$ 3 or statement 61 if RØACC > 3.

Statement 60 begins a sequence of calculations where two evaluation points within the interval (A, B) are redefined. The left hand value of the independent variable, XL, is located one-third of the distance between A and B; the right hand value, XR, is located two-thirds of the distance between A and B. The Function FMIMAX is utilized to define YL, the function being optimized evaluated at point XL. If ITERM is non-zero, control then passes to 160; otherwise, FMIMAX is used to define YR at point XR. If ITERM now has a non-zero value, control passes to statement 150. If not, the difference, YL-YR, is tested. If this difference is zero or negative, XMIMAX is set equal to XR and YMIMAX to YR before the return to the calling subroutine. If this difference is positive, XMIMAX is set equal to XL and YMIMAX to YL before the RETURN is executed.

Statement 61 begins the calculation of the actual Fibonacci number which is obtained through an iteration process where

$$E_1 = 1$$

$$E_2 = 1$$

$$E_K = E_{K-1} + E_{K-2} \qquad K = 3, 4, 5 \ldots .$$

The iteration process is continued until the first K is found where $E_K \geq$ RØACC or until K reaches the maximum of 40. If K reaches 40 before $E_K \geq$ RØACC, then control passes to 76 where the message "ERROR OF TYPE 3" is printed out before the return to the calling program. Otherwise, N is set equal to the value of K for which E_K first becomes greater than or equal to RØACC in statement 80. Then the two evaluation points in the first interval are located as follows

$$XL = A + \frac{E_{N-2}}{E_N} R$$

$$XR = A + \frac{E_{N-1}}{E_N} R$$

The corresponding functional value XL is then computed using FMIMAX. If ITERM $\neq$ 0, control passes to statement 160; if ITERM = 0, the functional value YR is determined from FMIMAX. The code ITERM is again tested and control sent to statement 150 if ITERM $\neq$ 0. If ITERM =0, calculations proceed and the index J is set equal to 1. Statement 90 begins the definitions of the following indices:

NJ = N-J

NJØNE = N-J-1

NJTWØ = N-J-2

The code JMIMAX is tested and control passes to statement 100 for a minimum search or to statement 101 for a maximum search.

Statement 100 tests the difference YR-YL and sends control either to statement 130 if the difference is zero or negative or to statement 110 if the difference is positive. Statement 100 performs the same test but sends control to statement 110 for a zero or negative difference or to 130 for a positive difference. Statements 110 and 130 both test the quantity J - N + 3, where N-2 is the total number of function evaluations to be made. Statement 110 sends control either to statement 160 for a positive or zero value or to statement 120 for a negative value. Likewise, statement 130 sends control to statement 140 for a negative value or to statement 150 for a zero or positive value.

Statement 120 defines the point XR to be the new end point of B of the search interval. The new interval R = B-A is defined, then the new XR is set equal to XL before a new value of XL is defined from the relation

$$XL = A + \frac{E(NJTW\emptyset)\, R}{E(NJ)}$$

The new corresponding functional values are then determined: YR is set equal to YL, then the new YL is defined by employing FUNCTION FMIMAX. The code ITERM is tested and control sent to statement 160 for a non-zero value or to statement 125 for a zero value. Statement 125 increases the index J by 1 and sends control back to statement 90 and the process of recalculating the interval and the evaluation point continues until the appropriate number of function evaluations have been made; i.e., J-N+3 reaches zero.

A similar process is employed following statement 140. The interval bound A is redefined as XL, a new R = B - A is calculated, the new XL is defined as XR, XR is redefined as

$$XR = A + \frac{E(NJ\emptyset NE)\, R}{E(NJ)} ,$$

YL is set equal to YR, and FMIMAX is called to compute the new functional value YR. Code ITERM is tested and a zero value sends control to statement 125. A non-zero value of ITERM sends control to statement 150.

Statement 160 defines the optimum value of the independent variable, XMIMAX, to be XL and the corresponding functional value YMIMAX to be YL before the return to the calling program.

Statement 150 defines the optimum value of the independent variable, XMIMAX, to be XR and the corresponding functional value YMIMAX to be YR before returning to the calling program.

5. Other Information

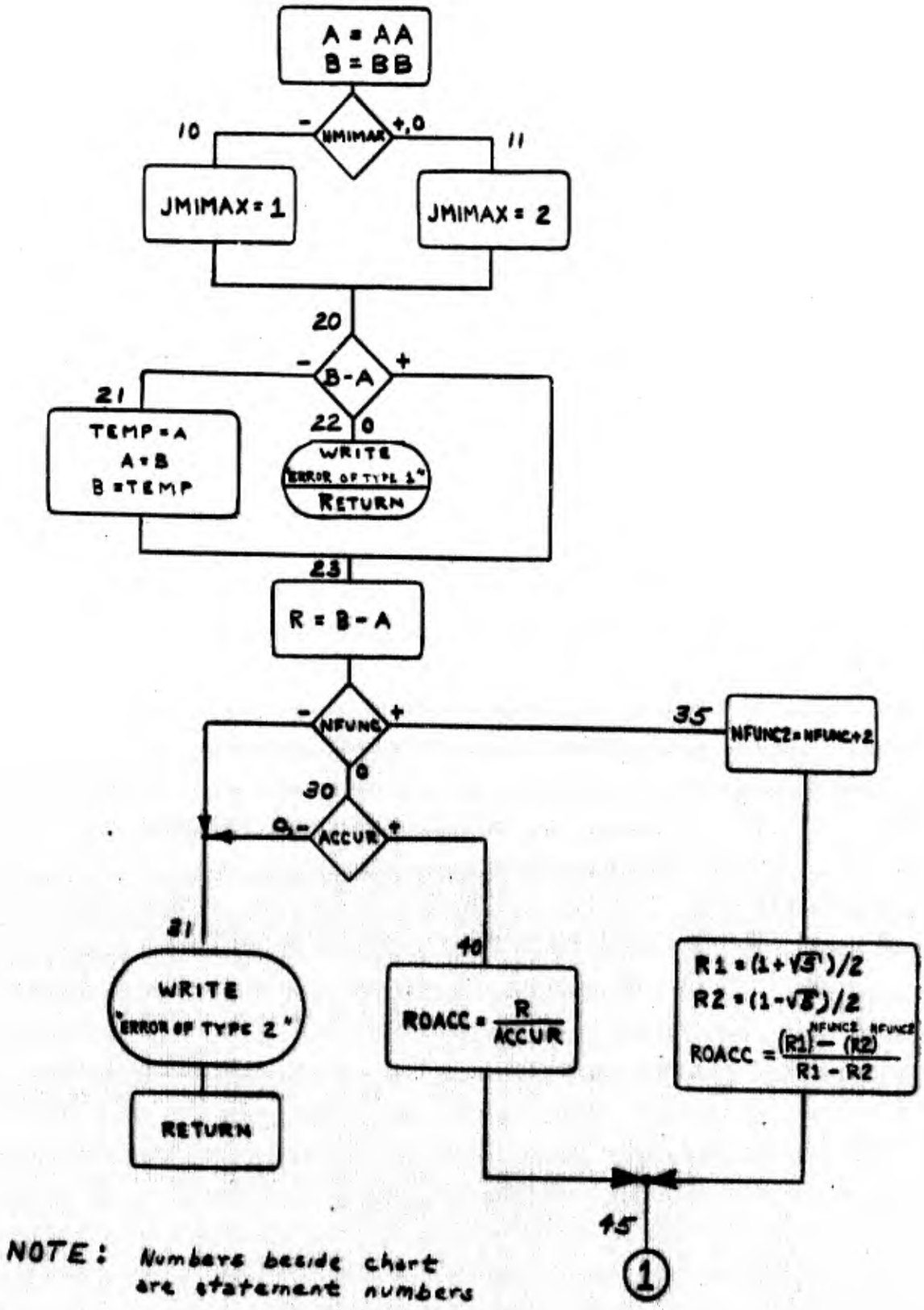
A. SUBROUTINE MIMAX is called by either MAIN or by FUNCTION GMIMAX.

B. SUBROUTINE MIMAX calls FUNCTION FMIMAX.

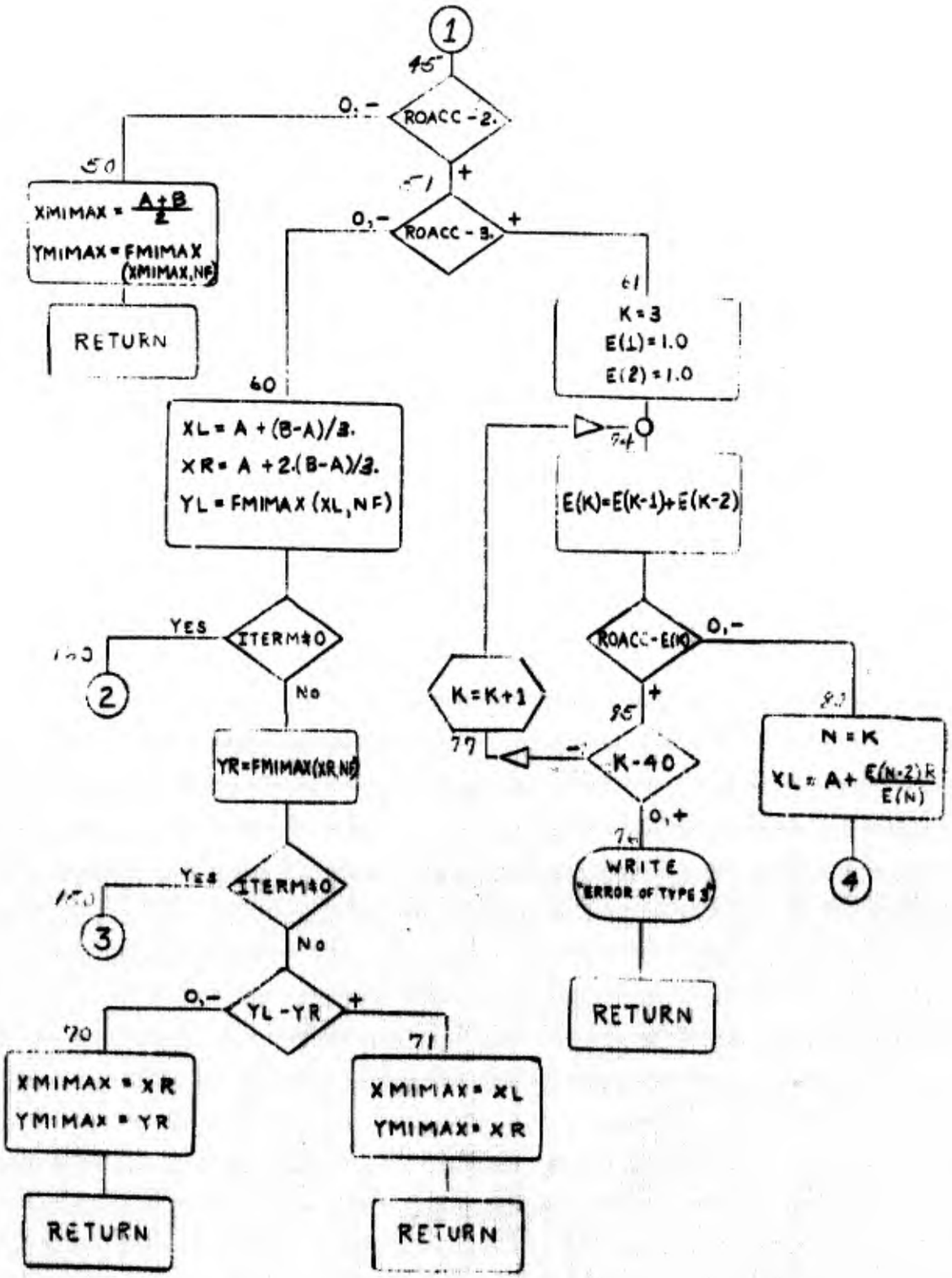
C. The flow chart on the following pages illustrates the logic of the search technique used in this subroutine.

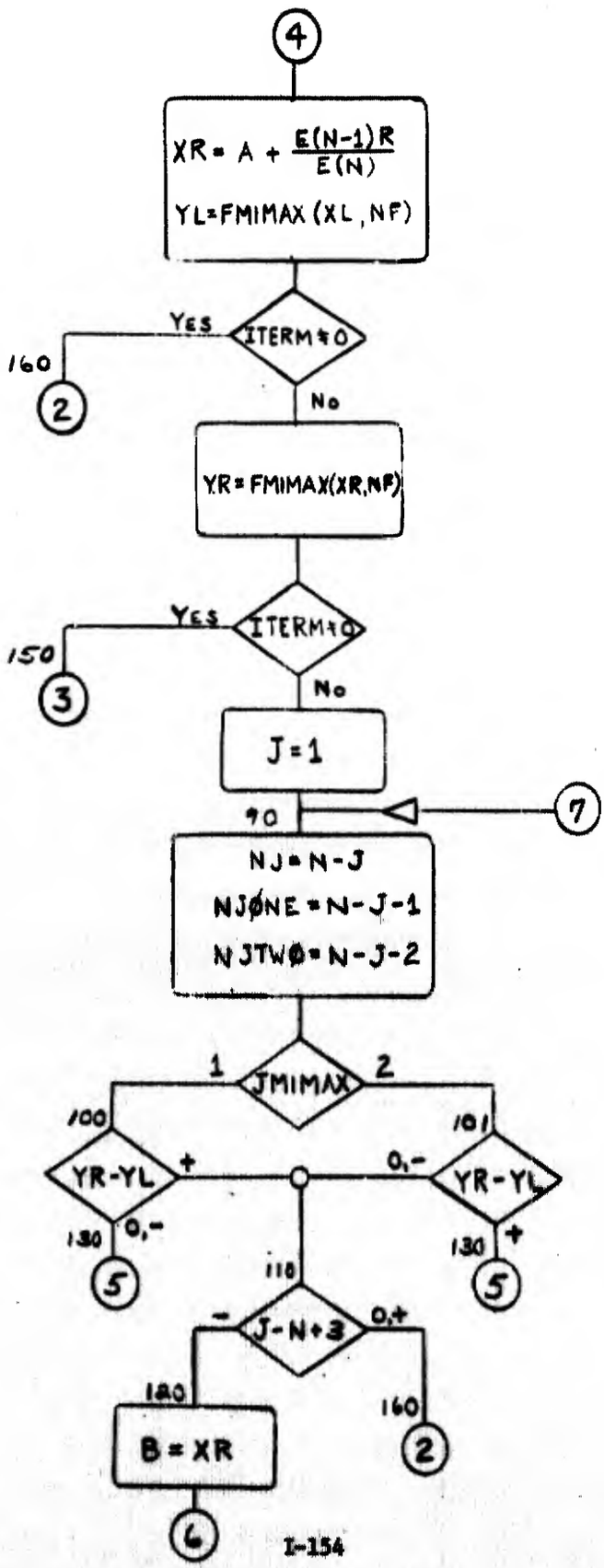
D. The error messages have the following meaning:

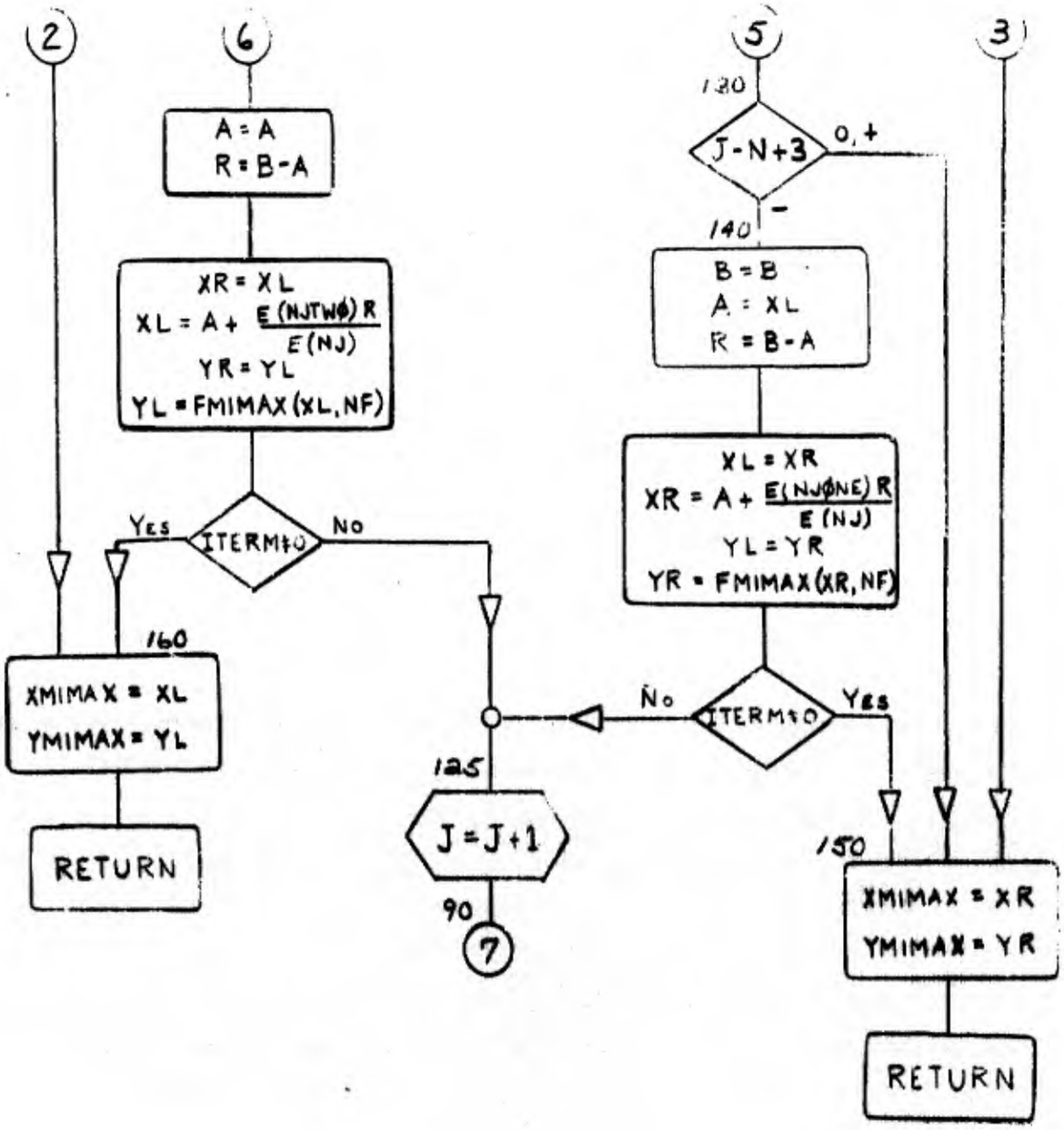
ERROR OF TYPE 1	The end points of the interval, AA and BB, are at the same location
ERROR OF TYPE 2	If NFUNC $\leq$ 0.0 and ACCUR $\leq$ 0.0, the routine has no way of defining the accuracy requirement.
ERROR OF TYPE 3	The accuracy requirement cannot be satisfied.



NOTE: Numbers beside chart are statement numbers


1
45
ROACC - 2.
0, -
+
50
XMIMAX = A+B/2
YMIMAX = FMIMAX (XMIMAX, NF)
RETURN
51
ROACC - 3.
0, -
+
60
XL = A + (B-A)/3.
XR = A + 2.(B-A)/3.
YL = FMIMAX (XL, NF)
61
K = 3
E(1) = 1.0
E(2) = 1.0
64
E(K)=E(K-1)+E(K-2)
ROACC-E(K)
0, -
+
K=K+1
77
K-40
-
85
0, +
70
WRITE "ERROR OF TYPE 3"
RETURN
80
N = K
XL = A + E(N-2)/E(N) R
4
YES
ITERM≠0
No
2
YR=FMIMAX(XR,NF)
YES
ITERM≠0
No
3
0, -
YL - YR
+
70
XMIMAX = XR
YMIMAX = YR
RETURN
71
XMIMAX = XL
YMIMAX = XR
RETURN


4
XR = A + E(N-1)R / E(N)
YL=FMIMAX (XL , NF)
YES
ITERM≠0
160
2
No
YR = FMIMAX(XR,NF)
YES
ITERM≠0
150
3
No
J=1
90
7
NJ = N-J
NJØNE = N-J-1
NJTWØ = N-J-2
1
JMIMAX
2
100
101
YR-YL
+
0,-
YR-YL
130
0,-
5
130
+
5
110
-
J-N+3
0,+
180
160
B = XR
2
6


2
6
5
3
120
J-N+3
0, +
-
140
A = A
R = B-A
XR = XL
XL = A + E(NJTW0) R / E(NJ)
YR = YL
YL = FMIMAX (XL, NF)
B = B
A = XL
R = B-A
XL = XR
XR = A + E(NJ0NE) R / E(NJ)
YL = YR
YR = FMIMAX (XR, NF)
Yes
ITERM+0
No
No
ITERM+0
Yes
160
XMIMAX = XL
YMIMAX = YL
RETURN
125
J = J+1
90
7
150
XMIMAX = XR
YMIMAX = YR
RETURN

FUNCTION FMIMAX (X, NF)

1. Purpose

In the Fibonacci optimization technique, FUNCTION FMIMAX is set equal in turn to each of the N unimodal functions of one variable being minimized (maximized) in accordance with the value of NF.

2. Input

*indicates integer quantity

Name	Source of Input	Description
NF*	MIMAX	the number code indicating the function to be evaluated
X	MIMAX	the abscissa for which FMIMAX is evaluated
VAL	FEV	the value of the function being evaluated; argument of FEV

3. Output

Name	Description
FMIMAX	the value of the function being evaluated by FEV
X	the abscissa for the function which FEV is evaluating when NF = 1 or 2
XX, 20	the abscissa for the function which FEV is evaluating when NF = 3

4. Numerical Procedure

Upon entry into FUNCTION FMIMAX, control is directed according to the value of NF to statements 1 (NF = 1 or 2) or 3(NF = 3). Statement 1 calls SUBROUTINE FEV as a function of X, then FMIMAX is set equal to the resulting value of VAL before the return to MIMAX is executed. Statement 3, defines XX(1) as having the value of X. This parameter is then used as an argument in the calling of FEV which yields the quantity VAL from which FMIMAX is defined before control returns to MIMAX.

5. Other Information

A. FUNCTION FMIMAX is called by SUBROUTINE MIMAX.

B. FUNCTION FMIMAX calls in SUBROUTINE FEV.

2.3.4 Two Variable Fibonacci Search

The two variable Fibonacci Search involves the following subroutines - GIMAX, GMIMAX, MIMAX, and FMIMAX.

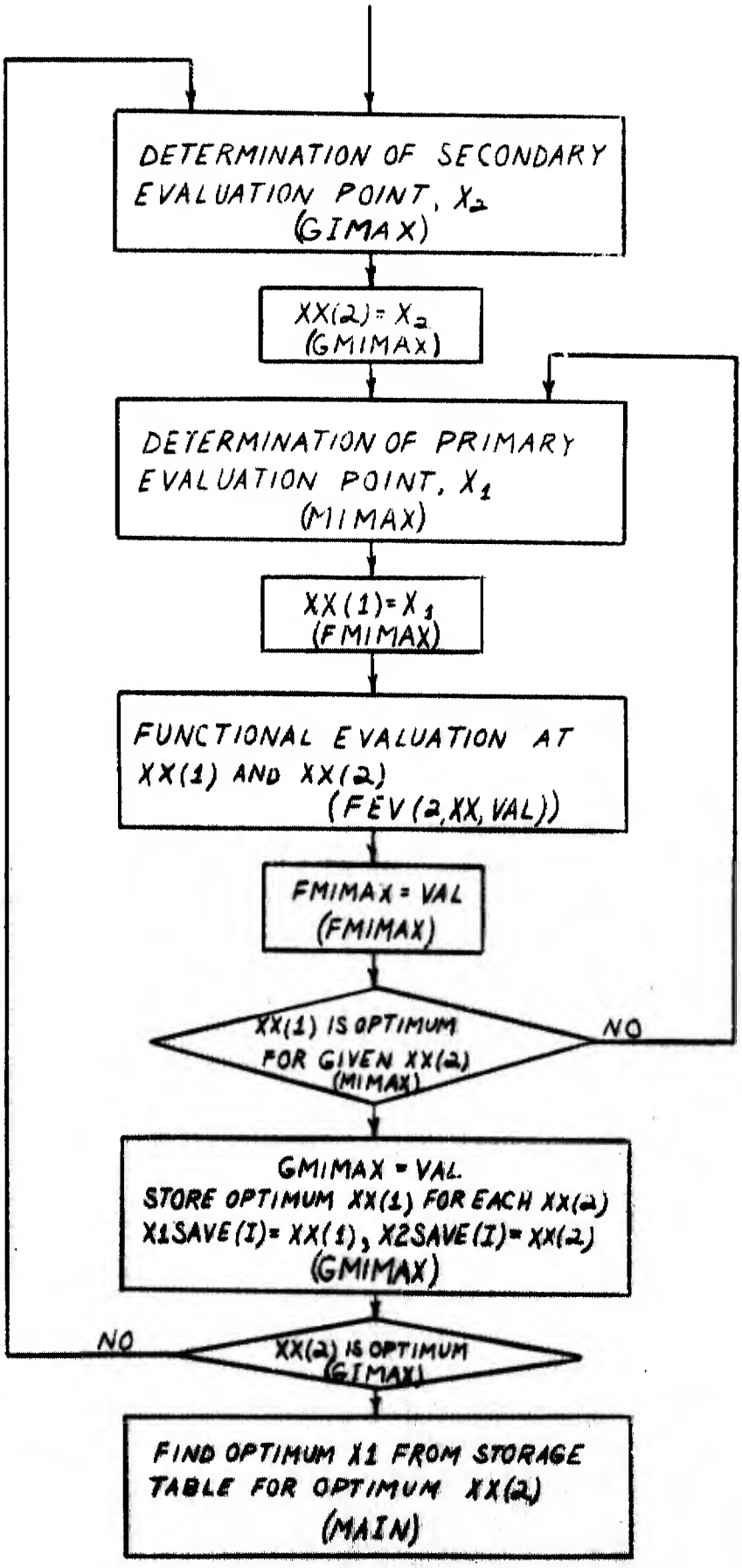
SUBROUTINE GIMAX optimizes the secondary independent variable of the penalty function. At each evaluation point of the secondary independent variable in this search, FUNCTION GMIMAX is called in to evaluate the function for the optimum value of the primary independent variable. This is accomplished by calling SUBROUTINE MIMAX.

The following section contains the descriptions of SUBROUTINE GIMAX and FUNCTION GMIMAX. SUBROUTINE MIMAX and FUNCTION FMIMAX appear in the preceding section.

SUBROUTINE GIMAX (AA, BB, NF, NMIMAX, ACCUR, NFUNC, XMIMAX, YMIMAX)

1. Purpose

SUBROUTINE GIMAX, the controlling subprogram for the two variable Fibonacci search technique, optimizes the secondary independent variable of the function. At each evaluation point in this optimization, GIMAX summons FUNCTION GMIMAX which calls for the optimization of the primary variable. Then, in GMIMAX, the secondary independent variable value and the optimum primary value are stored in a table. This process is repeated until the selected number of evaluations have been made on the secondary independent variable. In the main program, the table is then searched for the optimum secondary value producing also the associated primary value. The procedure is illustrated in the following diagram (where the subroutine or function where step is taking place is found in parentheses):


DETERMINATION OF SECONDARY EVALUATION POINT, X_2 (GIMAX)
$XX(2) = X_2$ (GMIMAX)
DETERMINATION OF PRIMARY EVALUATION POINT, X_1 (MIMAX)
$XX(1) = X_1$ (FMIMAX)
FUNCTIONAL EVALUATION AT XX(1) AND XX(2) (FEV(2,XX,VAL))
FMIMAX = VAL (FMIMAX)
XX(1) IS OPTIMUM FOR GIVEN XX(2) (MIMAX)
NO
GMIMAX = VAL
STORE OPTIMUM XX(1) FOR EACH XX(2)
X1SAVE(I) = XX(1), X2SAVE(I) = XX(2)
(GMIMAX)
NO
XX(2) IS OPTIMUM (GIMAX)
FIND OPTIMUM X1 FROM STORAGE TABLE FOR OPTIMUM XX(2)
(MAIN)

2. Input

*indicates integer quantity

Name	Common Location	Source of Input	Description
AA	-	MAIN	one boundary of the defined region within which search will take place
ACCUR	-	MAIN	desired accuracy
BB	-	MAIN	one boundary of the defined region within which search will take place
GMIMAX	-	FUNCTION GMIMAX	the evaluation of the penalty function at the optimum value of primary variable for the specified secondary variable value
ITERM	END *	FEV	integer code; non-zero value indicates a zero or undefined penalty function; zero value indicates a non-zero defined function
NF	- *	MAIN	code number of the function to be optimized
NFUNC	- *	MAIN	the number of values of the function to be utilized in the optimization
NMIMAX	- *	MAIN	integer code; value of +1 calls for maximizing calculation; value of -1 calls for minimization

3. Output

Name	Description
NF *	See Input
XL	the smaller of the two evaluation points withing the current search interval (A, B)
XMIMAX	optimum value of the secondary independent variable
XR	the larger of the two evaluation points within the current search interval (A, B)
YMIMAX	function value corresponding to the XMIMAX value of the secondary independent variable and the optimum value of primary independent variable.

4. Numerical Procedure

SUBROUTINE GIMAX is a duplicate of SUBROUTINE MIMAX with the single exception that, where MIMAX calls in FUNCTION FMIMAX, GIMAX calls in FUNCTION GMIMAX. This takes place a 7 locations within the subroutine: (a) once following each of the statements 50, 120, and 140 and (b) twice following each of statements 60 and 80.

See SUBROUTINE MIMAX description for details.

5. Other Information

A. SUBROUTINE GIMAX is called by the MAIN program only.

B. SUBROUTINE GIMAX calls FUNCTION GMIMAX.

C. SUBROUTINE GIMAX calls the IBM supplied functions DSQRT and FDXPI (exponentiation).

FUNCTION GMIMAX (X, NF)

1. Purpose

FUNCTION GMIMAX, part of the two variable Fibonacci search technique, optimizes the primary independent variable for each step in the optimization of the secondary variable, i.e., for each evaluation point selected during second variable optimization.

2. Input

*indicates an integer quantity

Name	Common Block Name	Source of Input	Description
ALB, 20	MINSK	MAIN	lower bound of the range of the primary independent variable within which the minimum (maximum) is to be found
ERR	FØPT	ZREADX	desired accuracy
KCØUNT	XXSAVE *	MAIN, GMIMAX	integer counter, number of times GMIMAX is used; used as an index for X1SAVE and X2SAVE.
LIMIT	IØPT*	ZREADX	the number of functional values to be utilized by the program
NF	*	GIMAX	the code which indicates the function GMIMAX is evaluating
UB, 20	MINSK	MAIN	the upper bound of the range of the primary independent variable within which the minimum (maximum) is to be found

2. Input (Concl'd)

Name	Common Block Name	Source of Input	Description
VAL		MIMAX	the value of the function at the minimum (maximum) of the primary independent variable
X		GIMAX	the current evaluation point of the secondary independent variable, an argument of GMIMAX
XX(1)	MINSK	MIMAX	the value at the extremum of the primary independent variable

3. Output

*indicates integer quantity

Name	Common Block Name	Description
ALB, 20	MINSK	See Input
ERR	FØPT	See Input
GMIMAX	-	the value of the function obtained from MIMAX, same as VAL
ITERM	END *	integer code which indicates if function is defined and non-zero, undefined, or zero
KCØUNT	XXSAVE*	See Input
LIMIT	IØPT*	See Input
UB, 20	MINSK	See Input

3. Output (Concl'd)

Name	Common Block Name	
X1SAVE, 40	XXSAVE	the value of XX(1) for the KCØUNT th pass through GMIMAX; the optimum value of the primary independent variable for each secondary evaluation point
X2SAVE, 40	XXSAVE	the stored value of the secondary evaluation point
XX, 20	MINSK	XX(1) - see input; XX(2) is the evaluation point of the second independent variable

4. Numerical Procedure

Function GMIMAX equates XX(2) to X before calling SUBROUTINE MIMAX for the limits ALB(1) and UB(1) to determine the optimum value of the dependent variable, VAL, and the corresponding value of the independent variable XX(1). GMIMAX is then equated to VAL, the integer KCØUNT is increased by one and tested. If KCØUNT is greater than 40, ITERM is set equal to 1 and control returns to GIMAX. If KCØUNT is less than or equal to 40, it is used as the index of the quantities X1SAVE and X2SAVE which are respectively set equal to XX(1) and XX(2) before control passes to GIMAX.

5. Other Information

A. FUNCTION GMIMAX is called by SUBROUTINE GIMAX.

B. FUNCTION GMIMAX calls in SUBROUTINE MIMAX.

3.0 BASIC ANALYSIS CALCULATIONS

Basic Analysis Calculations

The following sections deal with the basic analysis calculations which are performed for each vehicle, whether reentry vehicle or decoy. These consist of the determination of the trajectory after consideration of mass loss, noseblunting, and angle of attack effects and the determination of the radar cross section. The basic analysis calculations are called by SUBROUTINE F123 for the reentry vehicle and for each decoy.

3.1 Trajectory Calculations

The following subsections describe the computations which determine the vehicle trajectory. These are the initializing of trajectory and vehicle geometric parameters, the calculation of the derivatives of trajectory and vehicle geometric parameters, and the predictor - corrector integration of these derivatives, and the printout operations.

3.1.1 Initializing and Printout Operations

SUBROUTINE VIXEN sets the initial values of the trajectory variables then calls CHNTBL to assign the initial values of the vehicle geometric variables and the material properties of the heatshield for either input or built-in material. Then VIXEN either defines the input trajectory or wind tunnel conditions and calls the appropriate subroutines to calculate drag coefficient or calls in the predictor-corrector integration routine, ADM4RK which in turn calls the subroutines which calculate the derivatives. Then the detailed printout of trajectory related quantities is performed at the designated print interval. After the trajectory calculation is completed SUBROUTINE RITOUT is called to print a summary of the maximum and minimums at printout altitudes of angle of attack.

SUBROUTINE VIXEN

1. Purpose

SUBROUTINE VIXEN controls either directly or indirectly the calling of all the subroutines involved in the trajectory, mass loss, shape change, and drag coefficient calculations. VIXEN performs the following functions:

A. Defines plot titles

B. Sets initial values of variables being integrated

C. Assigns the appropriate initial geometry

D. Defines the Δ of integration and the accuracy limits on integrated variables

E. Controls the calculations for the tabular input trajectory and input wind tunnel conditions options

F. Calls in the integration subroutine which controls the evaluation of calculated trajectory

G. Tests for maximums and minimums in angle of attack at printout altitudes

H. Stores parameters at printout altitudes for the tape and plotting options.

I. Controls the detailed printout of trajectory information

2. Input

NOTE: 1. quantities in Equivalence column are ØCCUR locations unless otherwise designated

2. *- indicates integer quantity and, unless otherwise designated, an NØCCUR location

Name	Symbol	Common Location	Source of Input	
A, 514	A_i	301 - 814	ZPRS or READIT	cur
ALPHA	α	002	ROTATE, DEREQ	angl
ALPRIM	α'	003	ROTATE, DEREQ	angl
ALPTAB, 75	α(Table)	3646-3720	READIT	inpu
ALST	$\alpha_{STØP}$	122	F123	stop
AREF	A_{ref}	001	PRELIM	refe
CAPL	L	010	PRELIM	sha
CASE	-	128	READIT	case
CD	C_D	016	DEREQ or DRAGCØ	tota
CDB	C_{D_B}	099	DRAGCØ	base
CDFINF, 8	$C_{D_{f_\infty}}$	2793-2800	DRAGCØ	skin
CDI	C_{D_I}	100	DRAGCØ	tota
CDØWN, 16	-	3549-3564	SR2490 or READIT	lowe
CDP	C_{D_P}	098	DRAGCØ	pres
CDP0	$C_{D_{P_0}}$	101	DRAGCØ	zero
CHIGH, 16	-	3533-3548	SR2490 or READIT	uppe
CM	C_m	202	PRELIM	mon
CMQ	C_{m_q}	020	READIT or PRELIM	dam

unless

ignated,

nput	Description	Units
EADIT	curve fit coefficients	-
DEREQ	angle of attack	radians
DEREQ	angle of attack	radians
	input tabular angle of attack	degrees
	stop control on envelope α	radians
	reference area	ft^2
	sharp cone slant length	ft.
	case number	-
DRAGCØ	total drag coefficient	-
	base drag coefficient	-
	skin friction drag coefficient	-
	total induced drag coefficient	-
READIT	lower limit on variable accuracy	-
	pressure drag coefficient corrected for α effects	-
	zero angle of attack pressure drag coefficient	-
READIT	upper limit on variable accuracy	-
	moment coefficient	-
PRELIM	damping in pitch	-

B

2. Input (Cont'd)

Name	Symbol	Common Location	Source of Input	
CN	C_n	203	PRELIM	normal
DATE	-	127	READIT	date
DECDFP	$(\Delta C_{D_{f_\infty}})_P$	235	DRAGCØ	pressur
DECFTC	$(\Delta C_{D_{f_\infty}})_{TC}$	236	DRAGCØ	transve
DEL	-	-	ADM4RK	delta of
DELCDP	ΔC_{D_P}	234	DRAGCØ	induced
DELIN	-	187	SR2490 or READIT	largest
DELW	ΔW_{TOTAL}	097	PRELIM	total cha
DELW2	ΔW_2	226	PRELIM	change i
DELW3	ΔW_3	227	PRELIM	change i
DERIV, 16	-	-	ADM4RK	derivati
DNBNDZ	-	248	READIT	lowest a
DVALUE, 16	-	-	ADM4RK	integrat
EMØ		129	READIT	memo n
G	g	027	SR2490 or READIT	factor t
GAMF0	γ_{f_0}	105	F123	input in
HSRT0	h_s/RT_o	029	PRELIM	non-dim
IATMØS	-	08 *	READIT	option c
IDBL	-	IØCCUR (314)*	SR2490 or ZREADX	code co of 3 ind preset
INALPH	-	30 *	READIT	option f
IØP, 90	-	IØCCUR(1-90)*	ZREADX or SR2490	control options

ut	Description	Units
	normal force coefficient	-
	date	-
	pressure induced skin friction drag coefficient	-
	transverse curvature induced skin friction drag coefficient	
	delta of integration in ADM4RK	ft.
	induced pressure drag coefficient	
EADIT	largest value that DEL is allowed to have; preset to -2000.	ft.
	total change in weight = $\Delta W_2 + \Delta W_3$	lb.
	change in weight due to ablative effects alone	lb.
	change in weight due to thrusting effects alone	lb.
	derivatives to be integrated by ADM4RK	-
	lowest altitude for which read in atmosphere is used	ft.
	integrated values of variables	
	memo number	
EADIT	factor to convert slugs to lbs. mass, 32.174	lbm/slug
	input initial flight path angle	radians
	non-dimensionalized stagnation enthalpy	-
	option code for input atmosphere	-
READX	code controls units of radar cross section; a value of 3 indicates decibels, 4 indicates square meters; preset to 3.	
	option for input angle of attack with calculated trajectory	-
SR2490	control codes; used specifically for wake plotting options in VIXEN	-

B

2. Input (Cont'd)

Name	Symbol	Common Location	Source of Input	
ITAPE	-	29*	READIT	option for
JJHØLD	-	01 *	CHNTBL	error cod
LA	La	033	PRELIM or CHNTBL	axial leng
LAMDA		032	PRELIM or CHNTBL	bluntness
LL	-	- *	PRELIM	error cod
LØPT	_	07 *	SR2490 or READIT	trajectory
LP	-	ØCCUR(4000) *	ADM4RK	error cod
MAXVAL	-	06*	READIT	maximum or wind tu
MDØT, 32	$\dot{m}_i$	2708-2739	EVIL	mass loss
MHEAT	-	10 *	READIT	input cod
MINF	M_∞	035	PRELIM	free strea
MØPT	-	03 *	READIT	input cod
NPLØT, 5	-	24 - 28 *	READIT	input cod
NPRINT	-	14*	READIT	input cod
P0	P_0	109	READIT	input initi
PEPSB	$(P_e/P_s)_i$	2801-2808	AERØDY	distributi dimension
PHI0	Φ_0	112	F123	input initi
PI	π	042	SR2490	mathemat
PS	P_s	047	PRELIM	stagnation
PSI0	Ψ_0	114	F123	input initi
PSIALP	Ψ_α	200	DEREQ	thrust off
Q0	Q_0	110	READIT	input initi
QD	Q_D	051	PRELIM	dynamic

	Description	Units
	option for V, γ, Z tape storage	
	error code in CHNTBL	
L	axial length of vehicle	ft.
L	bluntness ratio, Rn/Rb	-
	error code in PRELIM	0
	trajectory option code	
	error code in ADM4RK	
	maximum number tabular values for input trajectory or wind tunnel conditions options	
	mass loss rate along vehicle	$lbm/ft^2 sec$
	input code controlling mass loss calculations	-
	free stream Mach Number	-
	input code controlling mass loss calculations	-
	input code controlling plots	-
	input code controlling detailed trajectory printout	-
	input initial angular velocity P	rad/sec
	distribution along body of edge pressure non-dimensionalized by stagnation pressure	-
	input initial Euler angle, Φ_o	radians
	mathematical constant	-
	stagnation pressure	lb/ft^2
	input initial Euler angle, Ψ_o	radians
	thrust offset angle	radians
	input initial angular velocity, Q_o	rad/sec
	dynamic pressure	lb/ft^2

B

2. Input (Cont'd)

Name	Symbol	Common Location	Source of Input	
QDØT, 32	$\dot{q}_i$	2676-2707	AERØDY	distributio
REYL	$R_{e_{\infty_L}}$	062	PRELIM	free strea cone slant
RN	Rn	052	CHNTBL	nose radiu
SMF	f	081	RØTATE	frequency
SMR0	R_o	111	READIT	input initia
T0	T_o	102	READIT	input initia
TH	T_h	201	PRELIM	total thrus
THEAL0	θ_{α_o}	113	F123	input initia
THETA	θ	076	READIT or CHNTBL	cone half a
THETAD	θ_D	069	CHNTBL	cone half a
TRAJRN, 75	R_n(table)	1644-1718	READIT	input traje
TRAJT, 75	t (table)	1344-1418	READIT	input traje
TRAJV, 75	V (table	1494-1568	READIT	input traje
TRAJW, 75	W_{TOTAL}(table)	1569 - 1643	READIT	input traje
TRAJZ, 75	Z (table)	1419 - 1493	READIT	input traje
TRJALP, 75	α(table)	1719 - 1793	READIT	input traje
TST	t_{STOP}	123	SR2490 or READIT	stopping c
TTMAT, 3		3565-3567	TEQUAT	component coordiate
UPBNDZ	-	247	READIT	upper altit
V0	V_o	106	READIT	input initia
W1	W_1	133	READIT	input initia
WTMINF, 75	M (table)	1119-1193	READIT	input table
WTØTAL	$W_{TØTAL}$	228	PRELIM	total vehic

	Description	Units
	distribution along body of aerodynamic heating rates	btu/ft^2-sec
	free stream Reynolds number based on sharp cone slant length	-
	nose radius	ft.
	frequency of cycle in α	cycles/sec
	input initial angular velocity R	rad/sec
	input initial time	
	total thrusting force	lb.
	input initial Euler angle, θ_{α_o}	radius
NTBL	cone half angle in radians	radians
	cone half angle in degrees	degrees
	input trajectory values for nose radius	in.
	input trajectory values for time	sec.
	input trajectory values for velocity	ft/sec
	input trajectory values for total weight	lb/
	input trajectory values for altitude	ft.
	input trajectory values for α	degrees
DIT	stopping control on time	sec.
	components of thrust vector in trajectory coordiate system	lb.
	upper altitude limit on use of input atmosphere	ft.
	input initial velocity	ft/sec
	input initial weight for first configuration in a case	lb.
	input table of Mach numbers for wind tunnel option	-
	total vehicle weight	lb.

B

2. Input (Concl'd)

Name	Symbol	Common Location	Source of Input	
WTPTØT, 75	P_{TOTAL}(table)	1269-1343	READIT	input tab
WTRINF, 75	R_{e_∞}/in (table)	1194-1268	READIT	input tab inch for
WTZ, 75	Z(table)	1044-1118	READIT	input tab
XBAR	$\bar{\chi}$	090	PRELIM	interacti
XBAR1	$\bar{\chi}_1$	126	PRELIM	rarefact
XR0	X_{r_o}	107	READIT	input ini
XUP	χ_{UP}	237	SR2490 or READIT	value of
Z0	z_o	108	READIT	input ini
ZBAR	$\bar{z}$	120	SR2490 or READIT	altitude
ZPR1		118	SR2490 or READIT	initial pr
ZPR2		119	READIT	second p
ZST		121	READIT	altitude
ZTR	z_{tr}	092	PRELIM	transitio
ZTURNX	z_{turn}	-	CHNTBL	altitude discontin per case

ut	Description	Units
	input table of total pressure for wind tunnel option	lb/ft^2
	input table of free stream Reynolds number per inch for wind tunnel option	1/in
	input table of altitudes for wind tunnel option	ft.
	interaction parameter	-
	rarefaction parameter	-
	input initial trajectory range	ft.
ADIT	value of $\bar{\chi}$ at the beginning of continuum flow	-
	input initial altitude	ft.
ADIT	altitude at which print increment is changed	ft.
ADIT	initial printout increment	ft.
	second printout increment	ft.
	altitude stopping control	ft.
	transition altitude	ft.
	altitude at which configuration of vehicle is changed discontinuously to the second input geometry per case	ft.

B

3. Output

NOTES: 1. quantities in Equivalence column are OCCUR location unless otherwise designated

2. * - indicates integer quantity and, unless otherwise designated, an NØCCUR location

Name	Symbol	Common Location	
ALBARP	$\bar{\alpha}'$	131	last m
ALMAX, 200	α_{max}	1044-1243	maxim
ALMIN, 200	α_{min}	1244-1443	minim
ALPENV	α_{enV}	096	envelo
ALPHA	α	002	instant
ALPLØT, 160	α	-	plottin
ALPRIM	α'	003	angle c
ALWIG2	$\bar{\alpha}'$	130	maxim
BETA	β	231	ballist
BETAP	$\Delta\beta/\Delta Z$	230	change
BETAPL, 160	β	PCCUR(641-800)	storag
CDBPLT, 160	C_{D_B}	-	plottin
CDFPLT, 160	C_{D_f}	-	plottin
CDIPLT, 160	C_{D_I}	-	plottin
CDPLØT, 160	C_D	-	plottin
DEL	-	-	delta o
DNBND, 16	-	-	see te
DØW	D/W_{TOTAL}	233	ratio o

ess

gnated,

Description	Units
last minimum in α'	radians
maximums in angle of attack	radians
minimums in angle of attack	radians
envelope value of angle of attack	radians
instantaneous angle of attack	radians
plotting storage for printout values of angle of attack	degrees
angle of attack	radians
maximum preceding last maximum in α'	radians
ballistic coefficient	-
change in ballistic coefficient per unit altitude	ft^{-1}
-800) storage for first 160 values of β for plotting	-
plotting storage for first 160 values of C_{D_B}	-
plotting storage for first 160 values of $C_{D_{f_{ac}}}$	-
plotting storage for first 160 values of C_{D_I}	-
plotting storage for first 160 values of total C_D	-
delta of integration in ADM4RK	ft.
see text	
ratio of total drag force to total weight	-

B

3. Output (Cont'd)

Name	Symbol	Common Location	
DVALUE, 16	-	-	integrate
FMAX, 200	f_{max}	2244 - 2443	values of
FMIN, 200	f_{min}	2244 - 2643	values of
FREQ	—	—	interval a see ADM
GRATE	-	095	multiplier
			a effects
HSRTPL, 160	h_s/RT_o	-	plotting s
IKMAX		11 *	total num
IKMIN	-	12*	total num
LCHNGE	-	*	control co
LL	-	*	error cod
LP	-	ØCCUR(4000)*	error cod
LPLØT	-	IØCCUR(302)*	counter fo
MDØT, 32	$\dot{m}_i$	2708-2739	mass loss
MINF	M_∞	035	free strea
NNTAPE	-	-	counter fo
ØCPLØT, 160	-	-	plotting s quantity s
PE1PLT, 160	P_{e_1}/P_s	-	plotting s point to s
PE7PLT, 160	Pe_7/P_s	-	plotting s diameter
PE8PLT, 160	Pe_8/P_s	-	plotting s maximum

Description	Units
integrated values of variables	-
values of frequency at altitudes for maximums in α	cycles/sec
values of frequency at altitudes for minimums in α	cycles/sec
interval at which ADM4RK returns to VIXEN, see ADM4RK	ft.
multiplier for $C_{D_{P_o}}$ which incorporates the integrated α effects over one cycle; see text.	-
plotting storage for first 160 printout values of h_s/RT_o	-
total number of maximums in α	-
total number of minimums in α	-
control code for CHNTBL	-
error code in PRELIM	-
error code	-
counter for plotting storage	-
mass loss rate along body	lbm/ft^2 sec
free stream Mach number	-
counter for tape storage	-
plotting storage for first 160 output values of any quantity stored in OCCUR	-
plotting storage for ratio of edge pressure at tangent point to stagnation pressure	-
plotting storage for ratio of edge pressure at maximum diameter of blunt cone	-
plotting storage for ratio edge pressure of sharp cone maximum diameter point to stagnation pressure	-

3. Output (Cont'd)

Name	Symbol	Common Location	
PLMINF, 160	M_{∞}	-	plot
PSPLØT, 160	P_s	-	plot
PTØTAL	P_{total}	125	tota
QD11PL, 160	$\dot{q}_{tang.\ pt.}$	-	plot
QD41PL, 160	$\dot{q}_{sonic}$	-	plot
QD7PLT, 160	$\dot{q}_7$	-	plott poin
QD8PLT, 160	$\dot{q}_8$	-	plott poin
QDØT, 32	$\dot{q}_i$	2676-2707	distr
QPLØT, 160	$\dot{q}_D$	-	plott
REYINF	Rey /ft	244	the tunn being
RN	Rn	052	nose
T	t_{cycle}	075	time
T1 to T28	-	-	plot
TAMAX, 200	t_{max}	1444-1643	time
TAMIN, 200	t_{min}	1644 - 1843	time
THEALP	θ_a	071	Eule
TIMER	t	080	insta
TIXM	-	-	plott
TIXT	-	-	plott
TIXZ	-	-	plott

Description	Units
plotting storage for free stream Mach number	-
plotting storage for stagnation pressure in atmospheres	atm.
total pressure	lb/ft^2
plotting storage for stagnation point heating rate	$Btu/ft^2 sec$
plotting storage for sonic point heating rate	$Btu/ft^2 sec$
plotting storage for blunt cone max. diameter point heating rate	$Btu/ft^2 sec$
plotting storage for sharp cone max. diameter point heating rate	$Btu/ft^2 sec$
distribution along body of heating rate	$Btu/ft^2 sec$
plotting storage for dynamic pressure	lb/ft^2
the Reynolds number per foot in the set of wind tunnel conditions for which drag calculations are being made	1/ft
nose radius	ft.
time for a cycle in angle of attack	sec.
plot title storage	
times at which maximums in angle of attack occur	sec.
times at which minimums in angle of attack occur	sec.
Euler angle Θ_a	radians
instantaneous trajectory time	sec.
plotting title	-
plotting title	-
plotting title	-

B

3. Output

Name	Symbol	Common Location	
TPLØT, 160	t	PCCUR(1-160)	plotting stora
UPBND, 16	-	-	see text
V	V	082	instantaneous
V0GDØT	$\dot{V}/g$	232	acceleration
V0GPLT, 160	$\dot{V}/g$	PCCUR(481-640)	plotting stora
VPLØT, 160	V	PCCUR(321-480)	plotting stora
VVAR	-	-	dummy variab
WL1P, 160	L_1(wake)	PCCUR(1281)	wake length f
WL2P, 160	L_2 (wake)	PCCUR(1441)	wake length f
WL3P, 160	L_3 (wake)	PCCUR(1601)	wake length f
WR1P, 160	R_1 (wake)	PCCUR(801)	wake radar c
WR2P, 160	R_2 (wake)	PCCUR(961)	wake radar c
WR3P, 160	R_3 (wake)	PCCUR(1121)	wake radar c
WTØTPL, 160	W_{total}	-	plotting stora printout altitu
XR	X_r	087	component of
YR	Y_r	199	side range du in Y direction
Z	Z	091	altitude
ZMAX, 200	Z_{max}	1844-2043	altitudes at w
ZMIN, 200	Z_{min}	2044-2243	altitudes at w

Description	Units
plotting storage of printout values of trajectory times	sec.
see text	-
instantaneous velocity	ft/sec
acceleration	g's
plotting storage for acceleration term	g's
plotting storage for velocity	ft/sec
dummy variable used in tape storage option	-
wake length for first frequency	meters
wake length for second frequency	meters
wake length for third frequency	meters
wake radar cross section for first frequency	square meters or decibels, see IDBL
wake radar cross section for second frequency	square meters or decibels, see IDBL
wake radar cross section for third frequency	square meters or decibels, see IDBL
plotting storage for values of total weight at printout altitudes	lb.
component of range in X direction	ft.
side range due to thrust offset; component of range in Y direction	ft.
altitude	ft.
altitudes at which maximums in angle of attack occur	ft.
altitudes at which minimums in angle ofattack occur	ft.

B

3. Output (concl'd)

Name	Symbol	Common Location	Description
ZPLØT, 160	Z	PCCUR(161-320)	plotting sto
ZST		121	altitude sto
ZUSE	Z	-	current alti

	Description	Units
1-320)	plotting storage for printout altitudes	ft.
	altitude stop control	ft.
	current altitude being used in trajectory calculations	ft.

B

4. Numerical Procedure

Initially, SUBROUTINE VIXEN defines the titles to be used in the plotting options, sets the counter NK to zero, assigns the internal program trajectory variables V, GAMF, TIME, Z, XR, P, Q, SMR, PSI, THEALP, PHI their input initial values, and zeroes YR and PSIALP. The trajectory-related quantities to be used in the integration subroutine, ADM4RK, are given their initial values, i.e., ZUSE is set equal to Z, and DVALUE locations 1 to 4 and 8 to 15 are set respectively to V, GAMF, TIME, XR, PSI, THEALP, PHI, Q, SMR, P, YR, and PSIALP. The following quantities are then zeroed: BETA BETAP, DELBET, T, SMF, ALPRIM, ALPHA, ALPENV, IKMAX, IKMIN, PRINTZ, ZSTØ, LPLØT, and NNTAPE. Then IALP is set to 1; the quantity FREQ which controls the return from SUBROUTINE ADM4RK is set equal to the negative of the initial printout increment ZPR1. The delta of integration, DEL, to be used in ADM4RK is set initially to the negative of one fifth of the initial print increment ZPR1. The value of DEL is then tested against DELIN; if DELIN is greater than or equal to DEL, DEL retains its value; however, if DELIN is less than DEL, then DEL is initially given the value of DELIN.

The SUBROUTINE VIXEN sets the control code LCHNGE = 1, used in SUBROUTINE CHNTBL, before calling in SUBROUTINE CHNTBL in statement 60 to define the geometric and material properties for the initial configuration as well as the initial values of DVALUE(5) through (7). The control code LP is set equal to 1, then the error code JJHØLD from CHNTBL is tested. If JJHØLD is not equal to -1, control passes to statement 50; if JJHØLD = - 1, the error code LP is set equal to 6 causing the

termination of trajectory calculations and an error message is printed out before control passes to statement 100. Statement 50 tests the current printout altitude, ZUSE which is the same as Z, against the output ZTURNX from CHNTBL. When the first configuration is defined in CHNTBL, ZTURNX is set equal to the input ZTURN. The test on ZUSE causes control to pass to statement 61 for altitudes greater than ZTURNX. When ZUSE becomes less than or equal to ZTURN, the integer code LCHNGE is set equal to 2 and control passes to statement 60. There CHNTBL is called to define the new configuration and to redefine ZTURNX to the value -1.0, which combined with the test in statement 50 prevents SUBROUTINE CHNTBL from being called more than two times per case.

Statement 61 begins the block of equations which define the upper and lower bounds on the absolute difference that is allowed between the extrapolated and interpolated values of the variables being integrated in ADM4RK. For each of the 16 terms in the DVALUE array the following criterion is used in defining the corresponding UPBND and DNBND. If the absolute value of DVALUE(J) is less than or equal to 1.0, define UPBND(J) = CHIGH(J) and DNBND(J) = CDØWN(J). If the $|DVALUE(J)| > 1.0$, then UPBND(J) = CHIGH(J) x $|DVALUE(J)|$ and DNBND(J) = CDØWN(J) x $|DVALUE(J)|$.

Next, the trajectory option code LØPT is tested. If LØPT is less than 3, i.e., 0, 1, or 2, control passes to statement 31. If LØPT equals 4, control passes to the calculations for the tabular input wind tunnel conditions which start at statement 41. If LØPT equals 3, the appropriate internal program quantities are equated in turn to each set of tabular trajectory variables in the input table and the $\dot{q}$'s and $\dot{m}$'s are zeroed, then the calculations for the drag coefficient are performed.

Beginning with statement 41, an identical procedure is followed with the quantities corresponding to the wind tunnel inputs, i.e., the internal variables are set equal to the (NK+1) corresponding set of variables in the input table, the drag calculations for the appropriate mass loss option are performed, the results printed out, then the value of the counter NK is increased by one, and control returned to statement 3, the procedure is repeated until NK reaches MAXVAL, the maximum number of values in input table.

In order to calculate a trajectory, LØPT = 0, 1, or 2, following the CONTINUE statement 31, SUBROUTINE ADM4RK is called in to perform the integration of the variables and to call in DEREQ which controls the calculation of the derivatives. Control returns from ADM4RK after the designated printout interval with the new values of the variables, DVALUE. TIME is set equal to DVALUE(3) and Z to ZUSE, then the error code LP is tested. If LP is 6, an error message is printed out and the printout sequence (which contains the incorrect numbers) is performed before the program terminates. If LP is not equal to 6, control passes to statement 7.

At statement 7, LØPT is tested. If LØPT is non-zero, control passes to statement 30; if LØPT = 0, indicated a three degree of freedom in rotation trajectory, the quantities IKMAX and IKMIN are tested. If IKMAX or IKMIN equals 200, control passes to statement 91; if not, the value of the integer IALP directs control to the appropriate statement within group of equations testing the maximums and minimums in angle of attack, statements 8 to 30 where the integrated α correction term GRATE is defined.

$$\text{GRATE} = \int_{\tau_1}^{\tau_2} \left(\bar{A}_2 \, \alpha + \bar{A}_3 \, \alpha^2 \right) d\tau$$

$$\bar{A}_2 = A_{12} + A_{15}\theta + A_{18}\theta^2 + (A_{21} + A_{24}\theta + A_{27}\theta^2)\lambda + (A_{30} + A_{33}\theta + A_{36}\theta^2)\lambda^2$$

$$\bar{A}_3 = A_{13} + A_{16}\theta + A_{19}\theta^2 + (A_{22} + A_{25}\theta + A_{28}\theta^2)\lambda + (A_{31} + A_{34}\theta + A_{37}\theta^2)\lambda^2$$

Following statement 30, if LØPT equals 2, indicated a simplified angle of attack trajectory using the Bessel function model, the envelope value of α

is set equal to the instantaneous α. Following statement 91, the same procedure takes place for the case where LØPT = 1, particle trajectory option where α may be read in from a table. Additional quantities required for calculated trajectory output are calculated, before tests of printout parameters are made. If LØPT is less than 3, control passes to statement 100; otherwise, printout quantities for input trajectory or wind tunnel conditions are defined and control passes to statement 80.

Following statement 100, the counter NNTAPE is increased by 1 (used with the tape storage quantities). Next, if the plotting counter, LPLØT, has reached the allowable maximum of 160, control passes to statement 80. If not, LPLØT is increased by one, if LPLØT is now equal to 160 a message that plots have been cut off. Next, the plotting storage quantities are equated the appropriate program variables then the CONTINUE statement 80 is reached. If IØP(74) equals 1, SUBROUTINE WAKE is called in to control the calculation of the properties of the vehicle wake. If the integer code NPRINT equals zero, control passes to statement 90, bypassing all detailed trajectory printout. If NPRINT is non-zero, the detailed trajectory printout proceeds in the following order (1) the translational trajectory parameters, (2) the drag coefficient, its components, the interaction and rarefaction parameters, (3) vehicle configuration parameters (4) then, if MØPT is non-zero, the pressure, heating, and mass loss distributions (5) if LØPT = 0 or 2, the rotational quantities, (6) if INALPH > 0 or LØPT = 1, the tabular value of angle of attack. If LØPT is greater than 2, control passes to statement 49 where NK is tested against MAXVAL. If all tabular input for wind tunnel and input trajectory options has not been used, control returns to statement 3 after NK is increased by 1; otherwise, control passes to statement 834.

Statement 90 is the beginning of the stopping tests. If neither the stopping time nor stopping altitude has been reached, control passes to statement 50 and calculations continue; otherwise, control passes to statement 51. Statement 51 calls in SUBROUTINE RITØUT and then passes on to test the NPLØT array. If all five values of NPLØT are zero, control passes to statement 833 bypassing the plotting; otherwise, it passes to statement 82. Following statement 82, the individual NPLØT's are tested and accordingly SUBROUTINE AVPLT is summoned to perform the appropriate plots.

Following statement 833, IØP(74) is tested; if equal to zero, control passes to statement 837, bypassing the testing of quantities IØP(77) through IØP(82) which, if equal to 1, cause SUBROUTINE AVPLT to be called to plot the parameters WL1P, WL2P, WL3P, WR1P, WR2P, and WR3P respectively. After statement 837, the tape option ITAPE is tested; if equal to zero, control passes to statement 834; if non-zero, the tape containing V, β , and Z is written. Following statement 834, the units of input angles are changed from radians back to degrees to prepare for a new case before returning to the calling subroutine.

5. Other Information

A. SUBROUTINE VIXEN is called by SUBROUTINE F123.

B. SUBROUTINE VIXEN calls in the following subroutines:

1. SUBROUTINE CHNTBL
2. SUBROUTINE PRELIM
3. SUBROUTINE AERØDY
4. SUBROUTINE MASSLØ
5. SUBROUTINE TØMALØ
6. SUBROUTINE DRAGCØ
7. SUBROUTINE ADM4RK
8. SUBROUTINE WAKE
9. SUBROUTINE RITØUT
10. SUBROUTINE AVPLT

C. SUBROUTINE VIXEN calls in the library function FDXPI.

SUBROUTINE CHNTBL (DVALUE, ZTURNX, LCHNGE)

1. Purpose

SUBROUTINE CHNTBL assigns the appropriate geometric and material properties as initial values at the reentry altitude Z0 and at altitude ZTURN, where the integration of the derivatives is restarted after an input discontinuous change in geometry and heatshield material. In addition, some initial values are set, certain numerical factors are defined, and some tests on input quantities performed.

2. Input

* indicates integer quantity and NØCCUR number code

Name	Symbol	Occur/Noccur Number	Source of Input	Descri
CMQIN1	$C_{m_{q_1}}$	124	READIT	input C_{m_q} - first
CMQIN2	$C_{m_{q_2}}$	125	READIT	input C_{m_q} - seco (after shape chang
LA	La	033	VIXEN	instantaneous vehi
LA1	La_1	138	F123	input axial length -
LA2	La_2	144	F123	input axial length - (after shape change
LAMDA	λ	32	VIXEN	instantaneous blunt
LAMDA1	λ_1	137	READIT	input bluntness rat
LAMDA2	λ_2	143	READIT	input bluntness rat (after shape change
LCHNGE		*	VIXEN	control code
MATLN1		20*	SR2490 or READIT	input material case
MATLN2		21*	SR2490 or READIT	input material code
MHEAT		10*	SR2490 or READIT	input mass loss co
MXTAB1		16*	SR2490 or READIT	No. of values in X_c first configuration
MXTAB2		17*	SR2490 or READIT	No. of values in X_c second configuratio
NGEØM		15*	SR2490 or READIT	input geometry cod
NØSEØP		05*	READIT	input nose blunting

Description	Value Preset	Units
input C_{m_q} - first configuration		
input C_{m_q} - second configuration (after shape change)		
instantaneous vehicle axial length		ft.
input axial length - first configuration		ft.
input axial length - second configuration (after shape change)		ft.
instantaneous bluntness ratio		
input bluntness ratio - first configuration		
input bluntness ratio - second configuration (after shape change)		
control code		
input material case - first configuration	1	
input material code - second configuration	1	
input mass loss code	0	
No. of values in X_{cg}/D, I, I_x table - first configuration	1	
No. of values in X_{cg}/D, I, I_x table - second configuration (after shape change)	1	
input geometry code	1	
input nose blunting code		

B

2. Input (Cont'd)

Name	Symbol	Occur/Noccur Number	Source of Input	D
RB	Rb	053	VIXEN	instantaneo
RB1	Rb_1	136	F123	input base (after shap
RB2	Rb_2	142	F123	input base
RN	Rn	052	VIXEN	instantaneo
RN1	Rn_1	135	F123	input nose
RN2	Rn_2	141	F123	input nose (after shap
TABI1, 50	I_1 (Table)	2933-2982	READIT or SR2490	input I tabl
TABI2, 50	I_2 (Table)	2983-3032	READIT or SR2490	input I tabl (after shap
TABIX1, 50	I_{x_1} (Table)	3033-3082	READIT or SR2490	input I_x tab
TABIX2, 50	I_{x_2} (Table)	3083-3132	READIT or SR2490	input I_x tab (after shap
TABZ1, 50	Z_1 (Table)	3133-3182	READIT	input Z for configurati
TABZ2, 50	Z_2 (Table)	3183-3232	READIT	input Z for configurati
THETA	θ	076	VIXEN	cone half a
THETA1	θ_1	134	READIT	input cone
THETA2	θ_2	140	READIT	input cone (after shap
TW1	$T_{w_{o_1}}$	149	READIT or SR2490	input initial configurati

Description	Value Preset	Units
instantaneous base radius		ft.
input base radius - first configuration (after shape change)		ft.
input base radius - second configuration		ft.
instantaneous nose radius		ft.
input nose radius - first configuration		ft.
input nose radius - second configuration (after shape change)		ft.
input I table - first configuration	1.0	$slug\text{-}ft^2$
input I table - second configuration (after shape change)	1.0	$slug\text{-}ft^2$
input I_x table - first configuration	1.0	$slug\text{-}ft^2$
input I_x table - second configuration (after shape change)	1.0	$slug\text{-}ft^2$
input Z for $X_{c_g}/D, I, I_x$ function - first configuration		ft.
input Z for $X_{c_g}/D, I, I_x$ function - second configuration (after shape change)		ft.
cone half angle of current configuration		radians
input cone half angle - first configuration		deg.
input cone half angle - second configuration (after shape change)		deg.
input initial wall temperature - first configuration	1200	^oR

B

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	Descr
TW2	$T_{w_{o_2}}$	168	READIT or SR 2490	input initial w configuration
TXCGD1, 50	$(Xcg/D)_1$	2833-2882	READIT	input X_{cg}/D t
TXCGD2, 50	$(Xcg/D)_2$	2883 - 2932	READIT	input X_{cg}/D t
W	W	084	VIXEN	initial weight
W1	W_1	133	READIT	input weight -
W2	W_2	139	READIT	input weight -
ZTURN	Z_{turn}	145	READIT or SR2490	altitude at wh

3. Output

BETA1	β_1	004		sublimation r
BETA2	β_2	005		sublimation r
BETA3	β_3	006		order of reac
BETA4	β_4	007		activation ter
CMQIN	C_{m_q}	196		input C_{m_q} f
CØST	cos θ	008		cosine of con
CP2	C_{p_2}	014		specific heat
CPG	C_{p_g}	015		specific heat

Source of Input	Description	Value Preset	Unit
READIT or SR 2490	input initial wall temperature - second configuration	1200.	^{o}R
READIT	input X_{cg}/D table - first configuration		
READIT	input X_{cg}/D table - second configuration		
VIXEN	initial weight - (Δ weight) ablation		lb.
READIT	input weight - first configuration		lb.
READIT	input weight - second configuration		lb.
READIT or SR2490	altitude at which configuration is changed	-1.0	pt.
	sublimation rate coefficient		$\frac{ft}{sec\ ^{o}R}$
	sublimation rate coefficient		$\frac{ft}{sec\ ^{o}R}\beta_3$
	order of reaction		
	activation temperature		^{o}R
	input C_{m_q} for current configuration		
	cosine of cone half angle		
	specific heat of solid		$\frac{Btu}{lbm\ ^{o}R}$
	specific heat of gas		$\frac{Btu}{lbm\ ^{o}R}$

B

3. Output (Cont'd)

Name	Symbol	Occur/Noccur Number	
DELHC	ΔH_c	023	heat of
DELRHØ	$\Delta\rho$	022	differe
DVALUE			resulti
EPSIL	ϵ	024	coeffic
F	F	025	heat of
FACTR2		190	numeri
FACTR3		191	numeri
FACTR4		192	numeri
FACTR5		193	numeri
FACTR6		194	numeri
FACTR7		195	numeri
HREF	H_{ref}	030	constan
JJHØLD		01 *	control
LA	La	033	instant
LA1F	La_{1F}	146	value o
LAMD1F	λ_{1F}	151	value o
LAMDA	λ	032	instant

occur er	Description	Units
	heat of decomposition	$\frac{Btu}{lbm}$
	difference between virgin and char density	lbm/ft^3
	resulting values of 16 variables being integrated	
	coefficient of emission	
	heat of ablation	Btu/lbm
	numerical factor used in subroutine EVIL	
	numerical factor used in subroutine EVIL	
	numerical factor used in subroutine EVIL	$\frac{Btu}{ft^3\ {}^oR}$
	numerical factor used in subroutine EVIL	$\frac{Btu}{ft^3}$
	numerical factor used in subroutine EVIL	LBM/ft^3
	numerical factor used in subroutine EVIL	LBM/ft^3
	constant = 0 for no combustion	
	control code	
	instantaneous vehicle axial length	ft.
	value of axial length for last Z before ZTURN	in.
	value of bluntness ratio for last Z before ZTURN	
	instantaneous bluntness ratio	

B

3. Output (Cont'd)

Name	Symbol	Occur/Noccur Number	Descript
MATLNØ		13 *	material numb
MAXTAB		04 *	No. of values i configuration
NGL	η_{GL}	039	laminar transp
NGT	η_{GT}	041	turbulent trans
NSL	η_{SL}	038	laminar transp
NST	η_{ST}	040	turbulent trans
PI	π	042	mathematical c
RB	R_b	053	instantaneous b
RB1F	$R_{b_{1F}}$	147	value of base r
RESINT	$(\sin\theta)^{-1}$	054	inverse of sin
RHØ2	ρ_2	058	char density
RN	Rn	052	instantaneous r
RN1F	Rn_{1F}	169	value of Rn at
SINT	$\sin\theta$	.64	sine of cone ha
SQCØST	$(\cos\theta)^2$	066	square of cosi
TABI, 50	$I_{(table)}$	894-943	moment of ine
TABIX, 50	$I_{x(table)}$	944-993	table of momer configuration
TABZ, 50	$Z_{(table)}$	994-1043	Z table for X_c

Description	Units
material number code for current configuration	
No. of values in X_{cg}/D, I, I_x table for current configuration	
laminar transpiration factor of gas	
turbulent transpiration factor of gas	
laminar transpiration factor of solid	
turbulent transpiration factor of solid	
mathematical constant	
instantaneous base radius	ft.
value of base radius for last Z before ZTURN	in.
inverse of sin θ	
char density	lbm/ft^3
instantaneous nose radius	ft.
value of Rn at last Z before ZTURN	in.
sine of cone half angle	
square of cosine of cone half angle	
moment of inertia table for current configuration	$slug/ft^2$
table of moment of inertia about x axis for current configuration	$slug/ft^2$
Z table for X_{cg}/D, I, I_x of current configuration	ft.

B

3. Output (Concl'd)

Name	Symbol	Occur/Noccur Number	Descr
TANT	TAN θ	070	tangent of cone ha
THET1F	θ_{1F}	150	value of θ at last
THETA	θ	076	cone half angle of
THETAD	θ_D	069	cone half angle of
TW, 32	Tw_i	2644-2675	matrix of wall ter
TW0	Tw_0	074	initial wall tempe
TWSTAG	Tw_{STAG}	073	wall temperature
TXCGD, 50	$X_{c.g.}/D$	844-893	X_{cg}/D table for c
W	W	084	initial weight - (Δ
W0	W_0	085	initial weight for
W1F	W_{1F}	170	value of total weig
WTH	W_{TH}	204	initial weight (Δ
ZTURNX			altitude control c

Description	Units
tangent of cone half angle	
value of θ at last Z before ZTURN	degrees
cone half angle of current configuration	radians
cone half angle of current configuration in degrees	degrees
matrix of wall temperatures along body	^{o}R
initial wall temperature for current configur ation	^{o}R
wall temperature at stagnation point	^{o}R
X_{cg}/D table for current configuration	
initial weight - $(\Delta \text{weight})_{\text{ablation}}$	lb
initial weight for current configuration	lb
value of total weight at Z just before ZTURN	lb
initial weight (Δ weight) thrusting	lb
altitude control code	ft.

B

4. Numerical Procedure

The equations in SUBROUTINE CHNTBL may be grouped into four main sections as follows: (1) definition of initial geometry at reentry, (2) definition of initial geometry at input altitude ZTURN, (3) definition of heatshield material properties, (4) miscellaneous calculations of numerical factors, setting initial values, and tests on inputs. The subroutine is called in at the reentry altitude Z0, where code LCHNGE = 1, to define the initial geometric and material properties. The subroutine is again called in at the first altitude which is either $\leq$ ZTURN, to define to the new initial configuration.

The subroutine initially sets JJHOLD = 0, then using a directed GØ TØ statement proceeds to statement 100 of LCHNGE = 1, initial reentry altitude, or to statement 200 of LCHNGE = 2, at or just below altitude ZTURN. Statement 100, the beginning of the first section, sets ZTURNX = ZTURN, then the tables for X_{cg}/D, I, I_x as a functions of Z, to be used in the remainder of program, are set equal to tables $(X_{cg}/D)_1$, I_1, I_{x_1} as functions of Z_1. CMQIN is set = CMQIN1, MAXTAB = MXTAB1, TW0 = TW1, MATLNØ = MATLN1, and JJ = 151. The directed GØ TØ statement controlled by the value of NGEØM causes the flow of the subroutine to be directed to the group of geometry calculations consistent with the input quantities. If NGEØM = 1, the flow of the subroutine is directed to the group of equations beginning with statement 102. Here the input quantities W1, THETA1, RN1, RB1 are respectively set equal to internal program quantities W, THETA, RN, RB, the quantity LA1 is calculated for use in SUBROUTINE AERØDY, and finally skips to statement 300.

For the other input geometry options, a similar procedure is followed. When NGEØM is 2, the group of equations starting with statement 103 is utilized. The internal program quantities W, THETA, RB, and RN are respectively set equal to the input quantities W1, THETA1, RB1 and

RB1*LAMDA1, and LA1 is calculated before proceeding to statement 300. The NGEØM = 3 option utilizes the group of equations beginning with statement 104. The internal program quantities W, RN, RB are respectively set equal to the input quantities W1, RN1, RB1. The cone half angle, THETA is calculated from the input LA1, RN1, RB1, then control goes to statement 300.

If the parameter LCHANGE is equal to 2, the directed GØ TØ statement causes control to go to statement 200, which marks the beginning of the equations defining the second initial configuration. Here, ZTURNX is set equal to a negative number, which prevents this subroutine from being called again. The final values of RN, RB, THETA, LA, LAMDA, WTØTAL at altitude just before ZTURN are stored under respective designations RN1F, RB1F, THETIF, LA1F, LAMDIF, and W1F. After these definitions, the remainder of this section of equations up to statement 300 exactly parallels the geometry calculations of the initial configuration.

The assignment of material properties for both configurations begins with statement 300. The material number code MATLNØ is tested and if it is larger than the allowable maximum value of 6 then an error message is written out, MATLNØ is set equal to 1, and the calculations continue. The directed GØ TØ, statement number 55, causes control to go to the appropriate set of material properties based on the value of MATLNØ. If MATLNØ = 1, control is directed to statement number 31 and the teflon properties; MATLNØ = 2 to statement 37 and LT_{α} properties; MATLNØ = 3 to statement 32 and ØTWR properties; MATLNØ = 4 to statement 38 and phenolic nylon properties; MATLNØ = 5 to statement 39 and carbon phenolic properties; MATLNØ = 6 to statement 33 and the input material properties. The sixteen input material properties are set equal to OCCUR values ØCCUR(JJ + 1), ØCCUR(JJ + 2), etc. where the appropriate JJ is defined with the configuration geometry as JJ = 151 for

the first configuration and JJ = 170 for the second configuration. After the material properties are defined control is directed to statement 34.

The last section of the subroutine performs the following functions: (1) Tests to ensure that THETA is within the range of applicability; if not, writes an error message and stops program; (2) sets THETAD equal to cone half angle in degrees, changes units of THETA to radians, and defines useful trigonometric functions of THETA; (3) sets the initial values of geometric quantities to be integrated; (4) sets initial values of TW matrix; (5) calculates numerical factors to be used in SUBROUTINE EVIL; and (6) tests program input options.

5. Other Information

A. SUBROUTINE CHNTBL calls in the following functions:

1. ASINR
2. DSIN
3. DCØS
4. DSQRT
5. DLØG

B. SUBROUTINE CHNTBL is called by SUBROUTINE VIXEN.

SUBROUTINE RITØUT

1. Purpose

SUBROUTINE RITØUT prints out a summary of the maximums and minimums in angle of attack as determined by the testing in SUBROUTINE VIXEN.

2. Input See VIXEN for locations in ØCCUR array of input quantities.
* indicates integer quantity

Name	Symbol		Description
ALMAX, 200	α'_{max}		maximums in α' from VIXEN
ALMIN, 200	α'_{min}		minimums in α' from VIXEN
FMAX, 200	f_{max}		frequencies corresponding to α'_{max}'s
FMIN, 200	f_{min}		frequencies corresponding to α'_{min}'s
IKMAX		*	number of maximums (max. 200)
IKMIN		*	number of minimums (max. 200)
NPRINT		*	printing option code
TAMAX, 200	t_{max}		times corresponding to α'_{max}'s
TAMIN, 200	t_{min}		times corresponding to α'_{min}'s
ZMAX, 200	Z_{max}		altitudes corresponding to α'_{max}
ZMIN, 200	Z_{min}		altitudes corresponding to α'_{min}'s

3. Output

None

4. Numerical Procedure

SUBROUTINE RITØUT begins by testing the printing code NPRINT. If NPRINT equals zero, control returns to the calling subroutine; if NPRINT is non-zero, control passes to statement 51. Statement 51 causes the program to printout the titles in FØRMAT statement 1051. Then the DØ loop including all the statements through 52 is executed for all values of J from 1 through IKMAX. Inside the loop each value of ALMAX(J) is multiplied by 57.29578 to change units from radians to degrees for the printout, then a write statement causes each set of t_{max}, z_{max}, f_{max}, a'_{max} to be printed out in form of FORMAT statement 1052. The printout of the minimums follows an identical procedure in the DØloop encompassing all the statements through 53, before returning to the calling subroutine.

5. Other Information

A. SUBROUTINE RITØUT is called by SUBROUTINE VIXEN.

B. SUBROUTINE RITOUT calls in no other subprograms

3.1.2. Numerical Integration of Trajectory Variables

The following section describes the operation of the predictor - corrector integration routine, ADM4RK, which summons SUBROUTINE DEREQ, the controlling subroutine for the derivative calculations.

SUBROUTINE ADM4RK (NZ, ZDEL, VALUE, DERN, UPBND, DNBND, FACTOR, FREQ, HLIMIT, LZ, ZXINDE, DELMIT)

1. Purpose

SUBROUTINE ADM4RK performs the integration of NZ first order differential equations of the form $\frac{dy_i}{dx} = f_i(x, y_1, \ldots y_{NZ})$ $i = 1, \ldots NZ$ by a four point predictor-corrector method which will alter the integration interval to maintain a required accuracy.

2. Input

Name	Source of Input	Description
DELMIT	VIXEN	the minimum value that the delta of integration is allowed to have
DERN	DEREQ	the array of NZ derivatives
DERNN	DEREQ	the array of NZ derivatives
DNBND	VIXEN	the lower bound on the absolute difference
FACTOR	VIXEN	the fraction by which the delta of integration is increased or decreased; must be less than 1.
FREQ	VIXEN	the interval of the independent variable, XINDEP, at which control is returned to the calling program (if L = 1 initially)

2. Input (Concl'd)

Name	Source of Input	Description
HLIMIT	VIXEN	the upper limit of integration; if L was initially 1, control returns to calling program when this value is reached
L	DEREQ	the control parameter; the value input may be either 1 or -5. It is reset by ADM4RK and should not be modified by user. See other information.
LZ	VIXEN	same as L
NZ	VIXEN	the number of equations to be integrated
UPBND	VIXEN	the upper bound on the absolute difference that is allowed between the extrapolated and interpolated values
VALUE	VIXEN	the array of NZ integrated values; on the first pass, this array is input from the initial conditions in VIXEN
ZDEL	VIXEN	the delta of integration supplied initially by VIXEN and modified by ADM4RK
ZXINDE	VIXEN	the value of the independent variable

3. Output

Name	Description
L	see Input and Other Information
TINDEP	designation for the independent variable in Runge Kutta evaluation

3. Output (Concl'd)

Name	Description
VALUE	the array of NZ integrated values
VALUEN	the array of NZ function values evaluated by the Runge Kutta calculation
XINDEP	same as ZXINDE
ZBAR	the extrapolated functional values of the Adams-Bashforth method
ZXINDE	see input

4. Numerical Procedure

The initial value X_o (ZXINDE) of the independent variable and the corresponding initial conditions $Y_1, Y_2, \ldots Y_{NZ}$ (the array VALUE) for the NZ equations is supplied to SUBROUTINE ADM4RK by SUBRØUTINE VIXEN. The routine first calculates the derivatives at X_o, then immediately returns control to the calling program to allow printout. When ADM4RK is reentered, it uses a fourth order Runge-Kutta method to start the integration by obtaining values of each function and its derivative at 3 equally spaced points X_1, X_2, X_3 separated by the interval ΔX (DEL). Next using the Adams-Bashforth method, the functional values at X_3 and the derivatives at all 4 points are used to extrapolate new functional values at $X_4 = X_3 + \Delta X$. The associated derivatives at X_4 are computed, and new interpolated functional values at X_4 calculated from

the derivatives at X_1, X_2, X_3, X_4 and the functional values at X_3. This repeated extrapolation and interpolation process is carried on until the end of the integration interval (HLIMIT) is reached.

The differences between the extrapolated and interpolated results are used to control the accuracy of the integration. If, at any point, this difference for any of the functions is larger than the specified upper limit UPBND, the ΔX of integration is reduced by the chosen factor FACTØR (i.e. DEL becomes DEL*(1.-FACTØR). Similarly, if this difference is smaller than the specified lower limit DNBND, ΔX is increased by the factor FACTØR (i.e. DEL becomes DEL*(1. + FACTOR)). The minimum ΔX used by the routine is DELMIT, specified by the VIXEN. If the required accuracy cannot be maintained, the control code L is set to 6, and control is returned to the calling program.

SUBROUTINE ADM4RK begins calculations by defining the quantities N, L, DEL, XINDEP which are respectively equal to NZ, LZ, ZDEL, ZXINDE. The input DELMIT, the minimum allowable delta of integration, is tested. If DELMIT is zero, it is set equal to DEL/1000. before control passes to statement 9002. Next, the control parameter L is tested. A negative L, causes L to be redefined as the absolute value of L before control passes to statement 23, which begins the calculations appropriate to the initial pass through the subroutine. For a zero value of L, control passes to 201 where L is reset to 6 before control is directed to statement 4. The directed GØ TØ of statement 1211 is executed when L is positive. From statement 1211, the subroutine flow is directed to statement 23 for

initial pass calculations when L is 1, to statement 204 for calculations preliminary to integration calculations for L values of 2, 3, and 4, or for L = 5 to statement 24 which is the beginning of the integration calculations.

The calculations following statement 23 for the initial pass consist of: (1) the definition of XFREQ (=XINDEP + FREQ), the next value of the independent variable for which control will be returned to the calling program, (2) the control code LL is set to 2, (3) the values of DEL and XINDEP are saved for the next pass as SAVDEL and SINDEP, respectively, (4) the value 41 is assigned to M which causes the entry into Runge Kutta routine from the assigned GØ TØ of statement 24,(5) SUBROUTINE DEREQ is called to calculate the derivatives which correspond to the initial values of the dependent variables,(6) if L has the error value of 6 control is sent to 220, otherwise if L ≠ 6 to statement 40.

Statement 204 is the beginning of a DØ loop which sets the N values of the derivative DERN and the N integrated values, VALUE, equal to the appropriate saved values SAVD and SAVE, respectively. The independent variable XINDEP is then set equal to the stored value SINDEP and control passes to statement 24.

Statement 24 is an assigned GØ TØ statement which directs flow to statement 41 and the Runge-Kutta starting integration scheme when M has been assigned the value 41. When M has the assigned value 42, control is directed to statement 42, the beginning of the Adams-Bashforth predictor-corrector integration procedure.

The calculations using the Runge-Kutta procedure begin with statement 41. Statement 41 is the beginning of a DØ loop which defines the N values of DERNM3 the derivatives at the initial point X_o from their counterparts in the DERN array, the derivatives at the initial value of the independent variable. Then the calculations contained in the DØ loop ending with statement 43 are performed. Each of the three passes through this loop represents the determination of the N functional values and the N derivatives at one of the three points - X_1, X_2, or X_3 - which are separated by the interval DEL. The calculations involved are as follows: First W1 is set equal to DEL/2.0 then the B0 array is zeroed. The DØ loop following statement 44 which ends with statement 50 nested inside the DØ loop ending at 43 evaluates the following set of equations, where the index J represents the subscript of the K quantities, X_o is initial point of the independent variables, $F_i(x)$ is dy_i/dx (DERN), and h is the delta of integration (DEL):

$$K_{1_i} = f_i(x_o)\, h \qquad j = 1 \qquad i = 1, \ldots N$$

$$K_{2_i} = f_i(x_o + \frac{h}{2})\, h \qquad j = 2 \qquad i = 1, \ldots N$$

$$K_{3_i} = f_i(x_o + \frac{h}{2})\, h \qquad j = 3 \qquad i = 1, \ldots N$$

$$K_{4_i} = f(x_o + h)\, h \qquad j = 4 \qquad i = 1, \ldots N$$

$$(\Delta y)_i = \frac{1}{6} \left[K_1 + 2K_2 + 2K_3 + K_4 \right]$$

$$x_1 = x_o + h, \quad (y_1)_i = (y_o)_i + (\Delta y)_i\, h$$

Each pass through this loop represents the evaluation of one of the K factors for each of N quantities being integrated, the incrementing of the f used in the K calculations (done by calling DEREQ to calculate derivatives at half steps h/2), and the evaluation of the new values of y_i's and the new x. The statements 45 through 49 define the appropriate numerical factors for use in the evaluating the K's and new y_i's. Following statement 50 is a DØ loop ending at 61 which stores the N derivatives calculated for each of the points X_1 and X_2 - DERNM2 and DERNM1, respectively - and redefines the VALUE array to equal the final VALUEN array of the preceding DØ loop. Following 61, the independent variable XINDEP is set the final value of TINDEP from the preceding DØ loop before statement 43 is reached.

When the calculations of the functional values and derivatives for the points X_1, X_2, and X_3 are completed (DØ loop ending with 43 has been executed), the quantities H1, H2, and H3 are each given the value of DEL. The quantity M is then assigned the value 42, this causes control to pass directly to the Adams-Bashforth integration from statement 24 for all subsequent passes through ADM4RK, unless the integration is restarted by setting L equal to 1 or -5. Control skips to statement 9042 bypassing statement 42 and the definition of XFREQ.

Statement 42 marks the beginning of the Adams-Bashforth predictor-corrector integration scheme. Following 42, the quantity XFREQ is increased by the amount FREQ before 9042 is reached. The directed GØ TØ then sends control to statement 1406 if L = 1, to 1207 if L is 5,

or to 1407 if L is 2, 3, or 4 (indicating respectively that a FREQ interval, HLIMIT, or both have been reached). The block of calculations between statements 1407 and 1406 describe the FREQ and HLIMIT testing. Statement 1407 tests the absolute value of the difference between XINDEP and XFREQ (to determine if a FREQ interval has been reached) against the absolute value of DEL. If $|XINDEP - XFREQ| \geq |DEL|$, a FREQ interval has been reached and control passes to 1408; if $< |DEL|$, a FREQ interval has not been reached and the test in 14060 is performed. Statement 14060 determines whether HLIMIT has been reached by testing $|XINDEP - HLIMIT|$ against $|DEL|$. If $|XINDEP - HLIMIT| > |DEL|$, the limiting value has not been reached and control passes to 1406 where the integration continues. If $|XINDEP - HLIMIT| \leq |DEL|$, the limiting value has been reached and control is sent to 1408.

The CONTINUE statement 1408 is followed by a DØ loop ending at 1410 which sets the N element arrays ZBAR, DERNN, DERN, DERNM1 and DERNM2 equal respectively to the arrays VALUE, DERN, DERNM1, DERNM2, and DERNM3. The code LL is set to 1, then statement 11 defines the quantity TEMP = |XINDEP - XFREQ|, which is subsequently tested against |DEL|. If TEMP > |DEL|, indicating that a FREQ interval has not been reached, control passes to statement 15. On the other hand, if TEMP $\leq$ |DEL|, the following quantities are defined: L = 2, SINDEP = XINDEP, SAVDEL = DEL, TEMP1 = |DEL|/DEL, DEL = TEMP*TEMP1 before reaching 15. Statement 15 redefines TEMP to be |XINDEP - HLIMIT|. This quantity is then tested to determine whether the limiting value of the independent variable has been reached. If TEMP < |DEL|, L is set equal to 1 before control passes to 18; if TEMP > |DEL|, control skips to 17; if TEMP = |DEL|,

control passes to 18. Statement 18 increases the value of L by 2 then the following are defined: SAVEDEL = DEL, TEMP1 = |DEL|/DEL, and DEL = TEMP*TEMP1 before 17 is reached. Statement 17 directs control to 22 if L = 1 or 5 or to 6 if L = 2, 3, or 4 (after passing through the previously described block of equations L must be either 2, 3, or 4.

The block of equations between statements 1406 and 2 performs the Adams-Bashforth predictor-corrector integration calculations. Following the CØNTINUE statement 1406, LL and L are both set to 1 before CØNTINUE statement 1207 is reached. After 1207, the following factors are defined beginning at statement 12 where H's are distances between the points

$W1 = H1 + H2$

$W2 = (H1 + H2)^2$

$W3 = 2H1 + H2$

$W4 = H1 + H2 + H3$

$W5 = 2(H1 + H2) + H3$

$W6 = 4H1 + 3H2 + H3$

$W7 = 2H1 + H2 + H3$

$W8 = (H1 + H2)\,H1$

$W9 = (H1 + H2)\,H3$

$W10 = H2 + H3$

$W11 = H2*H3$

$W12 = H1*H2$

$W16 = (H1 + H2)^2 + H3(H1 + H2)$

If L is 2, 3, or 4, control passes next to statement 202. If L is 1 or 5, TEMP is defined as XINDEP + DEL in statement 14 and DEL is set to TEMP - XINDEP. If |DEL| < DELMIT (indicating that the delta of integration is smaller than the specified minimum), control passes to 201

where L is set to 6 and then to statement 4. If $|DEL| \geq DELMIT$ control passes to 202. Beginning with statement 202, the following additional quantities are defined:

XINDEP = XINDEP + DEL

W13 = DEL/2.0

W14 = $DEL^3/2.0$

W15 = $DEL^2/3.0$

W17 = H1 + DEL

W18 = H1 + H2 + DEL

W19 = (H1 + H2) DEL

W29 = (2H1 + H2) DEL

These foregoing factors are used in the equations for determining the array of predicted values for the integrated variables, ZBAR, i.e.

$$B0(1) = \frac{DEL}{W4*W8}\,(W14 + W15*W6 + W13*(W2 + 2.0\,H1*W4 + W11) + H1*W6)$$

$$B1(1) = \frac{-DEL}{W10*W12}\,(W14 + W15*W5 + W13*W16)$$

$$B2 = \frac{DEL}{W1}\,(W14 + W15*W7 + W13*H1*W4)$$

$$B3 = \frac{-DEL}{W4*W10*H3}\,(W14 + W15*W3 + W13*W8)$$

$$ZBAR_i = (y_i)_{\substack{\text{predicted}\\ x = X_4}} = (y_i)_{\text{previous}} + B0(1)\left(\frac{dy_i}{dx}\right)_{x = X_3} + B1(1)\left(\frac{dy_i}{dx}\right)_{x = X_2}$$

$$+ B2\left(\frac{dy_i}{dx}\right)_{x = X_1} + B3\left(\frac{dy_i}{dx}\right)_{x = X_0}$$

SUBROUTINE DEREQ is then called to determine the N derivatives DERNN associated with the ZBAR values and XINDEP. If L has the value 6 indicating an error, statement 220 receives control. Otherwise, the N corrected integrated values, VALUEN, at point X_4 are determined from

$$A = (W14 + W15*W3 + W13*W8) / (W17*W18)$$

$$A0 = -(W14 + W15*(W3-DEL) + W13*(W8-W29) - DEL*W8)/W8$$

$$A1 = \frac{DEL}{W12*W17} (W14 + W15*(W1 - DEL) - W13*W19)$$

$$A2 = \frac{DEL}{H2*W1*W18} \left(\frac{W14}{3.0} + \frac{H1*W15}{2.0}\right)$$

$$VALUEN = (y_i)_{\substack{\text{corrected} \\ x = X_4}} = (y_i)_{\text{previous}} + A\left(\frac{dy_i}{dx}\right)_{\substack{x = X_4 \\ y = y \text{ predicted}}}$$

$$+ A0\left(\frac{dy_i}{dx}\right)_{x = X_3} + A1\left(\frac{dy_i}{dx}\right)_{x = X_2}$$

$$+ A2\left(\frac{dy_i}{dx}\right)_{x = X_3}$$

Then DIF the difference between the predicted and corrected value of each variable is evaluated DIF = |VALUEN - ZBAR|. This quantity is then tested against the upper accuracy bound specified for this difference, UPBND.

If the difference is within the limit, control passes to statement 13. If not, XINDEP is decreased by DEL, then DEL is decreased by the amount FACTOR*DEL after which L and LL are set to 1 for L < 5. Then control returns to 14 to perform the predictor-corrector calculation with the new delta of integration when L ≤ 5, or control passes to 220 for L = 6. After the CONTINUE statement 13, which is reached when the difference DIF ≤ UPBND, control passes to statement 5 if L = 2. For other values of L, the following quantities are redefined

H3 = H2

H2 = H1

H1 = DEL

and control passes to 11 for L = 1, to 5 for L = 2, 3, or 4, or to 22 for L = 5.

Statement 22 compares the N values of DIF with the specified lower accuracy bound DNBND; if DIF ≤ DNBND, DEL is increased by the amount DEL*FACTØR before reaching 7010; if DIF > DNBND, 7010 is reached directly. Statement 7010 sends control to statement 6 if L < 5, to 200 if L = 5, and to 220 if L = 6. Statement 200 sets SAVDEL = DEL then sends control to 21.

Statement 5 marks the beginning of the subroutines output procedure. It is followed by a DØ loop ending at 1040 which sets the N elements of the arrays VALUE and DERN respectively equal to ZBAR, the predicted values, and DERNN, the corresponding derivatives. Statement 21 then sets LL equal to 2. Then if L is either 1 or 5, the N elements of the following arrays are defined

DERNM3 = DERNM2

DERNM2 = DERNM1

DERNM1 = DERN

DERN = DERNN

VALUE = ZBAR

before statement 40 is reached. The code L is again tested. If L = 1, for a value of LL = 1 control passes to 1407, but for a value of LL = 2 L is reset to 2 and control passes to 206. If L = 2, 3, or 4, for LL = 1 control passes to 206 but for LL = 2 control passes to statement 4. If L = 5, a value of LL = 1 causes control to be sent to statement 12, but LL = 2 sends control to 4. Following CØNTINUE statement 206, a DØ loop stores the N derivatives DERN and the corresponding integrated values VALUE in the arrays SAVD and SAVE respectively. Then, if LL is 1, control passes to 12 or if LL = 2, to statement 4. Statement 4 sets DEL equal to the saved value SAVDEL and is followed by statement 220 where

NZ = N

LZ = L

ZDEL = DEL

ZXINDE = XINDEP

before the return to the calling program.

5. Other Information

A. SUBROUTINE ADM4RK is called by SUBRØUTINE VIXEN.

B. SUBROUTINE ADM4RK calls SUBRØUTINE DEREQ.

C. The control parameter L may have initial input values of 1 and -5 only. It is reset by ADM4RK and should not be modified by the user. The values 2, 3, 4, and 6 are those returned by SUBROUTINE ADM4RK under the following conditions.

1. L = 1. Indicates initial pass; must be set when FREQ and HLIMIT testing is desired.

2. L = 2. Indicates that a FREQ interval has been reached.

3. L = 3. Indicates that HLIMIT has been reached.

4. L = 4. Indicates that HLIMIT and FREQ interval have been reached simultaneously.

5. L = -5. Indicates that return to the calling program must be made after each successful integration step. This will be set to + 5 during the initial pass. No tests are made for FREQ or HLIMIT.

6. L = 6. An error return indicating that the integration interval is less than DELMIT.

D. Special care must be taken in modifying any elements of the ADM4RK calling sequence during a return to the calling program. In particular, it is impossible to modify any of the variables calculated within ADM4RK--DEL, VALUE, DERN, XINDEP, L-- since the routine saves its own values. In addition, if L is initially 1, FREQ must be set at least four times as large as the initial DEL, or no returns will be made before HLIMIT.

3.1.3 Evaluation of Derivatives for Trajectory Calculation

SUBROUTINE DEREQ controls the computation of the derivatives which are used in determining the trajectory. To accomplish this task, DEREQ calls for the preliminary calculations, heating and mass loss calculations where required, angle of attack calculations where required, the drag coefficient calculations, and the calculation of the translational quantities.

SUBROUTINE DEREQ(DVAL, ZUSE, DERIV, LL)

1. Purpose

SUBROUTINE DEREQ sets the 16 variables to their integrated values as determined in SUBROUTINE ADM4RK and calls in the subroutines according to the options being exercised to calculate the derivatives of these variables for the next step in the integration.

*indicates integer quantity and an NØCCUR number

Name	Symbol	Occur/Noccur Number	Source of Input	D
ALPTAB, 75	α (TABLE)	3646-3720	READIT	input α tabl
AREF	A_{ref}	001	PRELIM	reference a
AWREF	$A_{W\,ref}$	188	READIT	reference a
C_D	C_D	016	DRAGCØ	drag coeffic
CDTAB, 75	C_D(TABLE)	3383-3547	READIT	tabular inpu
DERIV, 16	-	-	TEQUAT and/or RØTATE	derivatives
DVAL, 16	-	-	ADM4RK	integrated
HTAB, 75	Z (TABLE)	3233-3307	READIT	input tabula
INALPH		30 *	READIT	input angle
LAMDA	λ	032	PRELIM	bluntness r
LL	-	- *	ADM4RK or PRELIM	integer err
MAXCD	-	18 *	READIT	input C_D co
MAXWCD	-	19 *	READIT	input C_{D_W}
MDØT, 32	$\dot{m}_i$	2708-2739	MASSLØ	mass loss r
MHEAT	-	10 *	READIT	input mass
MINF	M_∞	035	VIXEN or PRELIM	free stream

e of	Description	Units
IT	input α table	degrees
IM	reference area of vehicle	ft^2
IT	reference area corresponding to WCDTAB	ft^2
CØ	drag coefficient	-
IT	tabular input drag coefficient	-
AT and/or TE	derivatives with time of quantities to be integrated	-
RK	integrated values of variables	-
IT	input tabular altitude for use with CDTAB	ft.
IT	input angle of attack code	-
IM	bluntness ratio	-
RK or IM	integer error code	-
IT	input C_D code	-
IT	input C_{D_W} code	-
LØ	mass loss rate distribution	$\frac{lbm}{ft^2 - sec.}$
IT	input mass loss code	-
N or IM	free stream Mach number	-

B

Name	Symbol	Occur/Noccur Number	Source of Input	
MØPT	-	03 *	READIT	input heating/m
NØSEØP	-	05 *	READIT	input noseblunt
RBDØT	$\dot{R}_b$	060	NOSEBL	rate of change
RNDØT	$\dot{R}_n$	059	NOSEBL	rate of change
TW0	T_{W_0}	074	CHNTBL	initial wall tem
WCDTAB, 75	C_{D_W} (TABLE)	3458 - 3532	READIT	tabular input dr
WDØT	$\dot{W}$	086	TØMALØ	weight increme
WHTAB, 75	Z_W(TABLE)	3308- 3382	READIT	tabular altitude
XBAR	$\bar{\chi}$	090	PRELIM	interaction para
XUP	χ_{UP}	237	READIT or SR2490	upper limit on i
ZUSE	Z	-	VIXEN	altitude

rce of put		Units
ADIT	input heating/mass loss code	-
DIT	input noseblunting code	-
EBL	rate of change of base radius	ft/sec
EBL	rate of change of nose radius	ft/sec
TBL	initial wall temperature	^{o}R
DIT	tabular input drag coefficient increment	-
MALØ	weight increment due to ablation	lbm/sec
DIT	tabular altitude corresponding to C_{D_W}	ft.
LIM	interaction parameter	-
DIT or 90	upper limit on interaction parameter	-
EN	altitude	ft.

B

*indicates integer quantity

Name	Symbol	Occur/Noccur Number	
ALPHA	α	002	instantaneous
ALPRIM	α'	003	instantaneous
CD	C_D	016	total drag coe
DERIV, 16		-	derivatives wi in ADM4RK
GAMF	γ_F	026	flight path angl
LL		- *	integer error
P	P	043	angular velocit
PHI	ϕ	044	Euler angle, Φ
PSI	Ψ	045	Euler angle, Ψ
PSIALP	Ψ_a	200	thrust offset an
Q	Q	050	angular velocit
RB	R_b	053	base radius
RN	Rn	052	nose radius
SMR	R	065	angular velocit
THEALP	θ_a	071	Euler angle, Θ
TIMER	t (time)	080	time
TWSTAG	$T_{W_{STAG}}$	073	wall temperatu

ccur	Description	Units
	instantaneous angle of attack	radians
	instantaneous angle of attack	radians
	total drag coefficient	-
	derivatives with altitude of quantities being integrated in ADM4RK	-
	flight path angle	radians
	integer error code	-
	angular velocity	rad/sec
	Euler angle, Φ	radians
	Euler angle, Ψ	radians
	thrust offset angle	radians
	angular velocity	rad/sec
	base radius	ft.
	nose radius	ft.
	angular velocity	rad/sec
	Euler angle, θ_{α}	radians
	time	sec.
	wall temperature at the stagnation point	^{o}R

B

Name	Symbol	Occur/Noccur Number	De
V	V	082	velocity
W	W	084	(initial weight - Δ
WTH	W_{TH}	204	(initial weight - Δ
XR	X_R	087	range distance
YR	Y_R	199	side range distan
Z	Z	091	altitude

Description	Units
velocity	ft/sec
(initial weight - Δ ablative weight)	lb.
(initial weight - Δ thrusting weight)	lb.
range distance	ft.
side range distance due to thrust offset	ft.
altitude	ft.

B

4. Numerical Procedure

SUBROUTINE DEREQ begins by setting the internal program variable names equal to their corresponding integrated values, DVAL, from ADM4RK. These are transferred from ADM4RK as arguments of SUBROUTINE DEREQ. The variable names, their DVAL designations, and definitions are given below.

Name	DVAL	Definition
V	1	velocity
GAMF	2	flight path angle, negative number
TIME	3	time
XR	4	component of range in X direction
W	5	difference between the initial weight and weight loss due to ablation
RN	6	nose radius
RB	7	base radius
PSI	8	Euler angle, Ψ
THEALP	9	Euler angle, Θ_α
PHI	10	Euler angle, Φ
Q	11	angular velocity, Q
SMR	12	angular velocity, R
P	13	angular velocity, P
YR	14	side range (component of range in Y direction) due to offset thrust
PSIALP	15	thrust off set angle
WTH	16	difference between the initial weight and weight loss due to thrust.

The altitude Z is set equal to ZUSE, an argument of DEREQ, and TIME is set equal to TIMER. If Z is less than zero, control passes to statement 2 where an error message is printed out, the error code, LL, set to 6, and control returned to ADM4RK. If Z is greater than zero, SUBROUTINE PRELIM is called to perform preliminary calculations of geometric, flow field, and thrusting parameters. If the error code LL is set to 6 in SUBROUTINE PRELIM, control is returned to ADM4RK. Otherwise, the quantity MINF is tested; if MINF, M_∞, is less than 5.0, the LL is set to 6, an error message printed, and control passes to ADM4RK. If MINF $\geq$ 5.0, control passes to statement 20, where LAMDA, λ, is tested. If LAMDA $\leq$ 0.6, control passes to statement 22; if not, the error code LL is set to 6, an error message printed, and control returns to ADM4RK. Following statement 22, $\tilde{\chi}$ is tested against χ_{UP}. Since heating and mass loss calculations are performed only in the continuum flow regime or in the region of fairing between continuum and strong interaction flows, if $\tilde{\chi} < \chi_{up}$, control passes to statement 3 the beginning of the mass loss block. If $\tilde{\chi} \geq \chi_{UP}$, the wall temperature at the stagnation point is set to T_{W_0}, and the mass loss rates is used in DRAGCØ as well as the derivatives pertaining to mass loss and shape change are zeroed in the group of equations starting with statement 8. Then control passes to statement 4 bypassing the mass loss calculations.

The block of mass loss calculations begins with the testing of the input code MØPT in statement 3. If MØPT equals zero, control passes to statement 8 where the pertinent quantities and derivatives are zeroed before passing to statement 4. If MØPT $\neq$ 0, SUBROUTINE AERØDY

is summoned to compute the aerodynamic heating along the body. Then, if the input option code MHEAT equals zero, control passes to statement 8. If MHEAT ≠ 0, the mass loss calculations continue with the calling of SUBROUTINE MASSLØ which controls the calling of SUBROUTINE EVIL, where the mass loss rates, surface recession rates, and wall temperature are calculated. The mass loss rates are integrated in SUBROUTINE TØMALØ to obtain WDØT, the rate of change in weight due to ablation of the vehicle heatshield. Then DERIV(5) is set equal to WDØT. If input option code NØSEØP equals 1, control passes to statement 7 and the shape change calculations. If NØSEØP ≠ 1, the derivatives of the nose radius and base radius, DERIV(6) and DEPIV(7) repectively, are zeroed, before control passes to statement 4. Statement 7 calls SUBROUTINE NØSEBL to perform the shape change calculation, then DERIV(6) and DERIV(7) are respectively set equal to RNDØT and RBDØT before statement 4 is reached.

After statement 4, input option code INALPH is tested. If INALPH, number of values in the input angle of attack table, is greater than zero control passes to statement 17. There α and α' are set equal to a value determined, using FUNCTION TABLE, by linear interpolation of the input angle of attack, ALPTAB, as a function of HTAB; altitude and the time derivatives 8 through 13 for rotational angles and velocities are zeroed just before statement 18. If INALPH = 0, SUBROUTINE ROTATE is called to calculate angle of attack effects, if any. If the error code LL is set to 6 by RØTATE, control returns to ADM4RK; otherwise, control passes to statement 18.

Following statement 18, the integer code MAXCD, the number of values in the CDTAB table, is tested; if MAXCD = 0, SUBROUTINE DRAGCØ is called to compute the drag coefficient before control passes to statement 14. If MAXCD ≠ 0, control passes to statement 13, where CD is determined, using FUNCTION TABLE, by linear interpolation of the input table CDTAB as a function of altitude HTAB, before reaching statement 14. The quantity MAXWCD, the number of WCDTAB values input, is tested after statement 14; if MAXWCD = 0, control passes to statement 15. If MAXWCD ≠ 0, the increment in the drag coefficient WIRECD, is determined from a linear interpolation of input table WCDTAB as a function of WHTAB. This increment is corrected by the ratio of the reference areas and added to the drag coefficient obtained either from DRAGCØ or from the input table, CDTAB.

$$CD = C_D = CD + WIRECD * \frac{AWREF}{AREF}$$

Statement 15 precedes the calling of SUBROUTINE TEQUAT which evaluates the time derivatives of the particle trajectory and thrusting variables, DERIV(1 2, 4) and DERIV(14-16), and the derivative of time with altitude, DERIV(3). The derivative of time with altitude, dt/dz, is used as a multiplier for redefining the 15 time dependent derivatives to make them altitude dependent before the return to ADM4RK where the integration is performed.

5. Other Information

A. SUBROUTINE DEREQ is called in by SUBROUTINE ADM4RK.

B. SUBROUTINE DEREQ calls in the following subprograms:

1. SUBROUTINE PRELIM
2. SUBROUTINE AERØDY
3. SUBROUTINE MASSLØ
4. SUBROUTINE NØSEBL
5. SUBROUTINE TØMALØ
6. SUBROUTINE RØTATE
7. SUBROUTINE DRAGCØ
8. SUBROUTINE TEQUAT
9. FUNCTION TABLE

3.1.4 Preliminary Calculations

The preliminary calculations of aerodynamic coefficients, additional geometric parameters, flow properties in the free stream, and certain edge properties are carried out by SUBROUTINE PRELIM. In performing these computations, PRELIM employs FUNCTION TABLE to do interpolation in certain tables, SUBROUTINE ARFDT2 to call in the calculation of the 1962 Standard Atmosphere of SUBROUTINE COMP62, and SUBROUTINE LNTERP which is used to interpolate in the table of input atmospheric properties.

SUBROUTINE PRELIM(LP)

1. Purpose

SUBROUTINE PRELIM performs preliminary calculations of geometric, flow field, and thrusting parameters for use in calculating derivatives of quantities to be integrated. The subroutine may be divided into the following sections: (1) geometric definitions, (2) aerodynamic coefficients C_{n_α}, C_{m_α}, X_{cp}/D, and C_{m_q}, (3) atmosphere free stream quantities, (4) wind tunnel quantities, (5) thrusting quantities, (6) stagnation and edge properties, (7) definition of transition altitude, (8) definition of specific heats, (9) calculation of $\bar{\chi}$ and $\bar{\chi}_1$, (10) calculation of weight increments.

2. Input

*indicates integer quantity and NØCCUR number code.

Name	Symbol	Occur/Noccur Number	Source of Input	Descript
A, 514	A_i	301-814	2PRS	curve fit con
AE	A_e	214	READIT	thrusting noz
ASØUND	a_∞	-	ARFDT2 or LNTERP	speed of sour
CMQIN	C_{m_q}	196	CHNTBL	input C_{m_q} fo
CØDLAM		094	F123	numerical fa
CØST	cos θ	008	CHNTBL	cosine cone h
CPSZET	cos ψ_s	216	F123	cosine of thr
CTHZET	cos θ_s	215	F123	cosine of thr
DELY	ΔY	219	F123	component of
DELZ	ΔZ	220	F123	component of
DNBNDZ		248	READIT	lower altitud atmosphere
G	g	027	SR2490 or READIT	factor to con
GAMMA	γ	028	SR2490 or READIT	ratio of spec
IATMØS		08 *	READIT	input atmosp
IKCMQ		09 *	READIT	input C_{m_q} o
ITHRST		23 *	READIT	number of va
LA	La	033	PRELIM	instantaneous
LAMDA	λ	032	PRELIM	instantaneous

urce of Input	Description	Value Preset	Unit
ZPRS	curve fit constants		
EADIT	thrusting nozzle exit area		ft^2
RFDT2 or NTERP	speed of sound in atmosphere		ft/sec
HNTBL	input C_{m_q} for current configuration		
123	numerical factor used to calculate CØDRAG		lbm/ft^2
HNTBL	cosine cone half angle		
123	cosine of thrust angular misalignment component		
123	cosine of thrust angular misalignment component		
123	component of linear thrust offset		ft.
123	component of linear thrust offset		ft.
EADIT	lower altitude boundary on use of input atmosphere		ft.
R2490 or EADIT	factor to convert slugs to lbs. mass	32.174	ft/sec^2
R2490 or EADIT	ratio of specific heats	1.4	
EADIT	input atmosphere option code		
EADIT	input C_{m_q} option code		
EADIT	number of values in thrust table		
RELIM	instantaneous axial length		ft.
RELIM	instantaneous bluntness ratio		

B

2. Input (Cont'd)

Name	Symbol	Occur/Noccur Number	Source of Input	Descriptio
LØPT		07 *	SP2490 or READIT	trajectory option c
MAXTAB		04 *	CHNTBL	number of values i current configurat
MINF	M_∞	035	VIXEN	freestream Mach n
NTHRST		22 *	READIT	thrusting option co
PI	π	042	SR2490	mathematical cons
PTØTAL	P_{TOTAL}	245	VIXEN	total pressure in fr
R	R	057	SR2490 or READIT	gas constant
RB	Rb	053	DEREQ	instantaneous base
RESINT	$(\sin\theta)^{-1}$	054	CHNTBL	inverse of sine of c
REYINF	Rey_∞	244	VIXEN	free stream Reynol
RHØ	ρ_∞		LNTERP or ARFDT2	free stream densit
RN	Rn	052	DEREQ	instantaneous nose
SINT	$\sin\theta$	064	CHNTBL	sine of cone half an
SPSZET	$\sin\psi_s$	218	F123	sine of thrust angu
SQCØST	$(\cos\theta)^2$	066	CHNTBL	square of cosine of
STHZET	$\sin\theta_s$	217	F123	sine of thrust angu
TABRHØ, 50	ρ_∞ (TABLE)	3771-3820	READIT	input atmosphere d
TABSND, 50	a_∞ (TABLE)	3821-3870	READIT	input atmosphere s

	Description	Preset Value	Units
or T	trajectory option code		
L	number of values in X_{cg}/D, I, I_x table for current configuration		
	freestream Mach number		
T	thrusting option code		
	mathematical constant		
	total pressure in free stream		lb/ft^2
or T	gas constant	53.5	$\frac{ft\text{-}lb}{lbm\ ^oR}$
	instantaneous base radius		ft.
L	inverse of sine of cone half angle		
	free stream Reynolds number per inch		1/in.
P or 2	free stream density in atmosphere		lbm/ft^3
	instantaneous nose radius		ft.
L	sine of cone half angle		
	sine of thrust angular misalignment component		
L	square of cosine of cone half angle		
	sine of thrust angular misalignment component		
T	input atmosphere density		lbm/ft^3
T	input atmosphere speed of sound		ft/sec

B

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	
TABZ, 50	Z (TABLE)	994-1043	READIT	altitude func current conf
TANT	TAN θ	070	CHNTBL	tangent of c
TBATMZ, 50	Z (TABLE)	3721-3770	READIT	input atmos
TH0	Th_0	207	READIT	reference th
THDELT, 25	Δt (TABLE)	3618-3642	READIT	change in ti of THTH0,
THDELZ, 25	Δ Z (TABLE)	3593-3617	READIT	change in al of THTH0,
THETA	θ	076	CHNTBL	cone half an
THETAD	θ_D	069	CHNTBL	cone half an
THTH0, 25	Th/Th_o (TABLE)	3568-3592	READIT	non-dimensi
TIMER	t	080	VIXEN or DEREQ	time
TØFF	t_{off}	209	READIT	time for thr
TØN	t_{on}	208	READIT	time for thr
TRZTR	$Z_{trans.}$	243	READIT	input transit
TW, 32	Tw_i	2644-2675	CHNTBL or MASSLØ	wall temper
TXCGD, 50	X_{cg}/D (TABLE)	844-893	CHNTBL	X_{cg}/D table
UPBNDZ		247	READIT	upper altitud option
V	V	082	DEREQ	velocity
W	W	084	DEREQ	initial weigh
WTH	W_{TH}	204	DEREQ	initial weigh
Z	Z	091	DEREQ	altitude
ZØFF	Z_{off}	206	READIT	altitude for t
ZØN	Z_{on}	205	READIT	altitude for t

e of t	Description	Preset Value	Units
IT	altitude function for X_{cg}/D, I, I_x table for current configuration		ft.
BL	tangent of cone half angle		
IT	input atmosphere tabular altitude		ft.
IT	reference thrust		
IT	change in time from t_{on} for tabular input of THTH0, $(t - t_{on})$		sec
IT	change in altitude from Z_{on} tabular input of THTH0, $Z_{on} - Z$)		ft.
BL	cone half angle for current configuration		radians
BL	cone half angle for current configuration in degrees		degrees
IT	non-dimensional thrust table		
N or Q	time		sec.
IT	time for thrust shut off		sec.
IT	time for thrust onset		sec.
IT	input transition altitude		ft.
BL or LØ	wall temperature distribution along body		^{o}R
BL	X_{cg}/D table for current configuration		
IT	upper altitude bound for use of input atmosphere option		ft.
Q	velocity		ft/sec
Q	initial weight - $(\Delta \text{weight})_{ablation}$		lb.
Q	initial weight - $(\Delta \text{weight})_{thrust}$		lb.
Q	altitude		ft.
T	altitude for thrust shut off		ft.
T	altitude for thrust onset		ft.

B

3. Output

* indicates integer quantity and NØCCUR number code number

Name	Symbol	Occur/Noccur Number	Descri
AREF	A_{ref}	001	reference ar
CAPL	L	010	sharp cone s
CMALP	C_{m_α}	013	partial deriv
CMQ	C_{m_q}	020	damping in p
CNALP	C_{n_α}	012	partial deriv
CØDRAG		009	numerical fa in rarefied f
CØSLAM	$\lambda \cos \theta$	011	product of bl
CPE	C_{p_e}	017	constant pre cone boundar
CPW	C_{p_w}	018	constant pre
D	D	021	base diamete
DELW	ΔW	097	total weight
DELW2	ΔW_2	226	total weight
DELW3	ΔW_3	227	total weight
HS	H_s	031	stagnation en
HSRT0	H_s/RT_o	029	non-dimensi
HWBAR	$\bar{H}_w$	225	non-dimensi
LA	La	033	instantaneou
LAMDA	λ	032	instantaneou
LP		*	error code

Description	Units
reference area - area of vehicle base	ft^2
sharp cone slant length	ft.
partial derivative of moment coefficient with α	
damping in pitch	
partial derivative of normal force coefficient with α	
numerical factor used in DRAGCØ for drag calculations in rarefied flow	ft.
product of bluntness ratio and cosine of cone half angle	
constant pressure specific heat at edge of the sharp cone boundary layer	$\frac{Btu}{lbm-^oR}$
constant pressure specific heat at wall	$\frac{Btu}{lbm-^oR}$
base diameter	ft.
total weight change	lb.
total weight change due to ablation	lb.
total weight change due to thrusting	lb.
stagnation enthalpy	Btu/lbm
non-dimensional stagnation enthalpy	
non-dimensional wall enthalpy	
instantaneous axial length	ft.
instantaneous bluntness ratio	
error code	

B

3. Output (Cont'd)

Name	Symbol	Occur/Noccur Number	Descr
M	m	037	vehicle m
ME	M_e	036	Mach num
MINF	M_∞	035	Free stre
MUE	μ_e		viscosity
MUINF	μ_∞	034	free strea
MUW	μ_w		viscosity
MX	M_x	210	thrusting
MY	M_y	211	thrusting
MZ	M_z	212	thrusting
PE	P_e	048	pressure
PINF	P_∞	049	free strea
PINFPS	P_∞ / P_s	046	ratio of fr
PS	P_s	047	stagnation
QD	q_d	051	dynamic p
REYL	Rey_{∞_L}	062	free strea slant leng
RHØE	ρ_e	061	density at
RHØIN1	$\rho_{\infty 1}$	056	free strea

Description	Units
vehicle mass	slugs
Mach number at edge of sharp cone boundary layer	
Free stream Mach number	
viscosity at edge of sharp cone boundary layer	$\frac{\text{lbm}}{\text{ft-sec}}$
free stream viscosity	$\frac{\text{lbm}}{\text{ft-sec}}$
viscosity at wall	lbm/ft-sec
thrusting moment about x axis	ft-lb
thrusting moment about y axis	ft-lb
thrusting moment about z axis	ft-lb
pressure at edge of sharp cone boundary layer	lb/ft^2
free stream pressure	lb/ft^2
ratio of free stream pressure to stagnation pressure	
stagnation pressure	lb/ft^2
dynamic pressure	lb/ft^2
free stream Reynolds number based on sharp cone slant length	
density at edge of sharp cone boundary layer	lbm/ft^3
free stream density in lbm/ft^3	lbm/ft^3

B

3. Output (Concl'd)

Name	Symbol	Occur/Noccur Number	Descri
RHØINF	ρ_∞	055	free stream den
S	S	198	total surface dis point to maximu
SINTM	$M_\infty \sin \theta$	067	product of free half angle
TBMAT, 3	T_{x_B}, T_{y_B}, T_{z_B}	3643-3645	components of t system
TE	T_e	079	temperature at
TH	Th	201	total thrusting f
THINF	Th_∞	213	thrusting force i
TINF	T_∞	072	free stream tem
VE	V_e	083	velocity at edge
WTØTAL	W_{TOTAL}	228	instantaneous to
XBAR	$\bar{\chi}$	090	viscous interact
XBAR1	$\bar{\chi}_1$	126	rarefaction para
XBARST	$\bar{\chi}_{st}$	089	numerical facto
XCPD	$X_{c.p.}/D$	088	axial distance fr pressure non-di
ZTR	Z_{tr}	092	transition altitu

Description	Units
free stream density in slug/ft^3	slug/ft^3
total surface distance along vehicle from stagnation point to maximum diameter	ft.
product of free stream Mach number and sine of cone half angle	
components of the thrusting force in body coordinate system	lb.
temperature at edge of sharp cone boundary layer	^oR
total thrusting force	lb.
thrusting force in vacuum	lb.
free stream temperature	^oR
velocity at edge of sharp cone boundary layer	ft/sec
instantaneous total weight of vehicle	lb.
viscous interaction parameter	
rarefaction parameter	
numerical factor used in rarefaction parameter	
axial distance from stagnation point to the center of pressure non-dimensionalized by base diameter	ft.
transition altitude	

B

4. Numerical Procedure

SUBROUTINE PRELIM begins calculations with the geometric definitions of reference area surface distance, base diameter, bluntness ratio, axial length, sharp cone slant length, etc. for the instantaneous configuration. It then proceeds to calculate Newtonian values for C_{n_α} and C_{m_α} using the input tabular value of Xc.g/D. If C_{m_α} is positive an error message is printed, LP is set equal to 6 and control returns to the calling subroutine. If $C_{m_\alpha} \leq 0$, the quantity X_{cp}/D is calculated. A test is made on the input C_{m_q} option code, IKCMQ, which, if greater than zero, causes control to pass to statement 43 where CMQ is set equal to the input quantity. If IKCMQ is zero, CMQ is set equal to the value calculated from Newtonian equations, then control goes to statement 44 circumventing statement 43. Input code LØPT is tested; if equal to 4, control goes to statement 79 for wind tunnel option calculations, otherwise, it continues with the atmosphere calculations.

The input code IATMØS is tested. If IATMØS equals zero the SUBROUTINE ARFDT2 is called in to define the free stream density RHØ, in slugs/ft^3 and speed of sound, ASØUND, in ft/sec. Then the control goes to statement 48. If IATMØS is greater than zero, which means that the input atmosphere option is exercised, control goes to statement 47 and SUBRØUTINE LNTERP is called in. SUBRØUTINE LNTERP defines the quantities RHØ and ASØUND from input atmosphere tables when DNBNDZ < Z ≤ UPBNDZ or by calling in SUBRØUTINE ARFDT2, if Z is outside the range. Following statement 45 there are free stream flow properties defined, then control goes to statement 89, bypassing the wind tunnel calculations.

Following statement 79, the free stream flow properties and the velocity are calculated from the input wind tunnel conditions. Statement 89 follows these calculations and, if LØPT is greater than or equal to 3, which indicates use of input trajectory or input wind tunnel option, control passes to statement 16. If the trajectory is being calculated, LØPT of 0. 1, or 2, the quantity NTHRST, which corresponds to the input quantity NTHRUST, is tested. If NTHRST is equal to zero, the thrusting calculations are bypassed and control passes to statement 16. Otherwise, the directed GØ TØ statement passes control to statement 11 if NTHRST = 1, then thrust is function of altitude, or to statement 12 if NTHRST = 2, then thrust is a function of time.

In statement 11, if the instantaneous altitude, Z, is > ZØN + .0001 ft., control passes to statement 16. If not, Z is tested against ZØFF, where if Z < ZØFF control passes to statement 13; otherwise the free stream thrust is read in from tabular input as a function of the altitude, the printing codes JJTHR and KKTHR are defined, and control passes to statement 14. In statement 12, if the instantaneous time, TIME, < TØN + 1.0D-4, control passes to statement 16. If not, TIME is tested against TØFF, where if TIME $\geq$ TØFF control passes to statement 13, if not, the free stream thrust is read in from tabular input as a function of time, the printing codes JJTHR and KKTHR are defined.

Statement 14 continues the calculating of the thrusting parameters by correcting the thrusting force for back pressure, resolving the thrusting force into components in vehicle body coordinate system,

and calculates the moments it produces before passing control to statement 10.

Statement 13 tests the printing code JJTHR, which if non-zero causes control to pass to statement 16. If JJTHR is zero, which occurs only at the first altitude or time encountered after thrust shutoff altitude or time, the message indicating shutoff time and altitude is printed out, then JJTHR is set equal to 1. The group of equations following statement 16 zeroes out the components of the moment and thrusting force body coordinate system, the thrusting forces, and the printing code KKTHR when the thrusting option is not utilized or the thrust has been shutoff.

Statement 10 is the start of the calculations for stagnation enthalpy, stagnation pressure, free stream dynamic pressure, and sharp cone edge properties. The calculations are straightforward with the exception of the evaluation of Te/T_∞, that is TETINF:

$$\text{If } M_\infty \sin\theta < 5.7 \quad \frac{T_e}{T} = 1. + M_\infty \sin\theta\,(.0966 + .2267\, M_\infty \sin\theta)$$

$$\text{If } M_\infty \sin\theta \geq 5.7 \quad \frac{T_e}{T} = \sum_{KK=0}^{2} \sum_{II=0}^{4} A_{90+II+5KK} \left(M_\infty \sin\theta\right)^{II} \left(\frac{P_\infty}{2116.}\right)^{KK}$$

The quantity TRZTR is then tested and if non-zero, the transition altitude ZTR is set equal to input TRZTR and control skips to statement 45. If TRZTR is equal to zero, control passes to statement 46 where the following equations are used to evaluate ZTR:

$$TEMJ = (L)^{JJ}$$

$$TEMI = \begin{cases} (\theta_D)^{II} & \text{if } \theta_D \leq 15.^\circ \\ (15.)^{II} & \text{if } \theta_D > 15.^\circ \end{cases}$$

$$TEML = \begin{cases} \lambda & \text{if } \lambda \leq 0.3 \\ 0.3 & \text{if } \lambda > 0.3 \end{cases}$$

$$ZTR = \sum_{JJ=0}^{2} \sum_{II=1}^{1} \sum_{II=0}^{1} A_{300 + JJ + 3II + 6KK} \; (TEMJ)\,(TEMI)\,(TEML)^{KK}$$

These calculations are followed by statement 45. Then the flow regime index J is defined as follows:

$Z \geq ZTR$	$J = 2$	laminar flow
$Z > ZTR$	$J = 3$	turbulent flow

These are used to indicate the appropriate value of TW(J, 8), the sharp cone value at maximum diameter point, which is used in the equations for C_{p_w}, $\bar{H}_w$, μ_w, and c_e. The specific heat at constant pressure is then calculated for the edge of boundary layer conditions and for the wall conditions, under the following restrictions:

$$T_e > 5000. \quad C_{p_e} = A_{111} + A_{112}\; T_e$$

$$T_w(J,8) > 5000. \quad C_{p_w} = A_{111} + A_{112}\; T_w(J,8)$$

$$700. \leq T_e \leq 5000. \quad C_{p_e} = \sum_{II=0}^{5} A_{105+II}\,(T_e)^{II}$$

$$700. \leq T_w(J,8) \leq 5000. \quad C_{p_w} = \sum_{II=0}^{5} A_{105+II}\,(T_w(J,8))^{II}$$

$T_e < 700.$ $C_{p_e} = 0.2398$

$T_w(J,8) < 700$ $C_{p_w} = 0.2398$

Next the dimensionless, $\bar{H}_w$ and the viscosity μ_w are defined. These are followed by the evaluation of $\bar{\chi}$ and $\bar{\chi}_1$, the viscous interaction parameter and the rarefaction parameter, respectively.

If LØPT is not equal to 3, the weight increments due to ablation - DELW2, thrust - DELW3, the sum of these - DELW, and the total weight are calculated, then W_{TOTAL} is tested to ensure that it has a positive, nonzero value. If $W_{TOTAL} \leq 0.0$ then calculations are terminated and a warning message printed out. If LØPT equals 3, control skips to statement 91, where for all LØPT M is defined.

5. Other Information

A. SUBRØUTINE PRELIM is called by either SUBRØUTINE VIXEN or SUBRØUTINE DEREQ.

B. SUBRØUTINE PRELIM calls in the following other program subroutines and function

1. SUBRØUTINE ARFDT2
2. SUBRØUTINE LNTERP
3. FUNCTIØN TABLE

C. SUBRØUTINE PRELIM calls in the following library functions:

1. DSQRT
2. FDXPD
3. FDXPI

SUBROUTINE ARFDT2 (H, ASØUND, RHØ)

1. Purpose

SUBROUTINE ARFDT2 calls in the SUBROUTINE CØMP62, which performs calculations to obtain the free stream density and sound speed for the 1962 Standard Atmosphere, and changes the units of the density output of CØMP62.

2. Input

Name	Symbol	Source of Input	Description	Units
ASØUND	a_∞	COMP62	free stream speed of sound	ft/sec.
H	Z	LNTERP or PRELIM	altitude for which free stream conditions to be determined	ft.
P	P_∞	CØMP62	free stream pressure divided by sea level value	dimensionless
RHØ	ρ_∞	CØMP62	free stream density divided by sea level value	dimensionless
T	T_∞	CØMP62	free stream temperature	$^\circ R$

3. Output

ASØUND	a_∞		free stream speed of sound	ft/sec
RHØ	ρ_∞		free stream density	slugs/ft^3

4. Numerical Procedure

SUBROUTINE ARFDT2 sets the quantity RHØ0, ρ_o, equal to 0.0023769 which is the sea level density in slugs per cubic foot. Then SUBROUTINE CØMP62 is called for the altitude H to calculate the free stream properties — a_∞, ρ_∞, P_∞, T_∞, of the 1962 Standard Atmosphere. The non-dimensional free stream density obtained from CØMP62 is multiplied by ρ_o to obtain ρ_∞, RHØ, in units of slugs per cubic foot, before the return to the calling subroutine.

5. Other Information

A. SUBROUTINE ARFDT2 may be called by either SUBROUTINE PRELIM or SUBROUTINE LNTERP.

B. SUBROUTINE ARFDT2 calls in SUBROUTINE CØMP62.

SUBROUTINE CØMP62

1. Purpose

SUBROUTINE COMP62 computes the non-dimensional density and pressure, temperature in degrees Rankine, and the speed of sound in feet per second for the 1962 standard atmosphere.

2. Input

Name	Symbol	Source of Input	Description	Units
H	Z	ARFDT2	altitude	ft.

3. Output

ASØUND	a_∞	sound speed in free stream	ft.
P	P_∞ / P_o	non-dimensional free stream pressure	
RHØ	ρ_∞ / ρ	non-dimensional free stream density	
T	T_∞	free stream ambient temperature	^oR

4. Numerical Procedures

SUBROUTINE COMP62 has the following 26 value data tables stored for use in calculating the atmospheric properties:

TB	molecular scale temperature, $T_M(K)$, in degrees Kelvin
HB	geopotential altitude, $H_b(K)$, in meters
BM	gradient of molecular scale temperature with geopotential altitude, $L'_{M_b}(K)$, degrees Kelvin/meter
R	density, $\rho_b(K)$, in kilograms/cubic meter
PB	pressure parameter, $(10^6 P_b/P_o^2)$ in ft^2/lb, where $P_b(K)$ is atmospheric pressure and P_o is pressure at sea level both in lb/ft^2.
TWTM	molecular weight, M_b, gram/mole

In addition, the following constants are defined:

AG	gravitational constant g_o, 9.80665 m/sec^2
AM0	sea level value of molecular weight, M_o, 28.9644 gm/mole
ARR	universal gas constant, R*, 83.4.32 $gm\text{-}m^2/(^\circ K\ mol\ sec^2)$
AR	radius of earth, R, 6356766. meters
CØN1	conversion from feet to meters, .3048 meters/foot
CØN2	conversion from $^\circ K$ to $^\circ R$, 1.8 $^\circ R/^\circ K$
CØN3	conversion factor for non-dimensionalizing values in table PB, $P_o/10^6$ where P_o is sea level pressure, .0021156 lb/ft^2
CØN4	factor $\sqrt{\gamma R^*/CON2}$ used in calculating speed of sound

The calculations begin with the testing of the input geometric altitude, fortran symbol H. If H exceeds upper altitude limit 2275000.000001 ft., control passes to statement 3 where the quantities RHO, ASOUND, T, and P are set equal to constant values before a return to the calling subroutine is achieved. If H is less than or equal to the limit, control passes to statement 2. The geometric height in meters, ZZ, is computed from H and used to determined AH, the geopotential height. A DØ loop is then utilized to determine the appropriate tabular values to be used in calculations for atmospheric properties. This is accomplished by testing the geopotential altitude AH against the tabular values HB in the following tests:

```
      IF (AH - HB(I)) 7, 8, 9
9     IF (AH - HB(I+1)) 8, 7, 7
```

where I first has a value of 1 and is increased by 1 each time statement 7 is reached. The maximum value I can have is 26. If AH is less than the tabular value HB(I) control goes to statement 7, I is increased by 1, and this test is performed again until AH is either equal to or greater than HB(I). If AH = HB(I) control passes to statement 8 where K is set equal to current value of I and GØ TØ 10 statement is executed. If AH is greater than HB(I), the second IF statement is employed to test AH against the next value in the table HB(I + 1). If AH $\geq$ HB(I+1) statement 7 is utilized and testing continues, but, if AH < HB(I + 1), control passes to statement 8, where K is set equal to I, and is then directed to statement 10.

Statement 10 is the beginning of the calculations for free stream temperature. First the difference between the current geopotential altitude, AH, and the tabular geopotential altitude, HB(I) is determined.

This quantity is utilized in evaluating the following equation for molecular scale temperature, TM in degrees Kelvin,

$$T_M = T_b(K) + L'_{M_b}(K)\ (H - H_b(K))$$

Then the integer KK is set equal to K, but if this gives a KK = 26 KK is reset to 25, for use in the following equation for molecular weight in grams/mole

$$M = M_b(KK) + \frac{H - H_b(KK)}{H_b(KK+1) - H_b(KK)}\ (M_b(KK+1) - M_b(KK))$$

which is used to find the free stream temperature in degrees Rankine

$$T = T_\infty = T_M(CON2)\ \frac{M}{M_o}$$

Statement 12 tests the gradient of molecular temperature L'_M. If L'_M is non-zero, control is directed to statement 15; however, if L'_M is zero, control passes to statement 16.

Statement 15 begins the evaluation of the non-dimensional density and pressure for the non-constant temperature regime using the equations

$$RH\emptyset = \rho_\infty/\rho_o = \frac{\rho_b(K)}{\rho_b(2)} \left[\frac{T_b(K)}{T_M}\right]^{1.0 + \frac{g_o M_o}{R^* L'_{M_b}(K)}}$$

where $\rho_b(2)$ is sea level density, ρ_o.

$$P = \frac{P_\infty}{P_o} = P_b(K) \left[\frac{T_b(K)}{T_M} \right]^{\frac{g_o M_o}{R^* L'_{M_b}(K)}} \qquad \text{(CON3)}$$

Then control passes to statement 17. Statement 16 evaluates the non-dimensional pressure and density for constant temperature region using the following equations

$$P = \frac{P_\infty}{P_o} = P_b(K) \exp \left[- \frac{g_o M_o (H - H_b(K))}{R^* T_{M_b}(K)} \right] \quad \text{CON3}$$

$$RH\emptyset = \rho_\infty / \rho_o = \frac{\rho_b(K)}{\rho_b(2)} \exp \left[- \frac{g_o M_o (H - H_b(K))}{R^* T_{M_b}(K)} \right].$$

Control then passes to statement 17, where the speed of sound, ASØUND in ft/sec. is calculated form the equation

$$ASØUND = CON4 \sqrt{\frac{T}{M}} \text{ or } \sqrt{\frac{R^*}{M} \left(\frac{T}{1.8} \right)}$$

where T is in degrees Rankine.

5. Other Information

A. SUBROUTINE CØMP62 has the restriction that the altitude input, H, must be a positive number. The subroutine uses the first set of equations from "U. S. Standard Atmosphere 1962" with some modifications over the entire range of altitude. The subroutine compares with the standard atmosphere model in the following way:

From 0.0 to 300,000. feet, the subroutine is exact.

From 300,000. to 2.275,000. compares to less than 1/2 of 1%.

Above 2,275,000. the subroutine computes constant value.

B. SUBROUTINE CØMP62 is called by ARFDT2.

C. SUBROUTINE CØMP62 calls in the following library functions:

1. DEXP
2. DSQRT
3. FDXPD

LNTERP (TBATMZ, TABRHØ, TABSND, UPBNDZ, DNBNDZ, Z, RHØ, ASØUND)

1. Purpose

SUBROUTINE LNTERP does a logarithmic interpolation to obtain free stream density and a linear interpolation to obtain the free stream sound speed from the input tables when UPBNDZ $\geq$ Z > DNBNDZ. Outside these limits SUBROUTINE LNTERP calls in SUBROUTINE ARFDT2 to calculate the 1962 Standard Atmosphere values.

2. Input

Name	Symbol	Description	Units
DNBNDZ	Z_{DNBND}	lower altitude boundary on use of tables	ft.
TABRHØ	ρ_∞ (TABLE)	free stream density table must be in ascending order with altitude	lbm/ft^3
TABSND	a_∞ (TABLE)	free stream sound speed table must be in ascending order with altitude	ft/sec
TBATMZ	Z (TABLE)	tabular altitude in ascending order	ft.
UPBNDZ	Z_{UPBND}	upper altitude boundary on use of tables	ft.
Z	Z	altitude for which free stream conditions are desired	ft.

3. Output

Name	Symbol	Description	Units
ASØUND	a_∞	free stream sound speed	ft/sec
RHØ	ρ_∞	free stream density speed	slug/ft^3

4. Numerical Procedure

SUBROUTINE LNTERP begins by testing the altitude Z to determine whether the tabular input atmosphere properties or the calculated 1962 Standard Atmosphere properties are to be used. If Z < UPBNDZ control passes to statement 10, if Z = UPBNDZ to statement 40, nd if Z > UPBNDZ to statement 30. In statement 10, Z is tested against DNBNDZ. If Z ≤ DNBNDZ control passes to statement 30, if Z > DNBNDZ control passes to statement 40. Statement 30 calls SUBROUTINE ARFDT2 to calculate the free stream properties of the 1962 Standard Atmosphere before returning to the calling subroutine.

Statement 40 begins the interpolation of the tabular values with a DØ loop which determines the appropriate tabular points to be used. The DØ loop sets J = I and tests TBATMZ(I) against Z for I from 1 through 50 or until TBATMZ(I) ≥ Z. If no value of TBATM(I) ≥ Z, control calls EXIT. At the point where TBATMZ(I) first is greater than or equal to Z, control passes to statement 46 and the interpolation equations with J equal to the last I.

The evaluation of the interpolation equations is performed following statement 46.

EX = exponent in ρ equation = (Z-TBATMZ(J-1))/(TBATMZ(J)-TBATMZ(J-1))

$$RH\emptyset = \rho_{\infty} = \frac{TABRH\emptyset\,(J-1)}{32.174} * \left[\frac{TABRH\emptyset\,(J)}{TABRH\emptyset\,(J-1)}\right]^{EX}$$

RATI$\emptyset$ = (TABSND(J) - TABSND(J-1))/(TBATMZ(J) - TBATMZ(J-1))

AS$\emptyset$UND = a_{∞} = TABSND(J-1) + (Z - TBATMZ(J-1))*RATI$\emptyset$

Then control returns to the calling subroutine.

5. Other Information

A. SUBROUTINE LNTERP is called in by SUBROUTINE PRELIM.

B. SUBROUTINE LNTERP calls in

1. SUBROUTINE ARFDT2
2. functions EXIT and FDXPD

FUNCTION TABLE (X, XTAB, YTAB, KMAX, L)

1. Purpose

FUNCTION TABLE performs a one dimensional table look up using linear interpolation between table values.

2. Input

* indicates integer quantity

Name		Description
KMAX	*	integer code indicating the maximum number of values in the XTAB, YTAB table
L	*	integer control code
X		the value of the independent variable for which the corresponding dependent variable is desired
XTAB		tabular values of independent variable
YTAB		tabular values of dependent variable

3. Output

Name	Description
TABLE	value of the dependent variable which corresponds to the value of independent variable X.

4. Numerical Procedure

The calculations of FUNCTION TABLE begin with the testing of KMAX. If KMAX, the maximum number of values in the XTAB, YTAB table, is 1, control passes to statement 6 where TABLE is equated to the singular YTAB value. If IKMAX is not equal to 1, then code L is tested. If L is greater than zero, control passes to statement 2 where a linear interpolation is done between the L and (L-1) values of the tabular input to obtain ANS which is set equal to TABLE.

If L is less than or equal to zero, then the subsequent calculations search for the appropriate tabular values to use in the interpolation, regardless of whether the tabular values are in descending or ascending order. This is accomplished by first setting the code IGØ equal to 1, for ascending order, then testing the first tabular value against the second. If this test shows a descending order then IGØ is reset to a value of 2. A DØ loop encompassing the statements through statement 1 is utilized to determine the proper tabular values for use in the linear interpolation of statement 2. The loop is executed for values of K from 2 up until the appropriate $K \leq$ KMAX is reached. Within the loop the value of the quantity IGØ directs control to either statement 3 (IGØ = 1) or statement 4 (IGØ = 2).

In statement 3, if $X >$ XTAB(K) control passes to statement one causing K to be increased by one. Then the procedure is repeated until $X \leq$ XTAB(K), which sends control to statement 5, where L is set equal to that value of K, and control leaves the loop passing to statement 2. In statement 4 a similar procedure is followed if $X <$ XTAB(K) control passes to statement one, but if $X \geq$ XTAB(K) control passes to statement 5 then to statement 2. If K reaches KMAX without

satisfying the appropriate test, then L is set equal to KMAX.

Statement 2 evaluates the equation for the linear interpolation

$$ANS = YTAB(L-1) + \frac{X-XTAB(L-1)}{XTAB(L) - XTAB(L-1)} * (YTAB(L) - YTAB(L-1))$$

then TABLE is set equal to ANS before the return to the calling subroutine.

5. Other Information

A. FUNCTION TABLE is called by SUBROUTINES DEREQ, PRELIM and ROTATE.

B. FUNCTION TABLE calls no other subprograms.

3.1.5 Heating and Mass Loss Calculations

The following section contains the descriptions of the subroutines which perform the heating and mass loss calculations. The cold wall aerodynamic heating distribution along the body is computed by SUBROUTINE AERODY. These results are then used in the determination of the mass loss rate, wall temperature, and surface recession rate distribution by SUBROUTINE EVIL which is called in for each body station by SUBROUTINE MASSLO. TOMALO is then used to integrate the mass loss rate along the body to obtain the rate of change of vehicle weight due to heat-shield ablation. The wall recession rates are then used to determine the time rate of change of nose radius and base radius where cone half angle remains constants in SUBROUTINE NOSEBL.

SUBRØUTINE AERØDY

1. Purpose

SUBRØUTINE AERØDY calculates the ratio of edge pressure to stagnation pressure and the cold wall aerodynamic heating rates for the body stations appropriate to the configuration. For the sharp cone, these stations are the stagnation and maximum diameter points. The blunt cone is represented by the stagnation point, the tangent point, the 20, 40, 60, 75, 90 per cent stations based on initial axial length, the maximum diameter point, and, in turbulent flow only, the sonic point. In addition, the maximum diameter point sharp cone values are evaluated for purposes of comparison with the blunt cone results.

2. Input

Name	Symbol	Occur Number	Source of Input	Desc
A, 514	A_i	301-814	ZPRS	coefficients o
CAPL	L	010	PRELIM	sharp cone sl
FACTR1	-	189	SR2490	numerical fac
HSRT0	h_s/RT_o	029	PRELIM	non-dimensio
LA1	La_1	138	F123	input initial a
LA2	La_2	144	F123	input initial a
LAMDA	λ	032	PRELIM	instantaneous
MINF	M_∞	035	PRELIM	free stream M
PINFPS	P_∞/P_s	046	PRELIM	ratio of free s
PS	P_s	047	PRELIM	stagnation pre
RHØINF	ρ_∞	055	PRELIM	free stream de
RN	Rn	052	CHNTBL	instantaneous
SINT	sin θ	064	CHNTBL	sine of cone ha
TANT	tan θ	070	CHNTBL	tangent of cone
THETA	θ	076	CHNTBL	cone half angle
THETAD	θ_D	069	CHNTBL	cone half angle
V	V	082	DEREQ	velocity
Z	Z	091	DEREQ	altitude
ZTR	Z_{TR}	092	PRELIM	transition altit
ZTURN	Z_{TURN}	145	READIT	altitude at whi

urce of Input	Description	Units
RS	coefficients of curve fit equations	-
ELIM	sharp cone slant length	ft.
2490	numerical factor = (sea level density)$^{0.8}$	(slug/ft^3)$^{0.8}$
ELIM	non-dimensional stagnation enthalpy	-
23	input initial axial length of first configuration	ft.
23	input initial axial length of second configuration	ft.
ELIM	instantaneous bluntness ratio	-
ELIM	free stream Mach number	-
ELIM	ratio of free stream pressure to stagnation pressure	-
ELIM	stagnation pressure	lb/ft^2
ELIM	free stream density in slug/ft^3	slug/ft^3
NTBL	instantaneous nose radius	ft.
NTBL	sine of cone half angle	-
NTBL	tangent of cone half angle	-
NTBL	cone half angle in radians	radians
NTBL	cone half angle in degrees	degrees
REQ	velocity	ft/sec.
REQ	altitude	ft.
ELIM	transition altitude	ft.
ADIT	altitude at which configuration changes	ft.

B

3. Output

Name	Symbol	Occur Number	
PEPSB, 8	$(P_e/P_s)_i$	2801-2808	distributi pressure
QDØT, 32	$\dot{q}_i$	2676-2707	cold wall
XLA, 8	$(X/La)_i$	815-822	body stati

nber	Description	Units
	distribution along body of ratio of vehicle edge pressure to stagnation pressure	-
	cold wall aerodynamic heating along body	$\frac{Btu}{ft^2 - sec}$
	body stations for heating and mass loss calculations	-

B

4. Numerical Procedure

SUBROUTINE AERODY may be divided into three main groupings of equations (1) stagnation point heating, designation of station locations, and evaluation of the distribution of the ratio of edge pressure to stagnation pressure, (2) laminar cold wall heating, (3) turbulent cold wall heating. Each group contains calculations for both sharp and blunt cones.

The indices of the $\dot{q}$ array have the following significance in indicating flow regime and body station:

QDØT(1, 1) = $\dot{q}_{stag}$

QDØT(2, J) = $\dot{q}_{i=J}$ laminar flow

QDØT(3, J) = $\dot{q}_{i=J}$ turbulent flow

QDØT(4, 1) = $\dot{q}_{SONIC}$ turbulent flow

where J = 1 represents the tangent point

J = 2 through 7 correspond to the 20, 40, 60, 75, 90 per cent stations and the max diameter point

J = 8 the maximum diameter point on sharp cone having cone half angle θ.

These station location indices also apply for the pressure distribution and per cent axial distance, $(X/La)_i$.

The stagnation point heating is obtained from

$\lambda < 10^{-3}$ Rs = 1.0

$\lambda \geq 10^{-3}$ Rs = Rn

$$QD\emptyset T(1,1) = \dot{q}_{stag} = 17600. \sqrt{\frac{\rho_\infty}{.002375\ R_s}} \left(\frac{V}{26000.}\right)^{3.15}$$

Using a DO loop the ratio PEPSB(J), $(P_e/P_s)_i$, is zeroed for J = 1, 7. The sharp cone Pe/P_s at the maximum diameter point is computed.

$$PEPSB(8) = \left(\frac{P_e}{P_s}\right)_{i=8} = 0.0331\ e^{0.00640_D - 0.33(M_\infty - 5)^{0.85} + 0.000468\ \theta_D^{1.88032}}$$

and, if $\lambda < 10^{-3}$, control passes to statement 1. If $\lambda \geq 10^{-3}$, the blunt cone station locations are defined for the current configuration and the corresponding pressure distribution evaluated.

$$XLA(1) = (X/La)_1 = R_n(1.0 - \sin\theta)/La$$

$Z > Z_{TURN}$ $\quad DLA = La_1/La$

$Z \leq Z_{TURN}$ $\quad DLA = La_2/La$

$$XLA(2) = 1.0 - 0.8\ DLA = (X/La)_2$$

$$XLA(3) = 1.0 - 0.60\ DLA = (X/La)_3$$

$$XLA(4) = 1.0 - 0.40\ DLA = (X/La)_4$$

$$XLA(5) = 1.0 - 0.25\ DLA = (X/La)_5$$

$$XLA(6) = 1.0 - 0.1\ DLA = (X/La)_6$$

$$XLA(7) = 1.0 = (X/La)_7$$

The geometric quantity TEST is used to determine whether the boundary layer at a given point on the blunt cone has been swallowed, i.e., flow is conical.

$$TEST = \frac{1.13 \quad R_n}{La \ \tan^2 \theta}$$

The calculations for the blunt cone pressure distribution proceed with the evaluation for each index 1 through 7 of the equations nested between the DØ 2 statement and statement 2. Initially, the value of $(X/La)_i$ is tested. If $(X/La)_i$, for $i \neq 1$, is less than or equal to $(X/La)_1$, control passes to statement 21 where the $(X/La)_i$ is set equal to $(X/La)_1$ and the corresponding pressure ratio $(Pe/Ps)_i$ set to 0.0. This test is necessary for cases of severe noseblunting where the nose ablates past the indicated station. For X/La_1 or when $i \neq 1$, $X/La_i > X/La_1$ the X/La_i (for all i) is tested against the parameter TEST. If $(X/La)_i \geq$ TEST, control passes to statement 3, where $(Pe/Ps)_i$ is set equal to the conical pressure ratio $(Pe/Ps)_8$. If $(X/La)_i <$ TEST, the blunt cone pressure ratio is computed as follows:

$$XP = X_P = 0.174 \ \theta_D$$

$$YD = Y_P = 1.0/M_\infty$$

$$ZP = Z_P = 0.2 \ \log_e \left[(X/La)_i \ \frac{La \ \tan^2 \theta \ 10^4}{Rn} \right]$$

$$\theta_D \geq 20.^\circ \quad WP = W_P = \sum_{NN=0}^{1} \sum_{JJ=0}^{2} \sum_{KK=0}^{2} A_{41+NN+2JJ+6KK} \ X_P^{NN} \ Y_P^{JJ} \ Z_P^{KK}$$

$$\theta_D < 20.^\circ \quad WP = W_P = \sum_{NN=0}^{2} \sum_{JJ=0}^{2} \sum_{KK=0}^{2} A_{135+NN+3JJ+9KK} \ X_P^{NN} \ Y_P^{JJ} \ Z_P^{KK}$$

$$PEPSB(I) = (P_e/P_s)_i = W_P \tan^2\theta + \frac{P_\infty}{P_s}$$

The altitude Z is tested against transition altitude Z_{TR}; if $Z < Z_{TR}$, control passes to statement 5, the start of turbulent calculations. If $Z \geq Z_{TR}$, the laminar equations are evaluated.

$$QBL = \bar{Q}_L = 10.0^{0.5142\ \log_e(hs/RT_o)^{0.9736}}$$

$$K1L = K_{1L} = 0.9664 + 0.00528\ \theta_D + .000288\ \theta_D^2$$

$$K2L = K_{2L} = 1.0 + 1.782\ \lambda - 2.008\ \lambda^2$$

$$LAPS = La/P_s$$

If $\lambda < 10^{-3}$, the laminar sharp cone heating is obtained from

$$QD\emptyset T(2,8) = (q_8)_{LAM} = \frac{\bar{Q}_L}{K_{1_L} K_{2_L}} \sqrt{\frac{(Pe/P_s)_8}{(La/P_s)(2116.)(X/La)_8}}$$

before control is returned to the calling subroutine. If $\lambda \geq 10^{-3}$, the heating distribution is

$$QD\emptyset T(2,1) = \dot{q}_{TANG.\ PT._{LAM}} = 1.732\ \dot{q}_{STAG}\left(.00789 + \left(\frac{P_e}{P_s}\right)_1 \left(1.849 + \left(\frac{P_e}{P_s}\right)_1 \left(-1.6832 + 0.841\ \left(\frac{P_e}{P_s}\right)_1\right)\right)\right)$$

and, for I = 2 through 8,

$$QD\emptyset T(2, I) = \dot{q}_i = \frac{\bar{Q}_L}{K_{1L} K_{2L}} \sqrt{\frac{(P_e/P_s)\, i}{(2116.)\,(La/P_s)\,(X/La)_i}}$$

The return statement is then executed.

Statement 5 is the beginning of the turbulent flow heating rate calculations. The reference heating state $\bar{Q}_{TC}$ and the geometric constants K_{1T}, K_{2T}, K_{3T}, and K_{4T} are defined.

$$QBTC = \bar{Q}_{TC} = 10.0^{\,0.745\,(\log_e\, h_s/RT_o)^{\,0.8122}}$$

$$K1T = K_{1T} = 0.9 + 0.02\ \theta_D$$

$$K2T = K_{2T} = 0.6$$

$$K3T = K_{3T} = 0.69 + La\,(0.0318 - 0.00069La)$$

$$K4T = K_{4T} = 1.0$$

If $\lambda > 10^{-3}$, control passes to the blunt cone turbulent calculations beginning at statement 8. If $\lambda \leq 10^{-3}$, the sharp cone maximum diameter point heating rate is computed and control returns to the calling subroutine.

$$TM = P_e/2116.$$

$$QD\emptyset T(3, 8) = (\dot{q}_s)_{TURB} = \bar{Q}_{TC} \left[\left(\frac{P_e}{P_s}\right) TM\right]^{0.8} \Big/ \left[K_{1T}\, K_{2T}\, K_{3T}\, K_{4T} \left(\frac{X}{L_a}\right)_8^{0.165}\right]$$

The blunt cone turbulent heating rates evaluation begins in statement 8 with the computation of sonic point heating. The heating rates for the tangent point and the points on the conical frustum are then computed.

if $Z \leq 115000.$ ft $\quad MB = \bar{m} = 3.45$

if $Z > 115000.$ ft.

$$MB = \bar{m} = 2.2540 + Z(2.246 \times 10^{-5} + Z(-146.9 \times 10^{-12} + 367.1 \times 10^{-18} Z))$$

$$QD\emptyset T(4,1) = \dot{q}_{SONIC\ PT.} = \frac{3760.\ \rho_\infty^{0.8}\ V^{\bar{m}}}{R_s^{0.2}\ FACTR1\ 10.0^{4\bar{m}}}$$

$$FITANG = \phi_{TANG.\ PT.} = \left(\frac{P_e}{P_s}\right)^{\frac{4.1}{6}}_{1} \left[1.0 - \left(\frac{P_e}{P_s}\right)^{1/6}\right]^{0.4}$$

$$QD\emptyset T(3,1) = \dot{q}_{TANG.\ PT\ TURB.} = \frac{1.346\ 10^4}{(R_s(1-\theta))^{0.2}} \left(\frac{\rho_\infty}{2.375 \times 10^{-3}}\right)^{0.8} (.0001V)^{\bar{m}}\ \phi_{TANG.}$$

The denominators related to three categories into which $(X/La)_i$ values might fall are defined in preparation for determination of heating rates on the concial frustum. Here the quantity TEST is redefined.

$$TEST = 1.13\ R_n / (3.0\ La\ Tan^2\ \theta)$$

if $\lambda > 0.2 \quad K2T = K_{2T} = 1.0$

if $\lambda \leq 0.2 \quad K2T = K_{2T} = 0.6 + 2.0\ \lambda$

$$K3T = K_{3T} = 0.69 + La(0.0318 - .00069La)$$

$$K4T = K_{4T} = 1.0$$

$$DEN\emptyset M1 = K_{1T}\ K_{2T}\ K_{3T}\ K_{4T}$$

Then defining the denominator for the second region

$$K3T = K_{3T} = 0.69 + La(0.0318 - 0.00069\ La)$$

$$DEN\emptyset M2 = K_{1T}\ K_{2T}\ K_{3T}$$

and for the third possible range of $(X/La)_i$ values

$$K2T = K_{2T} = 0.6$$

$$K4T = K_{4T} = 1.0$$

$$TEM = La - Rn + Rn/\sin\theta$$

$$K3T = K_{3T} = 0.69 + TEM(0.0318 - 0.00069\ TEM)$$

$$DEN\emptyset M3 = K_{1T}\ K_{2T}\ K_{3T}\ K_{4T}$$

Using a DØ loop which includes all statements through 9, each value for $(X/La)_i$, the non-dimensional station location, is tested against the parameter TEST and the appropriate $\dot{q}_i$ defined according to the category into which $(X/La)_i$ falls. If $(X/La)_i \leq$ TEST, control passes to statement 10. If $(X/La)_i >$ TEST, but $\leq$ (3*TEST), control passes to statement 11. However, if $(X/La)_i >$ TEST and $>$ 3*TEST, the $\dot{q}_i$ is defined as

$$QD\emptyset T(3,I) = (\dot{q}_i)_{TURB} = \bar{Q}_{TC}\left[\left(\frac{P_e}{P_s}\right)_i \frac{P_s}{2116.}\right]^{0.8} / DEN\emptyset M3$$

then control passes to statement 9. In statement 10, the quantity DENØM is set equal to DENØM1, before passing to the definition of $\dot{q}_i$ in statement 12.

$$QD\emptyset T(3,I) = (\dot{q}_i)_{TURB} = \bar{Q}_{TC}\left[\left(\frac{P_e}{P_s}\right)_i \frac{P_s}{2116.}\right]^{0.8} / DEN\emptyset M$$

which is followed by statement 9. Statement 11, defines the following

$$TEM = (X/La)_i \rightarrow TEST$$

$$K4T = K_{4T} = 0.901 + TEM\,(-0.867 + 0.9660\ TEM)$$

$$DEN\emptyset M = DEN\emptyset M2 * K_{4T}$$

before utilized the equation for $\dot{q}_i$ of statement 12. When heating rates for all $(X/La)_i$ have been calculated the return to the calling subroutine is executed.

5. Other Information

A. SUBRØUTINE AERØDY is called by either SUBRØUTINE DEREQ or SUBRØUTINE VIXEN.

B. SUBRØUTINE AERØDY calls in the internal functions

1. DEXP
2. DSQRT
3. DLØG
4. FDXPD
5. FDXPI

SUBROUTINE MASSLØ

1. Purpose

SUBROUTINE MASSLØ controls the calling of SUBROUTINE EVIL. MASSLØ calls in EVIL for each body station required according to the particular shape and flow condition.

2. Input

*indicates integer quantity and NØCCUR number code.

Name	Symbol	Occur/Noccur Number	Source of Input	Desc
C	c	115	READIT or SR2490	stagnation mass loss
LAMDA	λ	032	PRELIM	instantane
QDØT, 32	$\dot{q}_i$	2676-2707	AERØDY	cold wall
SPD	-	068	EVIL	surface re
Z	Z	091	DEREQ	altitude
ZTR	Z_{TR}	092	PRELIM	transition
ZZ	-	-	EVIL	wall temp

3. Output

JJHØLD		01 *		code for b
KKHØLD		02 *		code for b
QDØT, 32	$\dot{q}_i$	2676-2707		cold wall
SDØT, 32	$\dot{s}_i$	2740-2792		surface re
TW, 32	Tw_i	2644-2675		wall temp
TWSTAG	Tw_{STAG}	073		stagnation

Description	Units
stagnation point heating multiplier for changing mass loss effects on nose	
instantaneous bluntness ratio	
cold wall aerodynamic heating at specified stations	$\frac{Btu}{ft^2\text{-sec}}$
surface recession rate calculated in subroutine EVIL	ft/sec
altitude	ft.
transition altitude	ft.
wall temperature calculated by subroutine EVIL	^oR
code for body position, flow condition	
code for body position, flow condition	
cold wall aerodynamic heating at specified stations	$\frac{Btu}{ft^2\text{-sec}}$
surface recession rate at specified stations	ft/sec
wall temperature at specified stations	^oR
stagnation point wall temperature	^oR

B

4. Numerical Procedures

This subroutine employs ASSIGN statements and an ASSIGNED GØ TØ statement to control the flow of the subroutine. The indices J and K indicate the flow regime and body station as follows:

J = 1 K = 1	stagnation point for all Z
J = 2 K = 1, 7	blunt cone laminar conical frustum points
J = 2 K = 8	sharp cone laminar maximum diameter point
J = 3 K = 1, 7	blunt cone turbulent conical frustum point
J = 3 K = 8	sharp cone turbulent maximum diameter point
J = 4 K = 1	turbulent blunt cone sonic point

Initially J and K are both set equal to 1, value 2 is assigned to NEXT (used in ASSIGNED GO TO statement), $\dot{q}(1, 1)$ is set equal to $\dot{q}(1, 1)$*C (adds effect of stagnation point multiplier), then control is directed to statement 100.

Statement 100 is the beginning of the mass loss rate calculation. Here JJHØLD is set equal to J and KKHØLD is set equal to K, since these are designations used in common. Then SUBROUTINE EVIL is called in for the current value of J and K to calculate the mass loss rate $\dot{m}(J, K)$, wall temperature ZZ, and surface recession rate SPD for each station required. $T_w(J, K)$ is set equal to ZZ and $\dot{S}(J, K)$ to SPD. Then the ASSIGNED GØ TØ statement - GØ TØ NEXT, (2, 200, 20, 6, 30) - causes the subroutine control to be directed to the statement number having the the same value as NEXT. When the flow is directed to statement 200, the return to the calling subroutine is accomplished.

After performing, the calculations for J = 1, K = 1 the subroutine continues to statement 2, $T_{w_{STAG}}$ is set equal to $T_w(1, 1)$ and $\dot{q}(1, 1) = \dot{q}(1, 1)/C$. Then Z is tested against ZTR to determine whether fluid flow is laminar or turbulent. If Z < ZTR, the laminar calculations are circumvented by skipping to statement 3, the beginning of turbulent calculations.

If Z ≥ ZTR, the calculations proceed to the testing of bluntness ratio, LAMDA. If LAMDA is less than or equal to 10^{-3}, then J = 2 and K = 8, 200 is assigned to next, the subroutine proceeds to statement 100 and does the laminar sharp cone calculations.

The assigned GØ TØ statement goes to the RETURN statement 200. If LAMDA > 10^{-3}, the control goes to statement 4 begins the laminar blunt cone calculations. In statement 4, J is set equal to 2. K is initialized to 0. In statement 20, K is set equal to K + 1 and in the next statement tested to see if the maximum allowable value has been reached. If K equals 9, control passes to RETURN statement 200. If K is less than 9, value 20 is assigned to NEXT and the GØ TØ 100 statement exercised. The mass loss calculations are performed and subroutine flow returns to statement 20 where value of K is increased by 1 causing mass loss calculation to be performed for the next body station. This sequence is repeated until the calculations for all the blunt cone stations and sharp cone maximum diameter point have been performed at which point K = 9.

The turbulent flow calculations proceed in an analagous manner with the addition of a sonic point calculation performed for the blunt cone. In statement 3, LAMDA is tested and if less than or equal to 10^{-3} proceeds with the sharp cone calculations. J is set equal to 3 and K to 8, 200 is assigned to NEXT, the mass loss calculations are called in, then the return to the calling subroutine is executed. If LAMDA

is greater than 10^{-3}, subroutine flow proceeds to statement 5, which begins the turbulent blunt cone calculations with the sonic point calculation. J is set equal to 4 and K to 1, 6 is assigned to NEXT, mass loss calculations are called in, after which the ASSIGNED GØ TØ statement causes flow of subroutine to go to statement 6. Statement 6 is the beginning of turbulent calculations for points on conical frustum. J is set equal to 3; K to 0; then, in statement 30, K is set equal to K + 1; if K is less than 9, 30 is assigned to NEXT and mass loss calculations called in. The ASSIGNED GØ TØ statement causes, the return to statement 30, where value of K is increased by one causing a repetition of the procedure until the value of K is 9, indicating that calculations for all the stations on the blunt cone and for sharp cone maximum diameter point have been performed.

5. Other Information

A. SUBRØUTINE MASSLØ calls in SUBRØUTINE EVIL.

B. SUBRØUTINE MASSLØ is called in by

1. SUBRØUTINE VIXEN
2. SUBRØUTINE DEREQ

SUBROUTINE EVIL (ZZ)

1. Purpose

SUBROUTINE EVIL is called by MASSLØ to calculate the mass loss rate, $\dot{m}_i$, the wall recession rate $\dot{S}_i$, and the wall temperature, T_{W_i}, for each of the body stations for which aerodynamic heating rates were obtained in SUBROUTINE AERØDY. The method of calculation is a function of the material being considered. For OTWR in turbulent flow, as well as LT_a, teflon, and the input material in both laminar and turbulent flow, an iterative steady state solution is used. OTWR in laminar flow, carbon phenolic, and phenolic nylon employ curve fits of $\dot{m}$ and $\dot{S}$ as functions of the cold wall aerodynamic heating, $\dot{q}$, from SUBROUTINE AERØDY, and T_W as a function of $\dot{S}$.

2. Input

Name	Symbol	Occur/Noccur Number	Source of Input	Des
BETA1	β_1	004	CHNTBL	sublimation r
BETA2	β_2	005	CHNTBL	sublimation r
BETA3	β_3	006	CHNTBL	order of react
BETA4	β_4	007	CHNTBL	activation tem
CP2	C_{P_2}	014	CHNTBL	specific heat o
DELRHO	$\Delta\rho$	022	CHNTBL	difference betw
EPSIL	ϵ	024	CHNTBL	coefficient of e
F	F	025	CHNTBL	heat of ablation
FACTR2	$\ln(1.1105 \times 10^7)$	190	CHNTBL	numerical fact
FACTR3	12.7657 + 2.9984	191	CHNTBL	numerical fact
FACTR4	$\rho C_{P_2} + \Delta\rho C_{P_g}$	192	CHNTBL	numerical fact
FACTR5	$\Delta\rho \Delta H_C$	193	CHNTBL	numerical fact
FACTR6	$\rho_2 \eta_{SL} + \Delta\rho \eta_{GL}$	194	CHNTBL	numerical fact
FACTR7	$\rho_2 \eta_{ST} + \Delta\rho \eta_{GT}$	195	CHNTBL	numerical fact
HREF	h_{REF}	030	CHNTBL	numerical fact
HS	h_S	031	PRELIM	stagnation enth
HSRT0	H_S/RT_o	029	PRELIM	non-dimensiona
JJHØLD		01 *	MASSLØ	code for body p

Source of Input	Description	Units
INTBL	sublimation rate coefficient	$\frac{ft}{sec\ ^{\circ}R}$
INTBL	sublimation rate coefficient	$\frac{ft}{sec\ ^{\circ}R^{\frac{1}{2}}}$
INTBL	order of reaction	-
INTBL	activation temperature	$^{\circ}R$
INTBL	specific heat of solid	$\frac{Btu}{lbm-^{\circ}R}$
INTBL	difference between virgin and char density	lbm/ft^3
INTBL	coefficient of emission	-
INTBL	heat of ablation	Btu/lbm
INTBL	numerical factor used in iterative solution	-
INTBL	numerical factor used in iterative solution	-
INTBL	numerical factor used in iterative solution	$\frac{Btu}{ft^3\ ^{\circ}R}$
INTBL	numerical factor used in iterative solution	$\frac{Btu}{ft^3}$
INTBL	numerical factor used in iterative solution	lbm/ft^3
INTBL	numerical factor used in iterative solution	lbm/ft^3
INTBL	numerical factor used in iterative solution	Btu/lbm
ELIM	stagnation enthalpy	Btu/lbm
ELIM	non-dimensional stagnation enthalpy	-
SSL∅	code for body position, flow condition	-

B

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	De
KKHØLD		02 *	MASSLØ	code for body
MATLNØ		13 *	READIT or SR2490	material optio
NST	η_{ST}	040	CHNTBL	turbulent trans
PS	P_S	047	PRELIM	stagnation pre
QDØT, 32	$\dot{q}_i$	2676-2707	AERØDY or MASSLØ	cold wall aero along the body
RHØ2	ρ_2	058	CHNTBL	char density
TINIT	T_{INIT}	132	READIT or SR2490	initial internal
TW, 32	T_{W_i}	2644-2675	CHNTBL or MASSLØ	wall temperatu
TW0	T_{W_0}	074	CHNTBL	initial wall ten
Z	Z	091	DEREQ	altitude
ZTR	Z_{TR}	092	PRELIM	transition altit

3. Output

HWBAR	$\bar{H}_W$	225		non-dimension
MDØT, 32	$\dot{m}_i$	2708-2739		mass loss rate
SPD		068		wall recession EVIL is perfor
ZZ		-		wall temperatu is performing

Description	Units
code for body position, flow condition	-
material option code	-
turbulent transpiration factor of solid	-
stagnation pressure	lb/ft^2
cold wall aerodynamic heating distribution along the body	$\frac{Btu}{ft^2\ sec}$
char density	lbm/ft^3
initial internal body temperature	^oR
wall temperature distribution along body	^oR
initial wall temperature for configuration	^oR
altitude	ft.
transition altitude	ft.
non-dimensional wall enthalpy	
mass loss rate distribution along vehicle	lbm/ft^2-sec
wall recession rate at point on body for which EVIL is performing calculations	ft/sec
wall temperature at point on body for which EVIL is performing calculations	^oR

B

4. Numerical Procedure

The equations of SUBROUTINE EVIL may be grouped into four main sections: (1) steady state ablation iterative solution, (2) laminar 0TWR calculations, (3) phenolic nylon relations, (4) carbon phenolic equations. The steady state ablation method which entails the iterative solution of simultaneous equations for the surface temperature and the wall recession rate, includes the following energy considerations - convective energy, conduction flux, surface radiation loss, and sublimation energy. The results for surface temperature and wall recession rate are then employed to determine the mass loss rate. For 0TWR and carbon phenolic in laminar flow, and phenolic nylon in laminar and turbulent flow, curve fits of $\dot{m}$ and $\dot{S}$ as functions of the cold wall aerodynamic heating and T_W as a function of $\dot{S}$ are utilized. The turbulent carbon phenolic $\dot{m}$ is obtained from the conduction flux, the radiation loss, and a calculated q*; $\dot{S}$ is a function of $\dot{m}$, and T_W a function of $\dot{S}$.

SUBROUTINE EVIL calculations begin by setting the quantities J and K equal to JJHØLD and KKHØLD, respectively, and ICØUNT initialized to zero. Then, if J is equal to 2 (value of 2 indicating laminar flow for points of conical frustum), the turbulent wall temperature for the point on the conical frustum being considered, TW(3, K), is set equal to TW(2, K) laminar wall temperature at that point and the turbulent sonic point wall temperature TW(4, 1) is set equal to laminar tangent point value, TW(2, 1). This is done in order to start the turbulent calculations with a reasonable wall temperature when the laminar calculations precede them. If the program is started in turbulent flow, the input initial wall temperature is used as a first value. If J is not equal to 2, the pre-setting

of the turbulent wall temperatures is unnecessary and the equations are bypassed.

Next, the material number code, MATLNØ, is tested in order to direct control the the appropriate method of calculation. If MATLNØ < 3, which indicates use of either teflon for MATLNØ = 1, or $L_{T_{ad}}$ for MATLNØ = 2, control passes to statement 14. If MATLNØ $\geq$ 3, the quantity is tested to see if it has the value 4. If MATLNØ = 4 for phenolic nylon, control passes to statement 52. If MATLNØ $\geq$ 3, but $\neq$ 4 (that is, it has value of either 3, 5, or 6), it is again tested. Now, if MATLNØ = 6 for an input material, control passes to statement 14. If MATLNØ equals either 3 for ØTWR or 5 for carbon phenolic, the altitude is tested to determine whether the flow is laminar, Z > ZTR, in which case control passes to statement 52, or turbulent, Z $\leq$ ZTR, in which case control proceeds to statement 14.

Statement 14 is the beginning of the iterative solution of the steady state ablation model. The wall temperature symbol used in the iteration, ZZ, is set equal to TW(J, K) which at time = 0 is equal to TWØ and at time > 0 is equal to the previous solution for the value of wall temperature at the point being considered. The quantity DS, the increment added to wall temperature for each iteration, is defined as

$$DS = T_{W_i} = 0.005\ T_{W_i} + \frac{25000.0}{T_{W_i}}$$

Statement 1, which is the beginning of the iterative loop, increases the value of ICØUNT by one. The wall recession rate SPD is defined as

$$SPD = \dot{S}_1 = \beta_1 ZZ + \exp\left[\log(\beta_2 ZZ^{\beta_3}) - \frac{\beta_4}{ZZ}\right]$$

The numerical factors necessary to calculate $\bar{H}_W$ are determined

$$TK1 = FACTR2 - 1.1112 \times 10^5/ZZ$$

$$K1 = K_1 = EXP(TK1)$$

$$\text{if } K1 < 1.0 \times 10^{-36}, \quad K1 = 0.0$$

$$PBAR = \bar{P} = P_e/2117.$$

$$K2 = \left[12.7659\, K_1 * K_1 + 4.\, K_1 * 2.9984 (K_1 + 4.\, \bar{P})\, FACTR3\right]^{1/2} - 12.7657\, K_1$$

$$K2 = K_2 = 0.5\ K2 / (K_1 + 4.0\bar{P})$$

$$\Gamma_1 = 13.654 + 469.585\, K_2$$

$$\Gamma_2 = 0.256012 + .5558\ 10^{-2} K_2$$

$$\Gamma_3 = 0.5345\ 10^{-5} - 0.\ 427\ 10^{-6} K_2$$

and using these

$$HW = \bar{H}_W = \left[\Gamma_1 + ZZ(\Gamma_2 + \Gamma_3\ ZZ)\right]/33.86.$$

The following coefficients are defined for use in determining the conduction flux

$$E1BAR = \bar{E}1 = \bar{H}_W/35.89$$

$$XCØM = X_{COM} = 0.349\ h_e/RT_o$$

$$E2BAR = \bar{E}2 = (X_{COM} + 0.5\ \bar{H}_W) / (X_{COM} + 17.945)$$

$$E3BAR = \bar{E}3 = 0.95 - (\bar{H}_W - H_{ref}) / (h_e/RT_o)$$

Then, if JJHOLD = 2 which indicates a point on the conical frustum in laminar flow, control passes to statement 2, there E1 = 0.0 and E2 = -0.185 are defined before passing to statement 4. If JJHØLD ≠ 2, but is ≥ 3 indicating either turbulent sonic point or turbulent conical frustum point, control passes to statement 3, there E1 and E2 are respectively set equal to 0.0 and -0.502 before control passes to statement 4. If JJHOLD satisfies neither condition, indicating a value of 1 for stagnation point, then E1 = - 0.037 and E2 = 0.0 and control passes to statement 4.

Statement 4 begins the evaluation of the terms of the simultaneous equations.

$$QD\emptyset T\emptyset F = \dot{q}_{0F} = \dot{q}\,(JJH\emptyset LD,\ KKH\emptyset LD)\ \overline{E1}^{E1}\ \ \overline{E2}^{E2}$$

If $\dot{q}_{0F} \leq 0.0$ control passes to statement 77, otherwise the following quantities are defined

$$TEM = ZZ - T_{INIT}$$

$$Z \geq Z_{TR} \quad FBAR = \bar{f} = SPD\ h_s\ FACTR6/\dot{q}_{0F}$$

$$Z < Z_{TR} \quad FBAR = \bar{f} = SPD\ h_s\ FACTR7/\dot{q}_{0F}$$

$$PHIB = \bar{\phi} = \exp\left[-\bar{f}\,(1.0 + 0.618\,\bar{f})\right]$$

The surface radiation term of heat flux is

$$QD\emptyset TT = \dot{q}_r = 0.47583\ 10^{-12}\,\epsilon\,(ZZ)^4$$

The conduction flux term is

$$QD\emptyset TC = \dot{q}_c = \dot{q}_{0F}\ \overline{E3}\ \bar{\phi}$$

If MATLNØ = 5, at this point control passes to statement 55, the beginning of the turbulent carbon phenolic equations. Otherwise, iterative solution is continued with the definition of the sublimation energy

$$QD\emptyset TS = \dot{q}_s = SPD\, \rho_2\, F$$

and the convective energy

$$LBAR = \bar{L} = \dot{q}_K = SPD(TEM*FACTR4 + FACTR5).$$

In solving the simultaneous equations, the convective energy must be balanced against the following sum of the other energy terms

$$RBAR = \bar{R} = \dot{q}_c - \dot{q}_r - \dot{q}_s$$

The quantity ARGU which must be zero or minimized to obtain a solution is

$$ARGU = \bar{L} - \bar{R}.$$

The quantity ICOUNT is tested; if it is > 100, indicating 100 passes through the evaluation of the equations, the last value of ZZ is taken as the solution as control passes to statement 50. If ICØUNT < 100, the quantity ARGU is tested — if $|ARGU| < 1.0$, control passes to statement 50 or, if $|ARGU| \geq 1.0$, but $< |0.01\, \bar{L}|$, control passes tc statement 50 and the current value of ZZ is taken as the solution. If $|ARGU| \geq 1.0$ and $\geq |0.01\, \bar{L}|$, ARGU is again tested— if < 0.0, control passes to statement 8; if = 0.0, control passes to statement 50; if > 0.0, control passes to statement 9. Statement 8, for ARGU < 0.0, tests the increment in wall temperature DS: 1) if DS < 0.0, control passes to statement 11 where DS is set equal to -0.5 DS, a positive

number, which causes an increase in wall temperature ZZ in statement 10 where ZZ = ZZ + DS, 2) if DS = 0.0, control passes to statement 50, 3) if DS > 0.0, control passes to statement 10 where wall temperature ZZ is increased by DS, i.e., ZZ = ZZ + DS. Statement 9, for ARGU > 0.0, tests the wall temperature increment: 1) if DS < 0.0, control passes to statement 10 where wall temperature ZZ is decreased when ZZ is set equal to ZZ = ZZ + DS, 2) if DS = 0.0, control passes to statement 50, 3) if DS > 0.0 control passes to statement 11 where DS is set equal to -0.5DS, a negative number, which causes a decrease in wall temperature in statement 10, ZZ = ZZ + DS. After passing through statement 10, for all cases where DS ≠ 0.0, control passes to statement 1 where ICØUNT is increased by 1 and the procedure repeated with the new value of ZZ.

In statement 50, ZZ is set equal to whichever value is larger of ZZ or 0.0, i.e., ZZ cannot be negative. In statement 77, if $\dot{q}_{0F} \leq 0.0$, SPD is set equal to 0.0. If $\dot{q}_{0F} > 0.0$, SPD is set equal to either SPD or 0.0, whichever is the larger of the two (cannot have a negative SPD). Then, using the solution for SPD and ZZ, the mass loss rate is defined

$$MDØT(J, K) = \dot{m} = SPD\ (\rho_2 + \Delta\rho)$$

before the return to the calling subroutine is executed.

Statement 52, which is reached when the MATLNØ = 3 or 5 in laminar flow or when MATLNØ = 4 in laminar and turbulent flow, passes control to statement 54 when MATLNØ = 5. Otherwise, it calculates the $\dot{m}$, SPD, and ZZ for OTWR in laminar flow if MATLNØ = 3, or the laminar

OTWR $\dot{m}$ and SPD for use in phenolic nylon equations when MATLNØ = 4. Defining the quantity

$$QD\emptyset TCW = \dot{q}_{CW} = QD\emptyset T\,(JJH\emptyset LD,\ KKH\emptyset LD)$$

to be used in the evaluation of $\dot{m}$ and SPD.

The value of $\dot{q}_{CW}$ is tested and accordingly the appropriate equation is utilized to evaluate $\dot{m}(J, K)$, after which control passes to statement 57.

$\dot{q}_{CW} < 13.0 \qquad \dot{m}(J, K) = 0.0$

$13.0 \leq \dot{q}_{CW} < 15.6 \qquad \dot{m}(J, K) = .00021 + .00015\,(\dot{q}_{CW} - 13.0)$

$15.6 \leq \dot{q}_{CW} < 250. \qquad \dot{m}(J, K) = -1.27424339 \times 10^{-3} + 1.36071670 \times 10^{-4}\,\dot{q}_{CW}$
$- 1.09091516 \times 10^{-6}\,\dot{q}_{CW}^{2} + 7.98275747 \times 10^{-9}\,\dot{q}_{CW}^{3}$
$- 1.65210579 \times 10^{-11}\,\dot{q}_{CW}^{4}$

$250. \leq \dot{q}_{CW} < 3000. \qquad \dot{m}(J, K) = -1.05650025 \times 10^{-3} + 7.61118699 \times 10^{-5}\,\dot{q}_{CW}$
$+ 3.34251700 \times 10^{-8}\,\dot{q}_{CW}^{2} - 6.91682422 \times 10^{-12}\,\dot{q}_{CW}^{3}$

$\dot{q}_{CW} > 3000. \qquad \dot{m}(J, K) = \dot{q}_{CW}(1.0 - 1500./h_s)/(3640. + 8.1\,h_s/RT_o)$

In statement 57, the quantity MØTWR ($\dot{m}_{OTWR}$) is set equal to $\dot{m}(J, K)$, then $\dot{q}_{CW}$ is tested in order to direct control to the appropriate equation for SPD after which control is directed to statement 73.

$$\dot{q}_{CW} < 1000.0 \quad SPD = 1.6115 \times 10^{-6}\, \dot{q}_{CW} - 5.23741 \times 10^{-4}$$

$$1000. \leq \dot{q}_{CW} < 3000. \quad SPD = -1.11196760 \times 10^{-4} + 4.03376719 \times 10^{-7} \dot{q}_{CW} + 9.70131261 \times 10^{-10}\, \dot{q}_{CW}^{2} - 2.45527504 \times 10^{-13}\, \dot{q}_{CW}^{3}$$

$$\dot{q}_{CW} \geq 3000. \quad SPD = \dot{m}_{OTWR} / (\rho_2 + \Delta\rho)$$

In statement 73, if $SPD < 1.0 \times 10^{-13}$, then SPD is set equal to 1.0×10^{-13}. The quantity SØTWR ($\dot{S}_{OTWR}$) is set equal to SPD, then if the material number code, MATLNØ, is equal to 4 (phenolic nylon), control passes to statement 53. If MATLNØ ≠ 4, the SPDL is defined as $\log_{10}$ SPD for use in the equation for wall temperature. The lowest allowable value for SPDL is -12.583, if a smaller value for SPDL results, then SPDL is set equal to the limiting value.

$$ZZ = 6.34634912 \times 10^{3} + 5.50628796 \times 10^{2}\, SPDL + 1.96585366 \times 10^{1} (SPDL)^{2}$$

If ZZ > 4850., ZZ is set equal to 4850.0 before the return to the calling subroutine.

Statement 53 is the beginning of the phenolic nylon $\dot{m}$ calculations. The quantity $\dot{q}_{CW}$ is tested and control directed to the appropriate $\dot{m}$ equation after which control passes to statement 61.

$$\dot{q}_{CW} \leq 100.0 \quad \dot{m}(J,K) = 10.0^{(-2.52288 + 7.3759 \times 10^{-3}\, \dot{q}_{CW})}$$

$100. < \dot{q}_{CW} \leq 3000.$ $\quad \dot{m}(J, K) = -1.62367642 \times 10^{-3}$

$+ 1.78922793 \times 10^{-4}\ \dot{q}_{CW} + 1.32113696 \times 10^{-8}\ \dot{q}_{CW}^{2}$

$- 5.08747513 \times 10^{-12}\ \dot{q}_{CW}^{3}$

$\dot{q}_{CW} > 3000.$ $\quad \dot{m}(J, K) = \dot{q}_{CW}(1.0 - 1700.0/h_s)\ /$

$(1845. + 11.1\ h_s/RT_o)$

If $\dot{m}(J, K) < 1.0 \times 10^{-4}$, then $\dot{m}(J, K)$ is set equal to 1.0×10^{-4}. If the previously defined $\dot{m}_{OTWR} < 1.0 \times 10^{-5}$, then $\dot{m}_{OTWR}$ is set equal to 1.0×10^{-5}. The wall recession rate for phenolic nylon is determined as a ratio of the $\dot{S}$ for OTWR from the following:

$$SPD = \frac{\dot{m}(J, K)}{\dot{m}_{OTWR}}\ \dot{S}_{OTWR}.$$

If $SPD < 1.0 \times 10^{-5}$, then SPD is set equal to 1.0×10^{-5}. The quantity SPDL is defined as the $\log_{10}$ SPD for use in determining the wall temperature

$SPD < 1.0 \times 10^{-4}$ $\quad ZZ = 250.0 + 1756.0\ (SPDL + 5.0)$

$SPD \geq 1.0 \times 10^{-4}$ $\quad ZZ = 6.94474035 \times 10^{+3} + 6.45367146 \times 10^{+2}\ SPDL$

$- 1.48589173 \times 10^{+2}\ (SPDL)^2$

If $ZZ > 4850$, then ZZ is set equal to 4850.0 before the return to the calling subroutine.

Statement 54 marks the beginning of the carbon phenolic calculations in laminar flow. The quantity QDØTCW ($\dot{q}_{CW}$) is set equal to QDØT(JJHØLD, KKHØLD) and tested to determine the appropriate $\dot{m}$ in equation. After the evaluation of $\dot{m}(J, K)$ control passes to statement 68.

$\dot{q}_{CW} < 8.0$ $\qquad \dot{m}(J, K) = 0.0$

$8.0 \leq \dot{q}_{CW} < 11.3$ $\qquad \dot{m}(J, K) = 6.5788 \times 10^{-4} - 1.59773 \times 10^{-4}\ \dot{q}_{CW} + 9.8485 \times 10^{-6}\ \dot{q}_{CW}^2$

$11.3 \leq \dot{q}_{CW} < 23.1$ $\qquad \dot{m}(J, K) = 1.1 \times 10^{-4} + 9.78022 \times 10^{-5} (\dot{q}_{CW} - 11.3)$

$23.1 \leq \dot{q}_{CW} < 1000$ $\qquad \dot{m}(J, K) = -1.73043255 \times 10^{-3} + 1.25965766 \times 10^{-4}\ \dot{q}_{CW} - 8.56939032 \times 10^{-8}\ \dot{q}_{CW}^2 + 4.65548312 \times 10^{-12}\ \dot{q}_{CW}^3$

Statement 68 is the beginning of the calculation for SPD; $\dot{q}_{CW}$ is tested to determine and evaluate the appropriate SPD equation before control passes to statement 69.

$\dot{q}_{CW} < 24.7$ $\qquad SPD = 1.0 \times 10^{-13}$

$24.7 \leq \dot{q}_{CW} < 275.$ $\qquad SPD = -2.212296558 \times 10^{-6} + 1.18955951 \times 10^{-7}\ \dot{q}_{CW} + 1.40098706 \times 10^{-10}\ \dot{q}_{CW}^2$

$$275.0 \leq \dot{q}_{CW} < 1000. \quad SPD = 7.66757321 \times 10^{-5} - 4.95033115 \times 10^{-7}\ \dot{q}_{CW} + 1.55581653 \times 10^{-9}\ \dot{q}_{CW}^{2} - 8.49239390 \times 10^{-13}\ \dot{q}_{CW}^{3}$$

$$\dot{q}_{CW} \geq 1000. \quad SPD = -3.186484 \times 10^{-4} + 3.869 \times 10^{-7}\ \dot{q}_{CW} + 2.171792 \times 10^{-10}\ \dot{q}_{CW}^{2}$$

Following statement 69, SPDL is defined as $\log_{10}$ SPD for use in evaluating the wall temperature

$$ZZ = 6.68741134 \times 10^{+3} + 4.46431845 \times 10^{+2}\ SPDL + 1.20991623 \times 10^{1}\ (SPDL)^{2}$$

If ZZ > 6160.0, then ZZ is set equal to 6160.0 before control returns to calling subroutine.

Statement 55 is the beginning of the mass loss rate, wall recession rate, and wall temperature calculations for carbon phenolic material which are dependent on some of the quantities calculated in the iterative solution.

QDNET = $\dot{q}_{net} = \dot{q}_c - \dot{q}_r$

ABAR = $\bar{A} = C_{P_2}\ (6160. - T_{W_0}) + F$

BBAR = $\bar{B} = \eta_{ST}$

QSTAR = $q^* = \bar{A} + \bar{B}\ hs/RT_o$

$in(J, K) = \dot{q}_{net} / q^*$

If $\dot{q}$ net < 0.0, $\dot{m}(J, K) = 0.0$

$$SPD = 0.8\ \dot{m}(J, K)\ /\rho_2$$

If $SPD \leq 1.0 \times 10^{-10}$, $SPD = 1.0 \times 10^{-10}$

$$SPD = \log_{10} SPD$$

If SPDL < -18.058, then SPDL is set equal to -18.058.

The wall temperature is then calculated from

$$ZZ = 6.68741134 \times 10^{+3} + 4.46431845 \times 10^{2}\ SPDL + 1.20991623 \times 10^{1}\ SPDL^{2}$$

before the return to the calling subroutine is accomplished.

5. Other Information

A. SUBROUTINE EVIL is called by SUBROUTINE MASSLØ.

B. SUBROUTINE EVIL calls in the following functions:

1. DEXP
2. DSQRT
3. DLØG
4. DLØG10
5. FDXPD

SUBRØUTINE TØMALØ

1. Purpose

SUBRØUTINE TØMALØ integrates the mass loss rates along the body to obtain the time rate of change in vehicle weight attributable to ablation effects. Both the laminar and the turbulent equations for the sharp cone are a result of integrating analytically. On the other hand, the integration of laminar and turbulent blunt cone mass loss is carried out by a trapezoidal rule integration.

2. Input

Name	Symbol	Occur/Noccur Number	Source of Input	Desc
CØST	cos θ	008	CHNTBL	cosine of c
LA	La	033	PRELIM	instantaneo
LAMDA	λ	032	PRELIM	instantaneo
MDØT, 32	$\dot{m}_i$	2708-2739	EVIL	matrix of r
PI	π	042	SR2490	mathematic
RN	Rn	052	DEREQ	instantaneo
SINT	sin θ	064	CHNTBL	sine of con
TANT	TAN θ	070	CHNTBL	tangent of
XLA, 8	$(X/La)_i$	815-822	AERODY	body statio
Z	Z	091	DEREQ	altitude
ZTR	Z_{TR}	092	PRELIM	transition

3. Output

WDØT	$\dot{W}$	086		change in v result of in in time

rce of put	Description	Units
NTBL	cosine of cone half angle	
ELIM	instantaneous axial length	ft.
ELIM	instantaneous bluntness ratio	-
L	matrix of mass loss rates at prescribed body points	lbm/ft^2 - sec.
490	mathematical constant, pi	-
REQ	instantaneous nose radius	ft.
NTBL	sine of cone half angle	-
NTBL	tangent of cone half angle	-
RODY	body station in percentage of axial length	-
REQ	altitude	ft.
ELIM	transition altitude	ft.
	change in weight per unit time due to ablation only - result of integrating $\dot{m}$'s over the body at any instant in time	lb/sec

B

4. Numerical Procedure

The equations of TØMALØ may be grouped into three main sections (1) sharp cone calculations, (2) laminar blunt cone calculations, (3) turbulent blunt cone calculations.

The first statement of the subroutine tests the bluntness ratio in order to direct the calculations to the appropriate set of equations. If LAMDA $\leq 10^{-3}$, the integration of mass loss along a sharp cone is determined by from the following equations:

$$Z \geq ZTR \qquad M_T = 2.9618 \sqrt{2.}\ \dot{m}(2,8)\ L_a^{\,2}\ \frac{\tan\theta}{\cos\theta}$$

$$Z < ZTR \qquad M_T = 3.04\,(2)^{0.2}\ \dot{m}(3,8)\ L_a^{\,2}\ \frac{\tan\theta}{\cos\theta}$$

The flow of the subroutine then circumvents the blunt cone calculations and goes to statement 3. If LAMDA $> 10^{-3}$, subroutine flow skips to statement 2 and performs the blunt cone calculations. The geometric factors TEM, TEM1, TEM2, and TEM3 are defined as

TEM = π La * La / (cos θ * cos θ)

TEM1 = $R_n(1-\sin\theta)/La$

TEM2 = 0.5 sin θ

TEM3 = (.766 - sin θ) / .234

If Z is less than ZTR, control is directed to statement 4, the beginning of the turbulent calculations. If Z is greater than or equal to ZTR the subroutine proceeds to the laminar calculations.

A trapezoidal rule integration is used to obtain the blunt cone laminar mass loss rate. First, the contribution from the vehicle nose cap is calculated using the equation

$$M_n = \pi Rn(1. - \sin\theta)\ (\dot{m}(1,1) + \dot{m}(2,1))\ Rn.$$

Next, the contribution of the conical frustum segments are determined from the equation

$$M_{c_i} = TEM\left[\left(\frac{X}{La}\right)_{i+1} - \left(\frac{X}{La}\right)_i\right]\left[TEM1 + TEM2\left(\left(\frac{X}{L_a}\right)_{i+1} + \left(\frac{X}{La}\right)_i\right)\right]$$

$$M_{c_i} = M_{c_i} * (\dot{m}(2,i) + \dot{m}(2,i+1))$$

for $i = 1, 2, \ldots 6$

Then, subroutine control skips around the turbulent blunt cone calculations to statement 7.

The turbulent flow equations for integrating blunt cone mass loss rate are similar to those for laminar flow. The nose cap contribution is

$$M_n = .7351 * R_n * R_n\ (\dot{m}(1,1) + \dot{m}(4,1) + (\dot{m}(4,1) + \dot{m}(3,1))\ TEM3)$$

For the conical frustum segments,

$$M_{c_i} = TEM * \left(\left(\frac{X}{La} \right)_{i+1} - \left(\frac{X}{La} \right)_i \right) \left(TEM1 + TEM2 \left(\left(\frac{X}{La} \right)_{i+1} + \left(\frac{X}{La} \right)_i \right) \right)$$

$$M_{c_i} = M_{c_i} \left(\dot{m}(3, i) + \dot{m}(3, i+1) \right)$$

for $i = 1, 2, \ldots 6$

The summation of the contributions from the nose cap and conical frustum segments is done in the equations following statement 7.

$$M_T = M_n$$

$$M_T = M_T + \sum_{i=1}^{6} M_{c_i}$$

In statement 3 $\dot{W}$ is set equal to $(-M_T)$ then the return to the calling subroutine is accomplished.

5. Other Information

A. SUBRØUTINE TØMALØ calls in the functions
   1. DSQRT
   2. FDXPD

B. SUBRØUTINE TØMALØ is called by
   1. SUBRØUTINE VIXEN
   2. SUBRØUTINE DEREQ

SUBRØUTINE NØSEBL

1. Purpose

SUBRØUTINE NØSEBL calculates the derivatives of the nose radius and base radius with time, using body geometric parameters and surface recession rates at the stagnation point and at the maximum diameter point of the cone. The shape change calculation, which assumes a constant cone half angle and retention of sphere cone shape is performed only in continuum flow. In the rarefied flow regime, the derivatives are set equal to zero.

2. Input

*indicates integer quantity and NØCCUR number code.

Name	Symbol	Occur/Noccur Number	Source of Input	Desc
CØST	cos θ	008	CHNTBL	cosine of
LAMDA	λ	032	PRELIM	instantane
SDØT, 32	$\dot{s}_i$	2740-2771	MASSLØ	surface re
SINT	sin θ	064	CHNTBL	sine of co
XBAR	$\bar{\chi}$	090	PRELIM	viscous in
XUP	χ_{up}	237	READIT or SR2490	value of X
Z	Z	091	DEREQ	altitude
ZTR	$Z_{TR.}$	092	PRELIM	transition

3. Output

RBDØT	$\dot{R}b$	060		time rate
RNDØT	$\dot{R}n$	059		time rate

e of ut	Description	Units
TBL	cosine of cone half angle	
LIM	instantaneous bluntness ratio	
SLD	surface recession rate along body	ft/sec
TBL	sine of cone half angle	
LIM	viscous interaction parameter	
DIT or 90	value of XBAR at start of continuum flow	
EQ	altitude	ft.
LIM	transition altitude	ft.
	time rate of change of nose radius	ft/sec
	time rate of change of base radius	ft/sec

B

4. Numerical Procedures

In statement 1, the quantity XBAR is tested against XUP. If XBAR > XUP, which indicates non-continuum flow, the control is directed to statement 2 and the subsequent zeroing of the derivatives RNDØT and RBDØT and return to the calling subroutine. If XBAR $\leq$ XUP, continuum flow, the numerical calculations for the derivatives proceed.

First the quantity RNDØT is calculated from the stagnation point surface recession rate and the sine of the cone half angle.

$$\dot{R}n = \dot{S}(1, 1) \frac{\sin \theta}{1 - \sin \theta}$$

The bluntness ratio, LAMDA, is then tested and for values greater than 10^{-3}, control is directed to statement 3. Here, the altitude Z is tested against transition altitude. If Z $\geq$ ZTR, the quantity TEM is defined as the laminar blunt cone side wall recession rate at cone maximum diameter. If Z < ZTR, the quantity TEM is defined as the turbulent maximum diameter side-wall recession rate.

$$Z < ZTR \quad TEM = \dot{S} \quad (3, 7)$$

$$Z \geq ZTR \quad TEM = \dot{S} \quad (2, 7)$$

Then in statement 4, RBDØT is defined as $\dot{R}b = -TEM/\sin \theta$ and is followed by the return to calling subroutine.

If the bluntness ratio is less than or equal to 10^{-3}, the sharp cone maximum diameter sidewall recession appropriate to the flow regime is used in defining the parameter TEM.

$Z < ZTR \quad TEM = 0.8706 * SD\emptyset T(3,8)$

$Z \geq ZTR \quad TEM = 0.7071 * SD\emptyset T(2.8)$

The calculations are directed to statement 4, where $\dot{R}b$ is defined, and then returned to the calling subroutine.

5. Other Information

A. SUBRØUTINE NØSEBL calls in no other subroutines or functions.

B. SUBRØUTINE NØSEBL is called by SUBRØUTINE DEREQ.

3.1.6 Rotational Calculations

The calculations for angle of attack are contained in SUBROUTINE ROTATE. Two methods for determining angle of attack are available. The first is an uncoupled three degree of freedom in rotation calculation. The second is a simplified model for calculating the angle of attack which employs Bessel functions of the first kind and Neumann functions (Bessel functions of the second kind).

SUBRØUTINE RØTATE (DERIV, LP)

1. Purpose

In order to incorporate the effects of angle of attack on the drag coefficient, SUBROUTINE ROTATE may be employed to determine angle of attack used in SUBROUTINE DRAGCØ calculations. The solution may be accomplished by either of two methods, where the value of the input quantity LØPT is the control parameter. When LØPT equals zero, an uncoupled three degree of freem in rotation calculation is performed in which the following derivatives are calculated:

DERIV(8) = $\dot{\psi}$

DERIV(9) = $\dot{\theta}_a$

DERIV(10) = $\dot{\phi}$

DERIV(11) = $\dot{Q}$

DERIV(12) = $\dot{R}$

DERIV(13) = $\dot{P}$

A simplified angle of attack solution requiring no integration is substituted when LØPT is equal to 2.

2. Input

* indicates integer quantity and an NOCCUR number.

Name	Symbol	Occur/Noccur Number	Source of Input	De
ALBARP	$\bar{\alpha}'$	131	VIXEN	last minimu
ALST	α_{STOP}	122	SR2490 or READIT	angle of att
ALWIG2	$\tilde{\alpha}'$	130	VIXEN	maximum p
AREF	A_{REF}	001	PRELIM	reference a
CD	C_D	016	DRAGCO	total drag c
CMALP	C_{m_α}	013	PRELIM	partial deri
CMQ	C_{m_q}	020	PRELIM	damping in
CNALP	C_{n_α}	012	PRELIM	partial deri
D	D	021	PRELIM	base diamet
G	g	027	SR2490 or READIT	conversion
LØPT	-	07 *	SR2490 or READIT	trajectory o
MAXTAB	-	04 *	CHNTBL	number of v
MX	M_x	210	PRELIM	thrusting m
MY	M_y	211	PRELIM	thrusting m
MZ	M_z	212	PRELIM	thrusting m
P	P	043	DEREQ	component
PHI	ϕ	044	DEREQ	Euler angle
PI	π	042	SR2490	mathematic

Source of Input	Description	Units
XEN	last minimum in α'	radians
R2490 or EADIT	angle of attack stop control	radians
XEN	maximum preceding last maximum in α'	radians
RELIM	reference area	ft^2
RAGCO	total drag coefficient	-
RELIM	partial derivative of moment coefficient with α	1/radian
RELIM	damping in pitch	-
RELIM	partial derivative of normal force coefficient with α	1/radians
RELIM	base diameter	ft
R2490 or EADIT	conversion from slug to lbm	lbm/slug
R2490 or EADIT	trajectory option code	-
HNTBL	number of values in X_{CG}/D, I, I_x table	-
RELIM	thrusting moment about X axis	ft-lb
RELIM	thrusting moment about Y axis	ft-lb
RELIM	thrusting moment about Z axis	ft-lb
ZREQ	component of angular velocity	rad/sec
CREQ	Euler angle Φ	radians
R2490	mathematical constant	-

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	
PSI	Ψ	045	DEREQ	Euler angl
Q	Q	050	DEREQ	component
QD	q_D	051	PRELIM	dynamic pr
RHØIN1	$\rho_{\infty 1}$	056	PRELIM	free strea
RHØINF	ρ_∞	055	PRELIM	free strea
SING0	$\sin(\gamma_{f_0})$	221	F123	sine of flig
SMR	R	065	DEREQ	component
T	t	075	VIXEN	time for co
TABI	$I_{(TABLE)}$	894-943	CHNTBL	moment of
TABIX	I_x (TABLE)	944-993	CHNTBL	transverse
TABZ	$Z_{(TABLE)}$	994-1043	CHNTBL	altitude tab
TCRIT	T_{CRIT}	077	READIT	limit on cy
TECØN	T_{ECON}	078	SR2490 or READIT	limit on c
THEALO	θ_{a_o}	113	READIT	value of θ
THEALP	θ_a	071	DEREQ	instantaneo
V	V	082	DEREQ	velocity
W	W	084	DEREQ	initial weig
XBAR	$\bar{X}$	090	PRELIM	viscous int
XUP	X_{UP}	237	PRELIM	viscous inte on continuu
Z	Z	091	DEREQ	altitude
Z0	Z_0	108	READIT	initial alti

rce of put	Description	Units
EREQ	Euler angle Ψ	radians
EREQ	component of angular velocity	rad/sec
RELIM	dynamic pressure	lb/ft^2
RELIM	free stream density in lbm/ft^3	lbm/ft^3
RELIM	free stream density in $slug/ft^3$	$slug/ft^3$
123	sine of flight path angle at reentry	-
EREQ	component of angular velocity	rad/sec
XEN	time for complete cycle of angle of attack	seconds
HNTBL	moment of inertia table	$slug-ft^2$
HNTBL	transverse moment of inertia table	$slug-ft^2$
HNTBL	altitude table for X_{CG}/D, I, I_x tables	ft
EADIT	limit on cycle time	seconds
12490 or EADIT	limit on cycle time	seconds
EADIT	value of θ_a at reentry	radians
EREQ	instantaneous value of θ_a	radians
EREQ	velocity	ft/sec
EREQ	initial weight - ablated weight	lb.
RELIM	viscous interaction parameter	-
RELIM	viscous interaction parameter value for upper bound on continuum flow	-
EREQ	altitude	ft.
RADIT	initial altitude	ft.

B

3. Output

Name	Symbol	Occur Number	Desc
ALALI	$\alpha/\alpha_{INITIAL}$	229	ratio of α t
ALPHA	α	002	angle of att
ALPRIM	α'	003	angle of att
CM	C_m	202	moment co
CN	C_n	203	normal for
DERIV, 16	-	-	derivatives
LP	-	- *	error code
SMF	f	081	frequency c

Description	Units
ratio of α to α at initial condition	-
angle of attack for use in $C_{D_{P_\alpha}}$ / $C_{D_{P_{\alpha=0}}}$ relation	radians
angle of attack used in maximum-minimum testing	radians
moment coefficient	-
normal force coefficient	-
derivatives to be integrated in ADM4RK	-
error code	-
frequency of angle of attack cycle	cycles/sec.

B

4. Numerical Procedure

The control parameter, LØPT is tested in the first statement of the subroutine. If LØPT is non-zero, the derivatives of the Euler angles and of the angular velocities, DERIV(J) for J = 8 through 13, are zeroed by statement 9, which is called for each value of J by a DØ statement. Then, for LØPT equal to 2, control passes to statement 15; for other non-zero values of LØPT control returns to the calling subroutine. When LØPT is zero, the zeroing of the derivatives is bypassed and control passes to statement 8.

Statement 8 is the beginning of the three degree of freedom in rotation calculations. First the sines and cosines of the Euler angles ϕ, Ψ, and θ_a are determined. The derivatives of the Euler angles are then found from the following formulations:

$$\text{DERIV(8)} = (R \cos\phi + Q \sin\phi) \cos\theta_a$$

$$\text{DERIV(9)} = Q \cos\phi - R \sin\phi$$

$$\text{DERIV(10)} = P + \text{DERIV(8)} \sin\theta_a$$

The product $\cos\theta_a \cos\Psi$ is tested; if less than zero, an error code LP is set equal to 6, an error message is printed out by statement 100, and control returns to the calling subroutine. If $\cos\theta_a \cos\Psi$ is greater than or equal to zero, the angle of attack α' is

$$\alpha' = \cos^{-1}\left[\cos\theta_a \cos\Psi\right]$$

and using this result

$$C_n = C_{n_\alpha} \alpha'.$$

Bypassing the error message and going to statement 4, the moments of inertia I and I_x are determined from the tabular inputs, using FUNCTION TABLE, and tested. If I equals zero, control passes to statement 13 where an error message is printed, error code set to 6, and return to calling subroutine enacted. If I_x equals zero, control goes to statement 14 where a similar procedure occurs. If both I and I_x are non-zero, the following terms are defined in preparation for calculation of the remaining derivatives:

$$C_m = C_{m_\alpha} \alpha'$$

$$TEM1 = q_D A_{ref} D/I$$

if $\alpha' = 0$ $\quad TEM2 = 0.0$

if $\alpha' \neq 0$ $\quad TEM2 = C_m / \sin \alpha'$

$$TEM3 = \sin \theta_\alpha \cos \Psi$$

$$TEM4 = 0.5 \; D \; C_{m_q} / V$$

$$TEM5 = (I - I_x) P / I$$

These are used to find

$$DERIV(11) = TEM1(TEM2(TEM3 \cos \phi + \sin \Psi \sin \phi) + (TEM4)Q) + (TEM5) SMR + \frac{M_y}{I}$$

$$DERIV(12) = TEM1(TEM2(\sin \Psi \cos \phi - TEM3 \sin \phi) + (TEM4)SMR) - (TEM5) Q + \frac{M_z}{I}$$

$$DERIV(13) = \frac{M_x}{I_x}$$

The viscous interaction parameter is tested. If the XBAR is greater than or equal to XUP, indicating rarefied flow regime, the quantity α, ALPHA, is set equal to α', ALPRIM, in statement 11 and control returns to the calling subroutine. If XBAR is less than XUP, continuum flow,

control passes to statement 10, where the cycle time T is tested. If T equals zero, which indicates that a complete cycle has not been completed and a maximum and minimum defined, control passes to statement 11, α is set equal to α', and return to calling subroutine is executed. If T is non-zero and greater than or equal to the input TCRIT, control passes to statement 11. If the non-zero T is less than TCRIT, but greater than or equal to TECØN, an integrated angle of attack effect is used in DRAGCØ, control passes to statement 12, which returns it to the calling subroutine. If the non-zero T is less than TCRIT and less than TECØN, the angular velocity P is tested to determine the appropriate definition for effective angle of attack as follows:

$$P = 0 \qquad \alpha = 2.0 \quad \frac{\tilde{\alpha}'}{\pi}$$

$$P \neq 0 \qquad \alpha = 0.5 \ (\tilde{\alpha}' + \bar{\alpha}')$$

where $\bar{\alpha}'$ is the last minimum and $\tilde{\alpha}'$ is the maximum preceding the last maximum, $\tilde{\alpha}'$. The control passes to statement 12 which returns it to the calling subroutine.

In place of this rather complicated and time consuming calculation, a simplified angle of attack model is available in the equations beginning with statement 15, when LØPT is set to 2. Statement 15 defines code LL used in function TABLE. If altitude, Z , is less than or equal to 100.0 feet control passes to statement 90, where α' is zeroed, the trajectory option code LØPT is set to 1 for a particle trajectory, and return to the calling subroutine is executed. If Z is greater than 100.0 feet, calculations continue with the determination of I from the input table using FUNCTION TABLE. If I is zero, the error code, error message

and return to calling subroutine following statement 13 are summoned. When I is non-zero, calculations continue by defining the quantity BETAZ

$$\text{BETAZ} = \cdot\cdot \log_e (\rho_{\infty_1} / .076474 \text{ lbm/ft}^3)$$

unless BETAZ $\leq$ .0001 then BETAZ is set equal to .0001. Then, if Z equals Z0 since no value for C_D has been determined yet, C_D is set to a nominal value of 0.8. The constants AK1 and AK2, and the cycle time TBAR are defined as follows:

$$\text{AK1} = A_{ref}\, g\, Z\left(2.0\ C_D - C_{n_a} + \frac{C_{m_q} D^2 W}{I g}\right) \Big/ \left(4.0\ \text{BETAZ}\ W\ |\sin \gamma_o|\right)$$

$$\text{AK2} = A_{ref}\, g\ Z^2 \left((C_{n_a} - C_D)\ \text{BETAZ}\ \frac{|\sin \gamma_o|}{Z} - \frac{C_{m_a} DW}{I g}\right) \Big/$$

$$\left(2.0 W (\text{BETAZ} * \sin \gamma_o)^2\right)$$

$$\text{TBAR} = -\ 2.0\ \pi Z / (\text{BETA} * V\ \sin \gamma_o \sqrt{\text{AK2} * \rho_\infty})$$

If the cycle time $\bar{t}$, TBAR, is less than or equal to input quantity TECØN, preset to 2.0 seconds, or is less than 1.0×10^{-10}, control passes to statement 16 and a non-oscillatory envelope calculation for angle of attack. If TBAR is greater than both TECØN and 1.0×10^{-10}, an oscillatory solution employing Bessel functions is performed as described below. The frequency F, SMF, is determined from $\bar{t}$ as $1/\bar{t}$. The argument of the Bessel functions, TEM, is defined by

$$\text{TEM} = 2.0 \sqrt{(\text{AK1} + \text{AK2})\rho_\infty}$$

and the subroutines BESSEL(TEM, 0.0 D0, 1.0 D-5, XJR, XJI, 1) and NEUMAN (1.0 D-5, TEM, 0.0D0, 10, 1.0D1, XNR, XNI, XJR(1), XJI(1), XJR(2), XJI(2), 1) are called in to calculate the Bessel functions of the first and second kind of the zeroth and first order. Following these calls is a group of equations for quantities defined at the reentry altitude Z0 which are bypassed when $Z \neq Z0$. When $Z = Z0$,

$$d(TEM)/dZ = -0.5\ TEM(BETAZ/Z)$$

$$DENAC2 = \text{denominator of } AC2 = \left(d(TEM)/dZ\right)\left(-Y_1(TEM) \times J_0(TEM) + J_1(TEM) * Y_0(TEM)\right)$$

$$NUMER1 = \text{first part of numerator of } AC2 = \left(J_0(TEM)\ AK1\ \theta_{a_0}\ 2.3769 \times 10^{-3}\ BETAZ\right) / \left(Z \exp(AK1\,\rho_\infty + BETAZ)\right)$$

$$NUMER2 = \text{second part of numerator of } AC2 = \theta_{a_0}\left(d(TEM)/dZ\right)\left(-J_1(TEM) \exp(-AK1\,\rho_\infty)\right)$$

$$AC2 = (NUMER1 - NUMER2) / DEN\,AC2$$

$$AC1 = \left[\theta_{a_0} \exp(-AK1\,\rho_\infty) - AC2\ Y_0(TEM)\right] / J_0(TEM)$$

$$a/a_{initial} = \exp(AK1\,\rho_\infty) \Big/ \sqrt{\pi \sqrt{AK2\,\rho_\infty}}$$

Then, for all $Z > 100.0$ feet

$$a' = \exp(AK1\,\rho_\infty)\ (AC1\ J_0(TEM) + AC2\ Y_0(TEM))$$

$$C_n = C_{n_a}\ a'$$

$$C_m = C_{m_a}\ a'$$

$$a = a'$$

after which control is returned to the calling subroutine.

Statement 16 is the beginning of the non-oscillatory envelope calculation

$$a/a_o = \exp(AK1\ \rho_\infty) \Big/ \sqrt{\pi\ \sqrt{AK2\ \rho_\infty}}$$

for $Z = Z0$ $\quad a/a_{initial} = a/a_o$

$$a' = 0.63661\ \theta_{a_o}\ \frac{(a/a_o)}{(a/a_{initial})}$$

$$f = 0.0$$

However, if $a' \leq a_{STOP}$, a' is set equal to zero and LØPT, the trajectory option code, is set to 1 so that only a particle trajectory will be calculated henceforth. If $a' > a_{STOP}$, control passes to statement 92, where these quantities are defined

$$C_n = C_{n_a}\ a'$$

$$C_m = C_{m_a}\ a'$$

$$a = a'$$

before the return to the calling subroutine is executed.

5. Other Information

A. SUBRØUTINE RØTATE is called in by SUBRØUTINE DEREQ.

B. SUBROUTINE RØTATE calls in the following program subroutines and functions:

1. SUBRØUTINE BESSEL
2. SUBRØUTINE NEUMAN
3. FUNCTION TABLE

C. In addition, the following library functions are utilized:

1. ACØSR
2. DEXP
3. DSIN
4. DCØS
5. DSQRT
6. DLØG

Bessel Function Calculations

The calculations for the Bessel function of the first kind are contained in subroutines BESSEL, JNXBES, and DRLIM. The Bessel functions of the second kind (Neumann functions) are calculated by subroutines NEUMAN, NEUMP0, NEUMQ0, DIVMLT, NEUMN0, and NEUMN1.

SUBROUTINE BESSEL (XR, XI, C, N, ZNR0, ZNI0, IT)

1. Purpose

SUBROUTINE BESSEL calculates for complex arguments the Bessel functions of integral order zero through N, where the maximum value for N is twenty-four.

2. Input

Fortran Symbol	Source of Input	Description
C	ROTATE	convergence criterion
IT	ROTATE	integer code = 1
N	ROTATE	integer indicating maximum order to be calculated
XI	ROTATE	imaginary argument of Bessel function
XR	ROTATE	real argument of Bessel function

3. Output

Fortran Symbol	Description
ZNI0	imaginary part of solution vector
ZNR0	real part of solution vector

4. Numerical Procedure

The descriptive use of the arguments for effective and proper use of the subroutine are as follows:

XR	real argument of the Bessel function
XI	imaginary argument of the Bessel function
C	convergence criterion for power series (10^{-5} gives good results)
N	maximum order of Bessel function (24 is maximum allowed)
ZNR0	real part of the solution vector (dimension 25, ZNR0 (1, IT) is zero order, ZNR0(2, IT) is first order, etc.)
ZNI0	imaginary part of the solution vector (dimension 25, ZNI0 (1, IT) is zero order, ZNI0(2, IT) is first order, etc.)
IT	integer code

The sum of the squares of XR and XI is calculated and designated as B and the integer N1 is set equal to N + 1. Then B is tested. If B = 0, subroutine continues to statement 4, where the real and imaginary parts of the solution vectors for orders N and N + 1 are set equal to zero. Control then passes to statement 5, where the recursive formula is used to zero the real and imaginary parts of the solution vectors for all remaining orders down to zeroth order. If B ≠ 0, the group of equations beginning with statement 4 is bypassed and control passes to statement 2. Here SUBROUTINE JNXBES is called in to generate, using the standard power series, the real and imaginary parts of the solution vectors (Bessel functions) of Nth and N-1 order, i.e., ZNR0(N + 1, IT), ZNI0(N + 1, IT) and ZNR0(N, IT), ZNI0(N, IT) respectively. The quantities XRB and XIB are then defined as

$$XRB = XR/(XR^2 + XI^2)$$

$$XIB = - XI/(XR^2 + XI^2)$$

before control passes to statement 5. The following recursive formulas are then used to calculate the real and imaginary parts of the solution

vectors for all remaining orders down to zeroth order.

$$ZNR0(N-I+1, IT) = 2.0\,(N-I+1)\,\left[ZNR0(N-I+2, IT) * XRB - ZNI0(N-I+2, IT) * XIB\right] - ZNR0(N-I+3, IT)$$

$$ZNI0(N-I+1, IT) = 2.0(N-I+1)\left[ZNR0(N-I+2, IT) * XIB - ZNI0(N-I+2, IT) * XRB\right] - ZNI0(N-I+3, IT)$$

for I = 2 to N

For real arguments between 1 and 10 there is better than 6 place accuracy. For complex arguments of increasing absolute value, there will be diminishing accuracy. However, the lower orders will have more significant figures. In order to obtain accurate results for the lower orders, it is necessary to use large N. Real arguments will give answers with more significant figures than complex arguments.

5. Other Information

A. SUBROUTINE BESSEL calls in SUBROUTINE JNXBES.

B. SUBROUTINE BESSEL is called by SUBROUTINE ROTATE.

SUBROUTINE JNXBES (ZR, ZI, NN, C, ZNR, ZNI)

1. Purpose

SUBROUTINE JNXBES is called by SUBROUTINE BESSEL to calculate the Bessel functions of the maximum order specified, N, and of the N-1 order.

2. Input

* indicates integer quantity

Name		Description
C		convergence criterion
NN	*	the order of the Bessel function to be determined
ZI		imaginary component of the argument of the Bessel function
ZR		real component of the argument of the Bessel function

3. Output

ZNI	imaginary component of the Bessel function
ZNR	real component of the Bessel function

4. Numerical Procedure

All variable names having Q as first letter are designated as complex numbers in SUBROUTINE JNXBES. The calculations of this subroutine begin with the definition of the complex number Q_x from the input real and imaginary components, ZR and ZI respectively, by the use of function DCMPLEX. The variable Z is then defined as the absolute

value of Q_x. Both the real and imaginary parts of QXXI (Q_{x_I}) and QXNU (Q_{x_ν}) are set equal to 0.0. The absolute value of Q_x, Z, is tested; if Z is less than 19.9999999999, control passes to statement 10; if Z is greater than or equal to the specified value, then control passes to statement 11.

Statement 11 is the beginning of the calculations of the Bessel function for large values of the argument. Preceding the statement DØ12 I = 2, 4000, 2 are initial definitions of quantities used both in the DØ loop ending with statement 12, and in the DØ loop beginning with statement 13 and ending with statement 15. The DØ loop ending with statement 12 evaluates the series

$$QP = P_{NN}(Z) = \sum_{k=0}^{M} (-1)^k \frac{(NN, 2k)}{(2Z)^{2k}}$$

$$= 1 - \frac{(4NN^2-1)(4NN^2-9)}{2!(8Z)^2} + \frac{(4NN^2-1)(4NN^2-9)(4NN^2-25)(4NN^2-49)}{4!(8Z)^4}$$

$$- \cdots\cdots$$

and the DØ loop beginning with statement 13 and ending with statement 15 evaluates the series

$$QQ = Q_{NN}(Z) = \sum_{k=0}^{M} (-1)^k \frac{(NN, 2k+1)}{(2Z)^{2k+1}} = \frac{4NN^2 - 1}{8Z}$$

$$- \frac{(4NN^2-1)(4NN^2-9)(4NN^2-25)}{3!(8Z)^3} + \cdots\cdots$$

until the convergence criterion is satisfied or M is 2000.

These quantities are used in the following equation

$$QANS = J_{NN}(Z) = \sqrt{\frac{2.0}{\pi Z}} \left\{ P_{NN}(Z) \cos\left(Z - \left(\frac{NN}{2} + \frac{1}{4}\right)\pi\right) + Q_{NN}(Z) \sin\left(Z - \left(\frac{NN}{2} + \frac{1}{4}\right)\pi\right) \right\}$$

The Bessel function $J_{NN}(Z)$ is separated into its real and imaginary components by employing SUBROUTINE DRLIM, into which QANS is fed and the 2 value array TEMP is produced. TEMP(1), the real, and TEMP(2), the imaginary component of $J_{NN}(Z)$, are utilized to define respectively ZNR and ZNI.

Statement 10 marks the beginning of the calculations for the Bessel function having an argument Z, whose absolute value is less than 19.9999999999. The following quantities to be used in the determination of the Bessel function of order NN are then defined

$$A = NN!$$

$$R = r = \sqrt{\left(\frac{Z_r}{2.0}\right)^2 + \left(\frac{Z_i}{Z_r}\right)^2}$$

$$THET = \theta = \arctan\left(\frac{Z_i}{Z_r}\right)$$

$$X2NR = X_{2N_r} = \cos(NN\theta)(Z_r^2 + Z_i^2)^{\frac{NN}{2}} / (2^{NN}\ NN!)$$

$$X2NI = X_{2N_i} = \sin(NN\theta)\ (Z_r^2 + Z_i^2)^{\frac{NN}{2}} / (2^{NN}\ NN!)$$

as well as the series

$$SUMR = 1.0 + \sum_{I=1}^{k \leq 32000} \frac{(-1)^I (Z_r^2 + Z_i^2)^I (\cos 2I\theta)(NN!)}{(4)^I \quad I! \quad (NN+I)!}$$

and

$$SUMI = \sum_{I=1}^{K \leq 32000} \frac{(-1)^I (Z_r^2 + Z_i^2)^I (\sin 2I\theta)}{(4)^I \quad I! \quad (NN+I)!}$$

where the number of terms in the series is dependent on the convergence criterion C. The absolute value of the difference between the last two terms of the series must be less than C. Using the previously defined terms and series to define the real and imaginary parts of the Bessel function, respectively

$$Z_{N_R} = X_{2N_R} \; SUMR - X_{2N_I} \; SUMI$$

$$Z_{N_I} = X_{2N_R} \; SUMI + X_{2N_I} \; SUMR$$

These components, if simplified, would yield the following expression for the Bessel function

$$J_{NN}(Z) = \left(\frac{1}{2} Z\right)^{NN} \sum_{I=0}^{k} \frac{\left(-\frac{1}{4} Z^2\right)^I}{I! \; (NN+I)!}$$

5. Other Information

A. SUBROUTINE JNXBES is called in by SUBROUTINE BESSEL.

B. SUBROUTINE JNXBES calls in

1. SUBROUTINE DRLIM
2. external function ATANQR
3. internal functions
   a. CDABS
   b. DSIN
   c. CDSIN
   d. DCØS
   e. CDCØS
   f. DSQRT
   g. CDSQRT
   h. FDXPI
   i. FCDXI
   j. CCDVD
   k. CDMPY

DRLIM(C, R)

1. Purpose

SUBROUTINE DRLIM sets the real and imaginary parts of a complex number C and sets them equal to the real numbers of the two value array R.

2. Input

Name	Source of Input	Description
C	JNXBES	complex number which is to be used to form two real numbers

3. Output

Name	Description
R(1)	real number which corresponds to the real part of the complex number, C
R(2)	real number which corresponds to the imaginary part of the complex number, C

4. Numerical Procedure

The two real two-valued arrays R and A are used in conjunction with the two complex numbers C2 and C in SUBROUTINE DRLIM. The quantities C2 and A are placed in equivalence which gives A(1) the value of the real part of C2 and A(2) the value of the imaginary part of C2. Then C2

is given the value of C, which because of the equivalence gives values to A(1) and A(2). The real numbers R(1) and R(2) are then set equal respectively to A(1) and A(2).

5. Other Information

A. SUBROUTINE DRLIM (C, R) is called by SUBROUTINE JNXBES using the arguments (QANS, TEMP).

B. SUBROUTINE DRLIM calls in no other functions or subroutines.

SUBROUTINE NEUMAN (C, XR, XI, N, XL, ZZNR, ZZNI, X0R, X0I, X1R, X1I, IT)

1. Purpose

SUBROUTINE NEUMAN calculates for complex arguments the Neumann functions of integral order zero through twenty-four. For arguments of magnitude less than XL, the standard power series is used to calculate the Neumann functions of order zero and one. For arguments whose magnitude is greater than XL, an asymptotic series solution is employed to obtain the Neumann functions of order zero and one. A formula is then used to generate each increasing order until the Nth order is determined.

2. Input

* indicates integer quantity

Name	Symbol		Description
C			convergence criterion for power series
IT		*	integer code = 1
N		*	maximum order of Neumann function
X0I	J_{o_i}		imaginary part of zeroth order Bessel function
X0R	J_{o_r}		real part of zeroth order Bessel function
X1I	J_{1_i}		imaginary part of first order Bessel function
X1R	J_{1_r}		real part of first order Bessel function
XI	X_i		imaginary argument of Neumann function
XL			solution value option, see explanation in part 1.
XR	X_r		real argument of Neumann function

3. Output

Name	Symbol	Description
ZZNI	Y_{N_i}	imaginary part of solution vector
ZZNR	Y_{N_r}	real part of solution vector

4. Numerical Procedure

SUBROUTINE NEUMAN defines the quantity D as the absolute magnitude of the argument.

$$D = \sqrt{X_r^2 + X_i^2}$$

If D equals zero, control passes to statement 10 where the real and imaginary parts of the zeroth and first order Neumann functions are zeroed, before passing to statement 11. If D is non-zero control passes to statement 8. D is then tested against XL. If $D \leq XL$, control passes to the power series calculations following statement 1. If $D > XL$, control passes to statement 2, which marks the beginning of the asymptotic series solution.

The asymptotic series for the zeroth order Neumann function is

$$Y_o(Z) = \sqrt{\frac{2}{\pi Z}} \left[P_o(Z) \sin\left(Z - \frac{\pi}{4}\right) + Q_o(Z) \cos\left(Z - \frac{\pi}{4}\right) \right]$$

where

$$P_o(Z) = 1 - \frac{(-1)(-9)}{2!\,(8Z)^2} + \frac{(-1)(-9)(-25)(-49)}{4!\,(8Z)^4} + \ldots.$$

$$Q_o(Z) = \frac{-1}{8Z} - \frac{(-1)(-9)(-25)}{3!\,(8Z)^3} + \ldots.$$

Employing the definitions $Z = X_r + iX_i$, $P_o(Z) = P_{N_r} + iP_{N_i}$, $Q_o(Z) = Q_{N_r} + i\,Q_{N_i}$, to expand the expression for $Y_o(Z)$

$$Y_o(Z) = \sqrt{\frac{2}{\pi}}\; Z^{-1/2} \left[(P_{N_r} + i\,P_{N_i}) \sin\left(X_r - \frac{\pi}{4} + i\,X_i\right) + (Q_{N_r} + i\,Q_{N_i}) \cos\left(X_r - \frac{\pi}{4} + i\,X_i\right) \right]$$

Using DeMoivre's theorem and expanding the sines and cosines

$$Y_o(Z) = \sqrt{\frac{2}{\pi}}\; Z^{-1/2} \left[(P_{N_r} + i\,P_{N_i}) \left\{ \sin\left(X_r - \frac{\pi}{4}\right) \cos(i\,X_i) + \sin(i\,X_i) \cos\left(X_r - \frac{\pi}{4}\right) \right\} + (Q_{N_r} + i\,Q_{N_i}) \left\{ \cos\left(X_r - \frac{\pi}{4}\right) \cos(i\,X_i) - \sin(i\,X_i) \sin\left(X_r - \frac{\pi}{4}\right) \right\} \right]$$

then

$$Y_o(Z) = \sqrt{\frac{2}{\pi}}\; \frac{\cos\left(\frac{\theta}{2}\right) - i\,\sin\left(\frac{\theta}{2}\right)}{(X_r^2 + X_i^2)^{1/2}} \left[(P_{N_r} + i\,P_{N_i}) \left\{ \sin\left(X_r - \frac{\pi}{4}\right) \left(\frac{e^{X_i} + e^{-X_i}}{2}\right) + \cos\left(X_r - \frac{\pi}{4}\right) \left(\frac{e^{X_i} - e^{-X_i}}{2i}\right) \right\} + (Q_{N_r} + i\,Q_{N_i}) \left\{ \cos\left(X_r - \frac{\pi}{4}\right) \left(\frac{e^{X_i} - e^{-X_i}}{2}\right) - \sin\left(X_r - \frac{\pi}{4}\right) \left(\frac{e^{X_i} - e^{-X_i}}{2i}\right) \right\} \right]$$

The expression for $Y_o(Z)$ is evaluated in the computations beginning with statement 2.

First SUBROUTINES NEUMP0 and NEUMQ0 are called to evaluate the real and imaginary parts of $P_o(Z)$ and $Q_o(Z)$, respectively. The the following quantities are defined

$$PI2RT = \sqrt{\frac{2}{\pi}}$$

$$XRP = X_r - \frac{\pi}{4} = X_r - 0.7853982$$

$$C\emptyset SR = \cos(R) = \cos\left(X_r - \frac{\pi}{4}\right)\left(\frac{e^{X_i} + e^{-X_i}}{2.0}\right)$$

$$C\emptyset SI = \cos(I) = -\sin\left(X_r - \frac{\pi}{4}\right)\left(\frac{e^{X_i} - e^{-X_i}}{2.0}\right)$$

$$SINR = \sin(R) = \sin\left(X_r - \frac{\pi}{4}\right)\left(\frac{e^{X_i} + e^{-X_i}}{2.0}\right)$$

$$SINI = \sin(I) = \cos\left(X_r - \frac{\pi}{4}\right)\left(\frac{e^{X_i} - e^{-X_i}}{2.0}\right)$$

$$\emptyset R = X_r / (X_r^2 + X_i^2)$$

$$\emptyset I = -X_i / (X_r^2 + X_i^2)$$

If the imaginary part of the argument, X_i, is zero, define

$$X\emptyset R = \sqrt{X_r/(X_r^2 + X_i^2)}$$

$$X\emptyset I = 0.0$$

and control passes to statement 7. If X_i is non-zero, define

$$\text{THTH0} = \theta = \tan^{-1}(\text{ØI}/\text{ØR})$$

$$\text{XØR} = -(X_r^2 + X_i^2)^{-1/2} \cos(\theta/2)$$

$$\text{XØI} = -(X_r^2 + X_i^2)^{-1/2} \sin(\theta/2)$$

These quantities are used to define

$$\text{ZNINR} = Q_{N_r} \cos(R) - Q_{N_i} \cos(I) + P_{N_r} \sin(R) - P_{N_i} \sin(I)$$

$$\text{ZNINI} = Q_{N_i} \cos(R) + Q_{N_r} \cos(I) + P_{N_i} \sin(R) + P_{N_r} \sin(I)$$

which are employed in the equations for the real and imaginary parts of $Y_o(Z)$

$$\text{ZZNR}(1, \text{IT}) = Y_{o_r} = \sqrt{\frac{2}{\pi}} (\text{XØR} * \text{ZNINR} - \text{XØI} * \text{ZNINI})$$

$$\text{ZZNI}(1, \text{IT}) = Y_{o_i} = \sqrt{\frac{2}{\pi}} (\text{XØI} * \text{ZNINR} + \text{XØR} * \text{ZNINI})$$

To determine the first order Neumann function from the Wronskian $W\left\{J_o(Z), Y_o(Z)\right\} = J_1(Z)\, Y_o(Z) - J_o(Z)\, Y_1(Z) = \frac{2}{\pi Z}$ which gives

$$Y_1(Z) = \frac{Y_o(Z)\, J_1(Z) - 2/(\pi Z)}{J_o(Z)}$$

we define the parameters

$$XUMR = Y_{o_r} J_{1_r} - Y_{o_i} Y_{1_i} - \frac{2}{\pi} (X_r / (X_r^2 + X_i^2))$$

$$XUMI = Y_{o_i} J_{1_r} + Y_{o_r} J_{1_i} - \frac{2}{\pi} (-X_i / (X_r^2 + X_i^2))$$

before SUBROUTINE DIVMLT is called to evaluate the equations for the real and imaginary parts of $Y_1(Z)$.

$$ZZNR(2, IT) = Y_{1_r} = (XUMR * J_{o_r} - XUMI * J_{o_i}) / (J_{o_r}^2 + J_{o_i}^2)$$

$$ZZNR(2, IT) = Y_{1_i} = (XUMR * (-J_{o_i}) + XUMI * J_{o_r}) / (J_{o_r}^2 + J_{o_i}^2)$$

Control then passes to statement 3, where the higher order Neumann functions are obtained from the recurrence relationship $Y_{\nu-1}(Z) + Y_{\nu+1}(Z) = \frac{2\nu}{Z} Y_\nu(Z)$ which may be cast in the following form for $I = \nu + 1$

$$Y_I(Z) = \frac{2(I-1)}{Z} Y_{I-1}(Z) - Y_{I-2}(Z)$$

This may be expressed in terms of its real and imaginary components for $1/Z = \frac{\cos\theta - i\cos\theta}{r}$

$$Y_I(Z) = Y_{I_r} + i Y_{I_i} = 2(I-1) \left[Y_{I-1_r} \frac{\cos\theta}{r} - Y_{I-1_i} \left(\frac{-\sin\theta}{r}\right) \right] - Y_{I-2_r}$$

$$+ i \left\{ 2(I-1) \left[Y_{I-1_i} \left(\frac{-\sin\theta}{r}\right) + Y_{I-1_r} \frac{\cos\theta}{r} \right] - Y_{I-2_i} \right\}.$$

The DØ loop which includes statement 4 computes the quantities

$$ZZNR(I, IT) = Y_{I-1}$$

$$ZZNI(I, IT) = Y_{I-1}$$

for all orders from 2 to the maximum N, i.e., for all I from 3 to N + 1, before returning to the calling subroutine.

Statement 1 obtains control when the absolute magnitude of the argument of the Neumann function is less than or equal to the input XL. Here SUBROUTINE NEUMN0 is called to determine the power series representation for the zeroth order Neumann function, and SUBROUTINE NEUMN1 for the first order function. Control then passes to statement 3 where the higher orders are computed using the recurrence formula.

5. Other Information

A. SUBROUTINE NEUMAN is called by SUBROUTINE RØTATE.

B. SUBROUTINE NEUMAN calls in the following subroutines:

1. SUBROUTINE NEUMP0
2. SUBROUTINE NEUMQ0
3. SUBROUTINE DIVMLT
4. SUBROUTINE NEUMN0
5. SUBROUTINE NEUMN1

C. SUBROUTINE NEUMAN calls in the following functions:

1. ATANQR
2. DEXP
3. DSIN
4. DCØS
5. DSQRT

SUBROUTINE NEUMP0 (XR, XI, C, PNR, PNI)

1. Purpose

SUBROUTINE NEUMP0 evaluates the real and imaginary components of the quantity $P_o(Z)$ which is used in the asymptotic solution for the Bessel function of the second kind of zeroth order:

$$Y_o(Z) = \sqrt{\frac{2}{\pi Z}} \left[P_o(Z) \sin\left(Z - \frac{\pi}{4}\right) + Q_o(Z) \cos\left(Z - \frac{\pi}{4}\right) \right]$$

where

$$P_o(Z) = 1 - \frac{(-1)(-9)}{2!\,(8Z)^2} + \frac{(-1)(-9)(-25)(-49)}{4!\,(8Z)^4} - \cdots$$

$$Q_o(Z) = \frac{-1}{8Z} - \frac{(-1)(-9)(-25)}{3!\,(8Z)^3} + \frac{(-1)(-9)(-25)(-49)(-81)}{5!\,(8Z)^5} - \cdots$$

2. Input

Name	Description
C	convergence criterion for series
XI	imaginary component of argument of Bessel function
XR	real component of argument of Bessel function

3. Output

Name	Description
PNI	imaginary part of series for $P_o(Z)$
PNR	real part of series for $P_o(Z)$

4. Numerical Procedure

The series for $P_o(Z)$ may be represented in the following manner:

$$P_o(Z) = 1.0 + \sum_{k=1}^{\infty} \frac{(4k-3)^2 (4k-1)^2 (-1)}{(2k)(2k-1)(8^2 Z^2)} \quad S(k-1)$$

where S(k-1) represents the preceding term in the series, i.e.,

$$S(0) = 1.0, \quad S(1) = \frac{(4(1)-3)^2 (4(1)-1)^2 (-1)}{[2(1)][2(1)-1](8Z^2)}$$

$$S(k-1) = \frac{(4(k-1)-3)^2 (4(k-1)-1)^2 (-1)}{(2(k-1))(2(k-1)-1)(8z^2)} \quad S(k-2)\, S(k-3) \ldots\ldots$$

Quantity $P_o(Z)$ and its representative series are divided into real and imaginary components, P_{N_r} and P_{n_i} respectively, to avoid the use of complex arithmetic in the subroutine.

$$P_o(Z) = P_{N_r} + i\, P_{N_i}$$

$$P_{N_r} = 1.0 + \sum_{k=1}^{\infty} X_{mult} (S_r(k-1)R_r - S_i(k-1)R_i) = \sum_{k=0}^{\infty} S_r(k)$$

$$P_{N_i} = \sum_{k=1}^{\infty} X_{mult} (S_r(k-1)\, R_i + S_i(k-1)R_r) = \sum_{k=1}^{\infty} S_i(k)$$

where

$$S(k-1) = S_r(k-1) + i\, S_i(k-1)$$

$$\frac{1}{Z^2} = R_r + i\, R_i, \quad \text{and} \quad X_{mult} = \frac{(4k-3)^2 (4k-1)^2}{(2k)(2k-1)\,64}$$

The calculations of SUBROUTINE NEUMP0 begin with the defining of quantities RR (R_r) and RI (R_i), and the setting of initial value for SR (S_r), SI (S_i), PNR (P_{N_r}) and PNI (P_{N_i}). The statement DØ 1 I = 1, 32000 where I corresponds to the k of the previous equation is the beginning of the evaluation of P_{N_r} and P_{N_i} as well as the series for S_r and S_i. The integer I is tested; if an even number, then integers MM = 1 and M = 2; if an odd number, then integers MM = 2 and M = 1. Then M replaces the K - 1 and MM the k counter on S_r and S_i. P_{N_r} is set equal to the sum of the last P_{N_r} and $S_r(M)$; P_{N_i} is set equal to the sum of the last P_{N_i} and $S_i(M)$. The absolute value of the difference between the last two terms of the series representing S_r is tested against the convergence criterion C. If this value is greater than C control passes to statement 6 and another term is added to the series for S_i and for S_r, and control returns to the DØ statement where I is increased by 1 and the entire procedure repeated. If this value is less than or equal to C, control passes to statement 5 where a similar test is performed on the terms of S_i. If the absolute value of the difference between the last terms in S_i is less than or equal to C, control passes to statement 7 and back to the calling subroutine. If this value is greater than C, control passes to statement 6 where the process of adding terms to the series continues.

5. Other Information

A. SUBROUTINE NEUMP0 functions properly for pure real argument Z; but, for complex or pure imaginary Z, an error in the definition of R_r causes erroneous results. The program under consideration uses this routine for real arguments only.

B. SUBROUTINE NEUMP0 is called by SUBROUTINE NEUMAN.

C. SUBROUTINE NEUMP0 calls in no other subroutines or internal functions.

SUBROUTINE NEUMQ0 (XR, XI, C, QNR, QNI)

1. Purpose

SUBROUTINE NEUMQ0 evaluates the real and imaginary components of the quantity $Q_o(Z)$ which is used in the asymptotic solution for the Bessel function of the second kind of zeroth order:

$$Y_o(Z) = \sqrt{\frac{2}{\pi Z}} \left[P_o(Z) \sin\left(Z - \frac{\pi}{4}\right) + Q_o(Z) \cos\left(Z - \frac{\pi}{4}\right) \right]$$

where

$$P_o(Z) = 1 - \frac{(-1)(-9)}{2!\,(8Z)^2} + \frac{(-1)(-9)(-25)(-49)}{4!\,(8Z)^4} \cdots$$

$$Q_o(Z) = \frac{-1}{8Z} - \frac{(-1)(-9)(-25)}{3!\,(8Z)^3} + \frac{(-1)(-9)(-25)(-49)(-81)}{5!\,(8Z)^5} \cdots$$

2. Input

Name	Description
C	convergence criterion for series
XI	imaginary component of argument of Bessel Function
XR	real component of argument of Bessel Function

3. Output

Name	Description
QNI	imaginary part of series for $Q_o(Z)$
QNR	real part of series for $Q_o(Z)$

4. Numerical Procedure

The series for $Q_o(Z)$ may be represented in the following manner:

$$Q_o(Z) = -\frac{1}{8Z} + \sum_{k=1}^{\infty} \frac{(4k+1)^2 (4k-1)^2 (-1)}{(2k)(2k+1)(8^2 Z^2)} \; S(k-1)$$

where S(k-1) represents the preceding term in the series, i.e.,

$$S(0) = -\frac{1}{8Z}, \quad S(1) = \frac{(4(1)+1)^2 (4(1)-1)^2 (-1)}{(2(1))(2(1)+1)(8Z^2)}$$

$$S(k-1) = \frac{(4(k-1)+1)^2 (4(k-1)-1)^2 (-1)}{(2(k-1))(2(k-1)+1)(8Z^2)} \; S(k-2) \; S(k-3) \ldots .$$

Quantity $Q_o(Z)$ and its representative series are divided into real and imaginary components, Q_{N_r} and Q_{N_i} respectively, to avoid the use of complex arithmetic in the subroutine.

$$Q_o(Z) = Q_{N_r} + i\, Q_{N_i}$$

$$Q_{N_r} = -\frac{X_r}{8(X_r^2 + X_i^2)} + \sum_{k=1}^{\infty} X_{mult} \left(S_r(k-1) R_r - S_i(k-1) R_i\right)$$

$$= \sum_{k=0}^{\infty} S_r(k)$$

$$Q_{N_i} = - \frac{X_i}{8(X_r^2 + X_i^2)} + \sum_{k=1}^{\infty} X_{mult} \left(S_r(k-1)\, R_i + S_i(k-1)\, R_r\right)$$

$$= \sum_{k=0}^{\infty} S_i(k)$$

where $S(k-1) = S_r(k-1) + i\, S_i(k-1)$, $\frac{1}{Z} = \frac{X_r - iX_i}{X_r^2 + X_i^2}$

$$\frac{1}{Z^2} = R_r + i\, R_i, \quad \text{and} \quad X_{mult} = \frac{(4k-3)^2\ (4k-1)^2}{(2k)\ (2k-1)\ 64}$$

The calculations of SUBROUTINE NEUMQ0 begin with the defining of quantities RR (R_r) and RI (R_i), and the setting of initial values for SR(S_r), SI(S_i), QNR (Q_{N_r}), and QNI (Q_{N_i}). The statement DO 1 I = 1, 32000 where I corresponds to the k^{th} of the previous equation is the beginning of the evaluation of P_{N_r} and P_{N_i} as well as the series for S_r and S_i. The integer I is tested; if an even number, then integers MM= 1 and M = 2; if an odd number, then integers MM = 2 and M = 1. Then M replaces the K-1 and MM the k counter on S_r and S_i. Q_{N_r} is set equal to the sum of the last Q_{N_r} and $S_r(M)$; Q_{N_i} is set equal to the sum of the last Q_{N_i} and $S_i(M)$. The absolute value of the difference between the last two terms of the series representing S_r is tested against the convergence criterion C. If this value is greater than C, control passes to statement 6 and another term is added to the series for S_i and for S_r then control returns to the DØ statement where I is increased by 1 and the entire procedure repeated. If this value is less than or equal to C, control passes to statement 5 where a similar test is performed on the terms of S_i.

If the absolute value of the difference between the last terms in S_i is less than or equal to C, control passes to statement 7 and back to the calling subroutine. If the value is greater than C, control passes to statement 6 where the process of adding terms to the series continues.

5. Other Information

A. SUBROUTINE NEUMQ0 functions properly for pure real arguments Z; but, for complex or pure imaginary Z, an error in the definition of R_r causes erroneous results. The program under consideration uses this routine for real arguments only.

B. SUBROUTINE NEUMQ0 calls in no other subroutines or internal functions.

SUBROUTINE DIVMLT (A, B, C, D, E, F)

1. Purpose

SUBROUTINE DIVMLT is used to perform certain divisions and multiplications involving the input quantities to obtain the output quantities.

2. Input

Name	Source	Description
A	NEUMAN	input quantity used as a multiplier
B	NEUMAN	input quantity used as a multiplier
C	NEUMAN	input quantity
D	NEUMAN	input quantity

3. Output

Name	Description
E	output quantity
F	output quantity

4. Numerical Procedure

The quantity R is defined as the sum of the squares of quantities C and D. If R is zero, control passes to statement 2, where the output quantities E and F are defined as zero before the return to the calling subroutine. If R is non-zero, control passes to statement 1 and the output quantities E and F are defined as

$$E = A*(C/R) - B*(-D/R)$$

$$F = A*(-D/R) + B*(C/R)$$

5. Other Information

A. SUBROUTINE DIVMLT is called by SUBROUTINE NEUMAN.

B. SUBROUTINE DIVMLT calls in no other subroutines or functions.

SUBROUTINE NEUMN0 (ZR, ZI, XR, XI, ZNR, ZNI, C)

1. Purpose

SUBROUTINE NEUMN0 calculates the real and imaginary components of the zeroth order. Neumann representation of the Bessel function of the second kind from the power series

$$Y_o(Z) = \frac{2}{\pi}\left\{\ln\left(\frac{1}{2}Z\right) + \gamma\right\} J_o(Z) + \frac{2}{\pi}\left\{\frac{(Z^2/4)}{(1!)^2} - \left(1+\frac{1}{2}\right)\frac{(Z^2/4)^2}{(2!)^2}\right.$$

$$+\left(1+\frac{1}{2}+\frac{1}{3}\right)\frac{(Z^2/4)^3}{(3!)^2} - \ldots . (-1)^{I+1}\left(1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{I}\right)$$

$$\left.\frac{(Z^2/4)^I}{(I!)^2}\right\}$$

2. Input

Name	Description
C	convergence criterion
XI	imaginary part of zeroth order Bessel function of the first kind, J_{o_i}
XR	real part of zeroth order Bessel function of the first kind, J_{o_r}
ZI	imaginary part of argument Z of Bessel function of second kind, Z_i
ZR	real part of argument Z of Bessel function of second kind, Z_r

3. Output

Name	Description
ZNI	imaginary part of the zeroth order Bessel function of the second kind, Y_{o_i}
ZNR	real part of the zeroth order Bessel function of the second kind, Y_{o_r}

4. Numerical Procedure

The terms of the series representing $Y_o(Z)$ may be expressed in terms of its real and imaginary components, where $Z/2 = \frac{Z_r + iZ_i}{2} = (r/2)e^{i\theta} = R(\cos\theta + i\sin\theta)$, $J_o(Z) = J_{o_r} + J_{o_i}$, and $\gamma = 0.577215660$:

$$Y_o(Z) = Y_{o_r} + i\,Y_{o_i} = \frac{2}{\pi}\left\{\ln R + i\theta + \gamma\right\}\left\{J_{o_r} + i\,J_{o_i}\right\}$$

$$+ \frac{2}{\pi}\left\{\frac{R^2(\cos\theta + i\sin\theta)^2}{(1!)^2} - \left(1 + \frac{1}{2}\right)\frac{\left[R^2(\cos\theta + i\sin\theta)^2\right]^2}{(2!)^2}\right.$$

$$+ \left(1 + \frac{1}{2} + \frac{1}{3}\right)\frac{\left[R^2(\cos\theta + i\sin\theta)^2\right]^3}{(3!)^2} - \cdots\cdots$$

$$\left.(-1)^{I+1}\left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{I}\right)\frac{\left[R^2(\cos\theta + i\sin\theta)^2\right]^I}{(I!)^2}\right\})$$

Using DeMoivre's theorem $(\cos\theta + i \sin\theta)^n = \cos(n\theta) + i \sin(n\theta)$ and separating the real and imaginary parts, we obtain

$$Y_{o_r} = \frac{2}{\pi}\left\{ (\gamma + \ln R)\ J_{o_r} - \theta\ J_{o_i} + \sum_{I=1}^{\infty} (-1)^{I+1} \left(1 + \frac{1}{2} + \ldots + \frac{1}{I}\right) \frac{(\cos 2I\theta)\ R^{2I}}{(I!)^2} \right\}$$

$$Y_{o_i} = \frac{2}{\pi}\left\{ (\gamma + \ln R)\ J_{o_i} - \theta J_{o_r} + \sum_{I=1}^{\infty} (-1)^{I+1} \left(1 + \frac{1}{2} + \ldots + \frac{1}{I}\right) \frac{(\sin 2I\theta)\ R^{2I}}{(I!)^2} \right\}$$

These equations for Y_{o_r} and Y_{o_i} are evaluated in SUBROUTINE NEUMN0.

The calculations of SUBROUTINE NEUMN0 begins with the definitions of the following quantities

$$R = \sqrt{Z_r^2 + Z_i^2}\ /2.0$$

$$RLG = \ln R$$

$$THET = \theta = \tan^{-1}(Z_r/Z_i)$$

and sets the values of Y_{o_r} and Y_{o_i} initially to

$$ZNR = Y_{o_r} = 2.0\,(\gamma + \ln R)\ J_{o_r} - 2.0\ \theta\ J_{o_i}$$

$$ZNI = Y_{o_i} = 2.0\ (\gamma + \ln R)\ J_{o_i} - 2.0\ \theta\ J_{o_r}$$

The initial values of FACT and T are set to 1.0 and SR(1) and SI(1) to 1.0×10^{65}.

The statement DØ 1 I = 1, 32000, where I corresponds to the I in the summations, is the beginning of the evaluation of Y_{o_r} and Y_{o_i}. The integer I is tested; if an even number, then integers MM = 1 and M=2; if an odd number, integers MM=2 and M = 1. The quantities SR(M) and SI(M) respectively represent the terms of the series in the definitions of Y_{o_r} and Y_{o_i}. As each term of SR(M) and SI(M) is calculated it is added respectively to Y_{o_r} and Y_{o_i} until the convergence criterion is satisfied, then control passes to statement 7. To satisfy the convergence criterion the absolute difference between the term just calculated and the last series term must be less than C for both SI and SR. If the convergence tests are not satisfied I is increased by one and the process repeated.

Starting with statement 7, the final values of Y_{o_r} and Y_{o_i} are multiplied by the factor $(2.0/\pi)$, then control passes to the calling subroutine.

5. Other Information

A. SUBROUTINE NEUMN0 is called by SUBROUTINE NEUMAN.

B. SUBROUTINE NEUMN0 calls in the functions

1. ATANQR
2. DSIN
3. DCØS
4. DSQRT
5. DLØG
6. FDXPD
7. FDXPI

SUBROUTINE NEUMN1 (ZR, ZI, XR, XI, ZNR, ZNI, C)

1. Purpose

SUBROUTINE NEUMN1 evaluates the real and imaginary components of the power series solution from the first order Bessel function of the second kind (Neumann function)

$$Y_1(Z) = -\frac{2}{\pi Z} + \frac{2}{\pi}\left[\ln(1/2\,Z) + \gamma\right] J_1(Z) + \frac{1}{\pi}\sum_{I=1}^{\infty} \frac{(-1)^I A_I \left(\frac{Z}{2}\right)^{2I-1}}{I!\,(I-1)!}$$

where $A_I = 1.0$ if $I = 1$

$$A_I = 2.0\left\{\sum_{j=1}^{I-1} \left(\frac{1}{j}\right)\right\} + \frac{1}{I} \quad \text{if } I > 1$$

2. Input

Name	Symbol	Description
C	-	convergence criterion
XI	J_{1_i}	imaginary component of first order Bessel function
XR	J_{1_r}	real component of the first kind Bessel function
ZI	$X_i = r \sin\theta$	imaginary part of the argument Z
ZR	$X_r = r \cos\theta$	real part of the argument Z

3. Output

Name	Symbol	Description
ZNI	Y_{1_i}	imaginary part of first order Neumann function
ZNR	Y_{1_r}	real part of first order Neumann function

4. Numerical Procedure

The real and the imaginary parts of the power series solution for $Y_1(Z)$ are

$$Y_{1_r} = \frac{2.0}{\pi}\left(\gamma + \ln\frac{r}{2}\right) J_{1_r} - \frac{2.0}{\pi}\,\theta\, J_{1_i} - \frac{2.0}{\pi}\,\frac{\cos\theta}{r} + \frac{1.0}{\pi}\sum_{I=1}^{\infty}\frac{(-1)^I\, A_I\left(\frac{r}{2}\right)^{2I-1}\cos(2I-1)\theta}{I!\,(I-1)!}$$

$$Y_{1_i} = \frac{2.0}{\pi}\left(\gamma + \ln\frac{r}{2}\right) J_{1_i} - \frac{2.0}{\pi}\,\theta\, J_{1_r} + \frac{2.0}{\pi}\,\frac{\sin\theta}{r} + \frac{1.0}{\pi}\sum_{I=1}^{\infty}\frac{(-1)^I\, A_I\left(\frac{r}{2}\right)^{2I-1}\sin(2I-1)\theta}{I!\,(I-1)!}$$

where $\gamma = 0.57721566$

$A_I = 1.0$ if $I = 1$

$A_I = 2.0\left\{\sum_{j=1}^{I-1}\frac{1}{j}\right\} + \frac{1}{I}$ if $I > 1$

The summation of Y_{1_r} will be designated $\sum_{I=1}^{\infty} S_r(I)$ and, similarly, for Y_{1_i} the summation is $\sum_{I=1}^{\infty} S_i(I)$.

SUBROUTINE NEUMN1, which is very similar to SUBROUTINE NEUMN0, begins computations by defining the following quantities

$$R = r/2 = \sqrt{X_r^2 + X_i^2} \quad /2.0$$

$$THET = \theta = \tan^{-1} (X_i/X_r)$$

$$B = r^2 = X_r^2 + X_i^2$$

$$X_{r_B} = X_r/r^2 = \cos \theta/r$$

$$X_{i_B} = -X_i/r^2 = -\sin \theta/r$$

The output quantities ZNR and ZNI are given the following initial values

$$ZNR = 2.0 (\gamma + \ln R) J_{1_r} - 2.0 \theta J_{1_i} - 2.0 \frac{\cos \theta}{r}$$

$$ZNI = 2.0 (\gamma + \ln R) J_{1_i} - 2.0 \theta J_{1_r} + \frac{2.0 \sin \theta}{r}$$

The additional initial values assigned are

FACT = (I - 1) ! = 1.0

SERS = A_1 = 1.0 **for I = 1**

SR(1) = 1.0 x 10^{65}

SR(1) = 1.0 x 10^{65}

The DØ loop ending with statement 1 evaluates the terms SR(M) = $S_r(I)$ and SI(M) = $S_i(I)$, where M = 1 if I is an even number and M = 2 if I is odd. With each pass through the loop ZNR and ZNI are increased respectively by $S_r(I)$ and $S_i(I)$, the convergence of the series tested, and the parameters FACT and SERS defined for the next pass through the loop before I is increased. This process continues until the series converge or I reaches a value of 32000. Each series is considered to be converged when the absolute value of the difference between the current series term and the term determined on the last pass is less than C. When convergence is indicated, the current values for ZNR and ZNI are divided by π to yield the input ZNR and ZNI.

5. Other Information

A. SUBROUTINE NEUMN1 is called by SUBROUTINE NEUMAN.

B. SUBROUTINE NEUMN1 calls the functions

1. ATANQR
2. DSIN
3. DCØS
4. DSQRT
5. DLØG
6. FDXPD
7. FDXPI

3.1.7 Drag Calculations

SUBROUTINE DRAGCO performs the calculations for determining the vehicle drag coefficient after accounting for the effects of mass loss, noseblunting, and angle of attack.

SUBRØUTINE DRAGCØ

1. Purpose

SUBRØUTINE DRAGCØ is employed to calculate the total drag coefficient of the blunt or sharp cone portion of the vehicle, when the option to input the drag coefficient CDTAB is not used. In the rarefied flow regimes, the total drag coefficient is calculated as a single entity. The total drag coefficient in the continuum flow regime is the sum of the component pressure, skin friction, base, and induced drag coefficients.

2. Input

* indicates integer quantity and NØCCUR number code

Name	Symbol	Occur/Noccur Number	Source of Input	
A, 514	A_i	301-814	ZPRS	coefficients
ALPHA	α	002	DEREQ, ROTATE, VIXEN	angle of att
ALPRIM	α'	003	DEREQ, ROTATE, VIXEN	angle of att
AREF	A_{REF}	001	PRELIM	reference a
B, 21	B_i	823-843	ZPRS	coefficients free molecu
CAPL	L	010	PRELIM	sharp cone
CØDRAG	$7.6489 * 10^{-9} M_W / (\sigma \rho_{\infty})$	009	PRELIM	factor used
CØSLAM	$\lambda \cos \theta$	011	PRELIM	product of b
CPE	C_{P_e}	017	PRELIM	constant pr layer
CPW	C_{P_W}	018	PRELIM	constant pr
D	D	021	PRELIM	base diame
FACTR9	$\sqrt{\gamma}$	197	F123	numerical f drag on sph
GAMF	γ_F	026	DEREQ	flight path a
GAMMA	γ	028	SR2490 or READIT	ratio of spe
GRATE		095	VIXEN	factor used of attack ef
HSRT0	h_s/RT_o	029	PRELIM	non-dimens
HWBAR	$\bar{H}_W$	225	EVIL or PRELIM	non-dimens

rce of put	Description	Units
PRS	coefficients for curve fits, see ZPRS for more detail	-
EREQ, OTATE, XEN	angle of attack for use in $C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}$ relation	radians
EREQ, OTATE, XEN	angle of attack for use in maximum-minimum testing	radians
RELIM	reference area	ft^2
PRS	coefficients for probability distribution between free molecule and continuum flow regimes	-
RELIM	sharp cone slant length	ft.
RELIM	factor used in finding mean free path λ_W	ft.
RELIM	product of bluntness ratio and cosine of cone half angle	-
RELIM	constant pressure specific heat at edge of boundary layer	$\frac{Btu}{lbm\ ^oR}$
RELIM	constant pressure specific heat at wall	$\frac{BTU}{lbm\ ^oR}$
ELIM	base diameter	ft.
23	numerical factor used in finding free molecule drag on spherical nose	-
EREQ	flight path angle	radians
2490 or EADIT	ratio of specific heats	-
XEN	factor used in correction C_{D_P} for averaged angle of attack effect, see VIXEN	-
RELIM	non-dimensional stagnation enthalpy	-
VIL or RELIM	non-dimensional wall enthalpy	-

B

2. Input (Cont'd)

Name	Symbol	Occur/Noccur Number	Source of Input	De
LA	La	033	PRELIM	axial length
LAMDA		032	PRELIM	bluntness ra
LØPT	-	07 *	SR2490 or READIT	trajectory o
MDØT, 32	$\dot{m}_i$	2708 - 2739	EVIL	mass loss r
ME	M_e	036	PRELIM	Mach numbe
MHEAT	-	10 *	SR2490 or READIT	mass loss o
MINF	M_∞	035	PRELIM	Mach numbe
MØPT	-	03 *	READIT	mass loss o
PE	P_e	048	PRELIM	pressure at
PI	π	042	SR2490	mathematica
PINF	P_∞	049	PRELIM	free stream
REYL	$Rey_{\infty L}$	062	PRELIM	free stream cone slant le
RHØE	ρ_e	061	PRELIM	density at ed
RHØIN1	ρ_{∞_1}	056	PRELIM	free stream
RHØINF	ρ_∞	055	PRELIM	free stream
SINT	$\sin\theta$	064	CHNTBL	sine of cone
SINTM	$M_\infty \sin\theta$	067	PRELIM	product of fr cone half ang

of	Description	Units
M	axial length of vehicle	ft.
M	bluntness ratio	-
or T	trajectory option code	-
	mass loss rate distribution along body	$\frac{\text{lbm}}{\text{ft}^2\text{-sec}}$
M	Mach number at edge of boundary layer	-
or T	mass loss option code	-
M	Mach number in free stream	-
T	mass loss option code	-
M	pressure at edge of boundary layer	lb/ft^2
	mathematical constant	-
M	free stream pressure	lb/ft^2
M	free stream Reynolds number based on sharp cone slant length	-
M	density at edge of boundary layer	lbm/ft^3
M	free stream density in lbm/ft^3	lbm/ft^3
M	free stream density in slug/ft^3	slug/ft^3
L	sine of cone half angle	-
M	product of free stream Mach number and sine of cone half angle	-

B

2. Input (cont'd)

Name	Symbol	Occur/Noccur Number	Source of Input	
T	t	075	VIXEN	time for
TANT	$\tan\theta$	070	CHNTBL	tangent
TCRIT	t_{crit}	077	READIT	limit on
TE	T_e	079	PRELIM	tempera
TECON	t_{econ}	078	READIT or SR2490	limit on
THETAD	θ_D	069	PRELIM	cone hal
TIMER	t (time)	080	DEREQ	instanta
TINF	T_∞	072	PRELIM	free str
TW, 32	T_{W_i}	2644-2675	MASSLØ	wall tem
TW0	T_{W_0}	074	CHNTBL	initial w
TWST	$T_{W_{ST}}$	148	READIT or SR2490	effective molecul
V	V	082	DEREQ	velocity
VE	V_e	083	PRELIM	velocity
WDØT	$\dot{W}$	086	TØMALØ	rate of c
X1LØW	$\chi_{1_{LOW}}$	240	READIT or SR2490	value of of fairin interacti
X1UP	$\chi_{1_{UP}}$	239	READIT or SR2490	value of of fairin interacti
XBAR	$\bar{\chi}$	090	PRELIM	interacti

Source of Input	Description	Units
VIXEN	time for one complete cycle in angle of attack	sec.
CHNTBL	tangent of cone half angle	-
READIT	limit on cycle time, t_{cycle}	sec.
PRELIM	temperature at edge of sharp cone boundary layer	^{o}R
READIT or SP2490	limit on cycle time, t_{cycle}	sec.
PRELIM	cone half angle in degrees	degrees
DEREQ	instantaneous time	sec.
PRELIM	free stream temperature	^{o}R
MASSLØ	wall temperature distribution along body	^{o}R
CHNTBL	initial wall temperature	^{o}R
READIT or SR2490	effective wall temperature for use in free molecule drag calculation	^{o}R
DEREQ	velocity	ft/sec
PRELIM	velocity at edge of boundary layer	ft/sec
TØMALØ	rate of change in weight due to ablation	lb/sec
READIT or SR2490	value of rarefaction parameter which is lower boundary of fairing region between free molecule and strong interaction flow regimes	-
READIT or SR2490	value of rarefaction parameter which is upper boundary of fairing region between free molecule and strong interaction flow regimes	-
PRELIM	interaction parameter	-

B

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	I
XBAR1	$\bar{\chi}_1$	126	PRELIM	rarefactio
XBARST	$\bar{\chi}_{ST}$	089	PRELIM	factor use
XLØW	χ_{LOW}	238	READIT or SR2490	value of in boundary o and continu
XUP	χ_{UP}	237	READIT or SR2490	value of in boundary o and continu
Z	z	091	DEREQ	altitude
ZETA	ζ	093	READIT or SR2490	accomodat
ZTR	z_{TR}	092	PRELIM	transition

e of ut	Description	Units
LIM	rarefaction parameter	-
LIM	factor used in determining XBAR1	-
DIT or 0	value of interaction parameter which is lower boundary of fairing region between strong interaction and continuum flow regimes	-
DIT or 0	value of interaction parameter which is upper boundary of fairing region between strong interaction and continuum flow regimes	
EQ	altitude	ft.
DIT or 0	accomodation coefficient	-
LIM	transition altitude	ft.

B

2. Input (cont'd)

Name	Symbol	Occur/Noccur Number	Source of Input	
T	t	075	VIXEN	time f
TANT	$\tan \theta$	070	CHNTBL	tangen
TCRIT	t_{crit}	077	READIT	limit t
TE	T_e	079	PRELIM	tempe
TECON	t_{econ}	078	READIT or SR2490	limit q
THETAD	θ_D	069	PRELIM	cone h
TIMER	t (time)	080	DEREQ	instan
TINF	T_∞	072	PRELIM	free s
TW, 32	T_{W_i}	2644-2675	MASSLØ	wall te
TW0	T_{W_0}	074	CHNTBL	initial
TWST	$T_{W_{ST}}$	148	READIT or SR2490	effecti molecu
V	V	082	DEREQ	velocit
VE	V_e	083	PRELIM	velocit
WDØT	$\dot{W}$	086	TØMALØ	rate o
X1LØW	$\chi_{1_{LOW}}$	240	READIT or SR2490	value of fairi interac
X1UP	$\chi_{1_{UP}}$	239	READIT or SR2490	value of fairi interac
XBAR	$\bar{\chi}$	090	PRELIM	interac

Source of Input	Description	Units
VIXEN	time for one complete cycle in angle of attack	sec.
CHNTBL	tangent of cone half angle	-
READIT	limit on cycle time, t_{cycle}	sec.
PRELIM	temperature at edge of sharp cone boundary layer	^{o}R
READIT or SR2490	limit on cycle time, t_{cycle}	sec.
PRELIM	cone half angle in degrees	degrees
DEREQ	instantaneous time	sec.
PRELIM	free stream temperature	^{o}R
MASSLØ	wall temperature distribution along body	^{o}R
CHNTBL	initial wall temperature	^{o}R
READIT or SR2490	effective wall temperature for use in free molecule drag calculation	^{o}R
DEREQ	velocity	ft/sec
PRELIM	velocity at edge of boundary layer	ft/sec
TØMALØ	rate of change in weight due to ablation	lb/sec
READIT or SR2490	value of rarefaction parameter which is lower boundary of fairing region between free molecule and strong interaction flow regimes	-
READIT or SR2490	value of rarefaction parameter which is upper boundary of fairing region between free molecule and strong interaction flow regimes	-
PRELIM	interaction parameter	-

B

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	Descr
XBAR1	$\bar{\chi}_1$	126	PRELIM	rarefaction para
XBARST	$\bar{\chi}_{ST}$	089	PRELIM	factor used in d
XLØW	χ_{LOW}	238	READIT or SR2490	value of interact boundary of fair and continuum fl
XUP	χ_{UP}	237	READIT or SR2490	value of interact boundary of fair and continuum fl
Z	z	091	DEREQ	altitude
ZETA	ζ	093	READIT or SR2490	accomodation co
ZTR	z_{TR}	092	PRELIM	transition altitud

rce of nput	Description	Units
ELIM	rarefaction parameter	-
ELIM	factor used in determining XBAR1	-
ADIT or 2490	value of interaction parameter which is lower boundary of fairing region between strong interaction and continuum flow regimes	-
ADIT or 2490	value of interaction parameter which is upper boundary of fairing region between strong interaction and continuum flow regimes	
REQ	altitude	ft.
ADIT or 2490	accomodation coefficient	-
ELIM	transition altitude	ft.

B

3. Output

Name	Symbol	Occur Number	D
CD	C_D	016	total drag co
CDB	C_{D_B}	099	base drag co
CDFINF, 8	$C_{D_{f_\infty}}$	2793-2800	skin friction
CDI	C_{D_I}	100	total induced
CDP	C_{D_P}	098	pressure dra averaged α
CDP0	$C_{D_{P_0}}$	101	pressure dra
DECDFP	$(\Delta C_{D_f})_P$	235	pressure indu
DECFTC	$(\Delta C_{D_f})_{TC}$	236	transverse cu coefficient
DELCDP	ΔC_{D_P}	234	induced pres
TZTEST	$t_{ZTR} + t_f$	246	sum of last ti in fairing reg

Description	Units
total drag coefficient	-
base drag coefficient	-
skin friction drag coefficient	-
total induced drag coefficient	-
pressure drag coefficient corrected for a non-averaged α	-
pressure drag coefficient for $\alpha = 0$	-
pressure induced skin friction drag coefficient	-
transverse curvature induced skin friction drag coefficient	-
induced pressure drag coefficient	-
sum of last time in fully laminar flow and total time in fairing region between laminar and turbulent flow	sec.

B

4. Numerical Procedure

The computationsof SUBROUTINE DRAGCO may be grouped in the following manner to facilitate description of the numerical calculations:

A. Miscellaneous Definitions

B. Calculation of $C_{D_{P_\alpha}}$

1) evaluation of equation for $C_{D_{P_{\alpha=0}}}$

2) evaluation of equation for $(C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}})$ ratio

3) product of 1) and 2) is $C_{D_{P_\alpha}}$

C. Rarefied Flow Drag Coefficient Calculations

1) strong interaction drag coefficient

2) free molecule drag coefficient

3) fairing between free molecule and strong interaction flow regimes and definition of total drag coefficient.

D. Continuum Flow Drag Coefficient Calculations

1) Base Drag Coefficient

2) Turbulent Calculations

a) evaluation of turbulent induced and skin friction drag coefficients.

b) determination of averaged α effect

c) computation for fairing between laminar and turbulent flow regions

d) definition of total drag coefficient

3) Laminar calculations

a) evaluation of laminar induced and skin friction drag coefficients

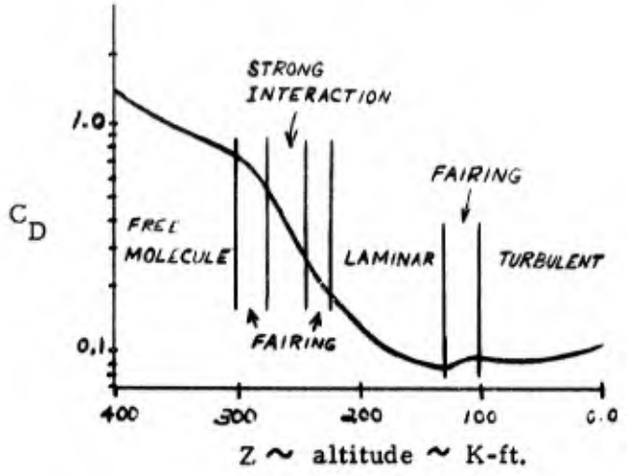
b) determination of averaged α effect

c) definition of total drag coefficient in fully laminar flow

d) definition of total drag coefficient in fairing region between strong interaction and laminar flow regimes.

The flow regimes are defined by the values of the interaction parameter $\bar{\chi}$ and the rarefaction parameter $\bar{\chi}_1$ in the following way:

Typical C_D Curve with Flow Regions



$\chi_1 > \chi_{1_{UP}}$ transitional free molecule flow

$\chi_{1_{UP}} > \chi_1 \geq \chi_{1_{LOW}}$ fairing between free molecule and strong interaction flow

$\chi_{1_{LOW}} > \bar{\chi}_1$ and $\bar{\chi} > \chi_{UP}$ strong interaction flow regime

$\chi_{UP} \geq \bar{\chi} \geq \chi_{LOW}$ fairing region between strong interaction flow and continuum laminar flow region

$\chi_{LOW} > \bar{\chi}$ and $Z \geq Z_{TR}$ fully laminar flow

$\chi_{LOW} > \bar{\chi}$, $Z < Z_{TR}$, and time $< t_{Z \sim ZTR} + t_F$ fairing between laminar and turbulent regions.

$\chi_{LOW} > \bar{\chi}$, $Z < Z_{TR}$, and time $\geq t_{Z \sim ZTR} + t_F$ fully turbulent flow

It is conceivable that the above definitions may not be sufficient to define the flow regime unambiguously in certain unusual cases; i.e., $\bar{\chi}_1 > \chi_{1_{LOW}}$ but $\chi_{UP} \geq \bar{\chi}$. Therefore, the following priorities are observed:

1) If $\bar{\chi}_1 > \chi_1$ but $\chi_{UP} \geq \bar{\chi}$, the fairing between free molecule and strong interaction regimes overrides and the value of χ_{UP} is reset so that the $\chi_{UP} \sim \bar{\chi}_{\bar{\chi}_1 \sim \chi_{1_{LOW}}}$ and fairing into continuum flow begins just after $\bar{\chi}_1$ reaches $\chi_{1_{LOW}}$.

2) If $\bar{\chi} > \chi_{LOW}$ but $Z < Z_{TR}$, the fairing between laminar flow and strong interaction region overrides then when $\bar{\chi}$ reaches χ_{LOW}, the fairing of laminar into turbulent flow begins.

3) If $\bar{\chi} < \chi_{LOW}$ for any value of $\bar{\chi}_1$, appropriate continuum calculations are performed depending on the value of Z tested against Z_{TR}.

The following description of SUBRØUTINE DRAGCØ is presented in accordance with the outline of the introductory comments.

A. Miscellaneous Definitions

The following quantities to be used in drag calculations are defined:

PEPINF = P_e/P_∞

SQRT3 = 1.732050807568877

TEMLAM = 1.0 - (λ cos θ) (λ cos θ)

$F1_K$ = 0.9 + M_∞ sin θ (-0.119 + 0.0108 M_∞ sin θ)

α_D = 180.0 α / π

B. Calculation of $C_{D_{P_a}}$

The pressure drag coefficient at zero angle of attack is then determined from the curve fit expression as follows:

if $\theta_D > 20.0^\circ$ N = 312

if $20.^\circ \geq \theta_D > 10.^\circ$ N = 247

if $\theta_D \leq 10^\circ$ N = 211

$$C_{D_{P_{a=0}}} = \sum_{II=0}^{2} \sum_{JJ=0}^{2} \sum_{KK=0}^{3} A_{N+II+3JJ+9KK} \left(\frac{1}{M_\infty}\right)^{II} (\lambda)^{JJ} \left(\frac{1}{\theta_D}\right)^{KK}$$

A sharp cone pressure drag coefficient is then determined for the given cone half angle from

$$C_{D_{P_{\lambda=0}}} = \frac{2.0\, P_e}{\rho_\infty V^2}$$

The value of angle of attack is then tested; if the absolute value of $\alpha_D > 4.0^\circ$, control passes to statement 220. If the absolute value of $\alpha_D \leq 4.0^\circ$ the following equation is evaluated to obtain the angle of attack correction ratio $C_{D_{P_a}} / C_{D_{P_{a=0}}}$. The ratio $C_{D_{P_a}} / C_{D_{P_{a=0}}}$ is set equal to 1.0, then, if α_D is zero, control passes to statement 230.

If $4.0 \geq |\alpha_D| > 0.0$, the following equation is evaluated to obtain the ratio $C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}$:

$$C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}} = \sum_{II=0}^{3} \sum_{JJ=0}^{2} \sum_{KK=0}^{2} A_{421+II+4JJ+12KK} (\theta_D)^{II} (|\alpha_D|)^{JJ} (\lambda)^{KK}$$

If the equation yields a value less than 1.0, $C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}$ is set equal to 1.0. Then control passes to statement 230.

Statement 220 is the beginning of the computation for $C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}$ when $|\alpha_D| > 4.0^\circ$. The quantity TEML is set equal to the absolute value of α_D; however, if TEML is greater than 40.0, it is set equal to 40.0. The following parameters are then defined:

$$X = \log_{10} \theta_D$$

$$Y = \log_{10} (TEML)$$

$$ZZ = \lambda$$

and used to evaluate the following curve fit equation

$$FUN = \log_{10} (C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}) = \sum_{II=0}^{3} \sum_{JJ=0}^{2} \sum_{KK=0}^{2} A_{348+II+4JJ+12KK} X^{II} Y^{JJ} ZZ^{KK}$$

If FUN is less than 0.0, it is set equal to 0.0. From the above, we obtain the ratio

$$C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}} = (10.)^{FUN}$$

Following statement 230, the pressure drag coefficient at zero angle of attack is multiplied by the angle of attack correction ratio $C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}$ to obtain the pressure drag coefficient with non-averaged angle of attack effects.

If the interaction parameter, $\bar{\chi}$, has a value less than χ_{LOW} control passes to statement 1, the beginning of the continuum calculations. Otherwise, the rarefied flow calculations are performed.

C. Rarefied Flow Calculations

The following definitions are made preparatory to computing the sharp cone strong interaction drag coefficient:

$$T_o = T_\infty (1.0 + 0.5 (\frac{\gamma - 1.0}{2}) M_\infty^2)$$

$$TEM = T_{W_{ST}} / T_o$$

$$TEM3 = \log_e (\bar{\chi}_{ST})$$

If $\theta_D < 15.0^\circ$, control passes to statement 204; if not, the following equations are used to evaluate the sharp cone strong iteraction drag coefficient:

$$SUM = \sum_{II=0}^{1} \sum_{JJ=0}^{3} \sum_{KK=0}^{1} A_{384+II+2JJ+8KK} (TEM3)^{II} (\theta_D)^{JJ} (TEM)^{KK}$$

$$CDST = C_{D_{ST}} = e^{SUM}$$

Control then passes to statement 3. In a similar fashion, $C_{D_{ST}}$ is determined for values of $\theta_D \geq 15.0^\circ$ beginning with statement 204.

$$SUM = \sum_{II=0}^{1} \sum_{JJ=0}^{1} \sum_{KK=0}^{1} A_{200+II+2JJ+4KK} (TEM3)^{II} (\theta_D)^{JJ} (TEM)^{KK}$$

$$CDST = C_{D_{ST}} = e^{SUM}$$

Statement 3 is the beginning of the drag calculations for the transitional-free molecule flow regime, where the mean free path of the molecules is of the same order as a typical body dimension. The sharp cone transitional-free molecule drag coefficient is obtained from a probabilistic model which takes into account the continuous variation in aerodynamic properties between continuum and free molecule flow regimes. The free molecule drag coefficient, a continuum drag coefficient based on Newtonian results, and the probability function indicated below are combined to yield the sharp cone transitional-free molecule drag coefficient.

$$CDF = C_{D_{FM}} = \left\{ e^{-M_\infty^2 \sin^2\theta} \left(\frac{1.0}{M_\infty \sin\theta \sqrt{\pi}} + \frac{0.5}{M_\infty^2} \sqrt{\frac{T_{W_{ST}}}{T_\infty}} \right) + 2.0 + \frac{1}{M_\infty^2} + \frac{\sin\theta}{M_\infty} \sqrt{\pi \frac{T_{W_{ST}}}{T_\infty}} \right\}$$

$$CDN = C_{D_N} = 2.0 \sin^2 \theta$$

$$LAMWD = \frac{\lambda_W}{D} = \frac{C\emptyset DRAG}{M_\infty D} \sqrt{\frac{2.25\pi}{\gamma} \frac{T_{W_{ST}}}{T_\infty}}$$

$$\frac{\lambda_W}{D} \geq 0.04 \qquad CAPP = P = \sum_{I=1}^{21} B_I \left(\log_e \left(\frac{\lambda_W}{D}\right)\right)^{I-1}$$

$$\frac{\lambda_W}{D} < 0.04 \qquad CAPP = P = 0.506 - 0.147 \log_{10} \left(\frac{0.04}{\lambda_W/D}\right)$$

$$CDTRFM = C_{D_{Trans.-FM}} \Big|_{sharp} = P(C_{D_{FM}} - C_{D_N}) + C_{D_N}$$

Henceforth, the transitional - free molecule flow regime will be referred to merely as the free molecule regime for the sake of brevity.

The bluntness ratio is tested and a value less than 10^{-3} passes control to statement 30, where the total free molecule drag, CDFM, is set equal to the sharp cone value, before continuing to statement 32. If $\lambda \geq 10^{-3}$, the contribution of the spherical nose in free molecule flow is calculated and included in the calculation for total C_D in free molecule flow:

$$TEM = \log \left[\frac{Rey_{\infty_L} D}{L \; M_\infty FAC \; R9}\right]$$

if $|TEM| \leq 10^{-4}$ then TEM is set equal to 10^{-4}.

if TEM > 5.298 then TEM is set equal to 5.298

$$CDFMS = C_{D_{FM_{sphere}}} = 2.0 \sum_{J=1}^{20} A_{173+J} \, TEM^{J-1}$$

$$CDFM = C_{D_{FM_{total}}} = C_{D_{FM_{sphere}}} \lambda^2 \cos^2\theta + C_{D_{Trans-FM_{sharp}}} (1.0 - \lambda^2)$$

Control these passes to statement 32 where the sharp cone strong interaction C_D is corrected for bluntness and angle of attack effects.

$$CDS = C_{D_S} = (C_{D_{ST}} - C_{D_{P_{\lambda=0}}}) (1.0 - \lambda^2) \cos \alpha' + C_{D_{P_\alpha}}$$

This is followed by the defining of the total drag coefficient according to the regime indicated from the testing of $\bar{\chi}$ and $\bar{\chi}_1$

If $\bar{\chi}_1 \geq \chi_{1_{UP}}$ $C_D = C_{D_{FM_{total}}}$

If $\bar{\chi}_1 < \chi_{1_{UP}}$, $\bar{\chi}_1 < \chi_{1_{LOW}}$, and $\bar{\chi} > \chi_{UP}$ $C_D = C_{D_S}$

If $\bar{\chi}_1 < \chi_{1_{UP}}$, $\bar{\chi}_1 < \chi_{1_{LOW}}$, and $\bar{\chi} \leq \chi_{UP}$ control passes to the continuum flow calculations following statement 1.

If $\bar{\chi}_1 < \chi_{1_{UP}}$, $\bar{\chi}_1 \geq \chi_{1_{LOW}}$, the following procedure is used to fair between the free molecule and strong interaction values.

$$C_D = C_{D_{FM_{total}}} \left(\frac{\bar{\chi}_1 - \chi_{1_{LOW}}}{\chi_{1_{UP}} - \chi_{1_{LOW}}} \right) + C_{D_S} \left(1.0 - \frac{\bar{\chi}_1 - \chi_{1_{LOW}}}{\chi_{1_{UP}} - \chi_{1_{LOW}}} \right)$$

Then, if $\bar{\chi}_1 > \chi_{UP}$, control returns to the calling subroutine. If not, χ_{UP} and χ_{LOW} are redefined and respectively set equal to $\bar{\chi}$ and $\bar{\chi} - 2.0$ with the restriction that χ_{LOW} cannot be smaller than 0.5. This means that χ_{UP} and χ_{LOW} are reset until $\bar{\chi}_1 < \chi_{1_{LOW}}$ and that they retain the values given at the altitude just before $\bar{\chi}_1$ becomes less than $\chi_{1_{LOW}}$.

D. Continuum Flow Calculations

The continuum flow calculations begin with statement 1 and the evaluation of the base drag coefficient which is a function of the Mach number and the non-dimensionalized rate of change in weight due to ablation.

$$Y = |\dot{W}| / (\rho_{\infty_1} V A_{ref})$$

unless $Y < 0.0$, then $Y = 0.0$ or if $Y > 0.06$, $Y = 0.06$

$$X = 1.0 / M_\infty$$

unless $M_\infty > 24.0$, then $X = 1.0/24.0$

if $M_\infty < 7.0$, N = 487

if $M_\infty \geq 7.0$, N = 499

The base pressure ratio P_b/P_∞ is then found from

$$PBPINF = P_b/P_\infty = \sum_{II=0}^{3} \sum_{JJ=0}^{2} A_{N+II+4JJ} \; X^{II} \; Y^{JJ}$$

which is used in determining base pressure drag coefficient, C_{D_B},

$$CDB = C_{D_B} = (1.0 - P_b/P_\infty) / (0.7 M_\infty^2)$$

The interaction parameter, $\bar{\chi}$, is tested against χ_{LOW} to determine if full continuum flow conditions have been reached or if the appropriate flow regime is that determined by the fairing between continuum and strong interaction regions. If $\bar{\chi} \geq \chi_{LOW}$ indicating the fairing region, control passes to statement 10 and the laminar flow calculations. If $\bar{\chi} < \chi_{LOW}$ full continuum flow, Z is tested against Z_{TR}. If $Z \geq Z_{TR}$, control passes to statement 10 and laminar flow calculations; if $Z < Z_{TR}$, subroutine proceeds with the evaluation of the turbulent equations.

In the continuum calculations, the skin friction drag coefficient is represented by the following array:

CDFINF(I, J, K) where

I = 1 sharp cone, I = 2 blunt cone

J = 1 turbulent, J = 2 laminar

K = 1 with blowing, K = 2 without blowing

Turbulent Flow Calculations

The flow properties needed for the computation of skin friction and induced drag are determined from subroutine inputs by the following relationships:

$$h_e = C_{p_e} T_e$$

$$\frac{h^*}{h_e} = 0.5 + \frac{0.5\, C_{p_w} T_{w_{i=8}}}{h_e} + 0.099\, M_e^2 (\gamma - 1.0)$$

$$h^* = (h^*/h_e)\ h_e$$

If the reference enthalpy $h^* \leq 1110.0$ Btu/lbm. control passes to statement 21, where T^* is set equal to $3.5964\, h^*$. If $h^* > 1110.$ Btu/lbm, the reference temperature, T^*, is defined in the following manner:

$$TEM = P_e / 2116.0$$

If $TEM \geq 10$, then $TEM = 10.0$

$$T^* = \sum_{II=0}^{3} \sum_{JJ=0}^{2} A_{163+II+4JJ} (h^*)^{II} (TEM)^{JJ}$$

If the value of $T^* < 0.0$, control passes to EXIT. For positive or zero T^*, the remaining reference flow properties are defined:

$$\mu^* = 32.2 \left[\frac{2.27 \times 10^{-8} (T^*)^{1.5}}{T^* + 198.6} \right]$$

$$\tau = \frac{T^*}{1.8} \left[1.0 - 0.125 \log_{10} (Pe/2116.) \right]$$

$$z_T = 2.5 + 0.1 \tanh\left(\frac{T}{500.0} - 7.0\right) + 0.4 \tanh\left(\frac{T}{1000.} - 7.0\right)$$

$$+ \tanh\left(\frac{T}{2500.} - 5.8\right)$$

$$\rho^* = 39.65 \left(\frac{P_e}{2116.0 \; z_\uparrow \; T^*}\right)$$

If L, sharp cone slant length, is less than 2.0 feet:

$$A_{118} = -4.4666$$

$$A_{119} = 156.0$$

$$A_{120} = -665.0$$

If $L \geq 2.0$ feet

$$A_{118} = \sum_{II=0}^{6} A_{400+II} \; L^{II}$$

$$A_{119} = \sum_{II=0}^{6} A_{407+II} \; L^{II}$$

$$A_{120} = \sum_{II=0}^{6} A_{414+II} \; L^{II}$$

These coefficients are used to evaluate the quantity TEM2, which is used in conjunction with REYSTA to define the sharp cone skin friction drag coefficient, CDFINF(1, 1, 2)

$$TEM2 = A_{120} + \log_{10}\left(\frac{\rho^* V_e}{\mu^*}\right)\left[A_{119} + A_{118}\ \log_{10}\left(\frac{\rho^* V_e}{\mu^*}\right)\right]$$

$$REYSTA = Rey_{*L} = \frac{\rho^* V_e L}{\mu^*}$$

$$CF0 = C_{f_0} = 0.37\,(1.15)\ \frac{\rho^*}{\rho_e}\ \frac{1.0}{\left[\log_{10}(Rey_{*L})\right]^{2.584}}$$

$$CDFINF(1,1,2) = C_{D_{f_\infty}}\ \text{sh, turb, no blowing}$$

$$= 0.5\,(0.852)\ \frac{\rho^*\ V_e}{\rho_\infty V^2 \tan\theta\ TEM2}$$

Correcting the skin friction C_D for blowing we obtain

$$CDFINF(1,1,1) = CDFINIF(1,1,2)\ /\ \left[1.0 + \frac{\dot{m}_i = 8}{\rho_e V_e C_{f0}}\right]$$

and, in addition, for bluntness effects with blowing

$\lambda < 0.32$

$$CDFINF(2,1,1) = CDFINF(1,1,1)\left[1.0 - (0.80 + 0.052\ M_\infty)\right]$$

$$= C_{D_{f_\infty}}\ \text{bl, turb, w.b.}$$

$\lambda \geq 0.32$

$$CDFINF(2,1,1) = CDFINF(1,1,1)\ (0.744 - 0.1664\ M_\infty)$$

$$= C_{D_{f_\infty}}\ \text{bl, turb, w.b.}$$

Using these results, the turbulent blunt, no blowing $C_{D_{f_\infty}}$ is determined from

$$\text{CDFINF}(2, 1, 2) = \text{CDFINF}(2, 1, 1) * \frac{\text{CDFINF}(1, 1, 2)}{\text{CDFINF}(1, 1, 1)}$$

Next, the induced drag coefficient, which for turbulent flow consists only of the induced pressure drag coefficient, is computed. First, H_W is set equal to $\bar{H}_W$, the non-dimensional wall enthalpy; however, if no mass loss calculations have been performed, $\bar{H}_W$ is undefined and H_W is set to $(0.24\ T_W\ /\ 33.86)$. The recovery enthalpy is calculated, as $H_r = 0.9\ h_s/RT_o$, for use in ratio the ratio $h_W/\ h_r$ and TEM is defined as $(\rho_e\ V_e\ C_{f_e})$

Using the above, to find induced effects

$$\text{DTHE1} = \Delta\theta_1 = \frac{C_{fo}}{3.8}\left(2.0 + \frac{\dot{m}_{i=8}}{\text{TEM}} + \frac{1.0}{1.0 + 1.2\,\dot{m}_{i=8}/\text{TEM}}\right)$$

$$\text{DTHE2} = \Delta\theta_2 = \Delta\theta_1\left(0.547\frac{h_w}{h_r} + M_e\left(0.53 + 0.68\ \frac{h_w}{h_r}\right)\right.$$

$$\left. + M_e^{\,2}\left(0.083 + 0.106\ \frac{h_w}{h_r}\right)\right)$$

$$\text{DTHE} = \Delta\theta = \Delta\theta_2 + \frac{1.6\,\dot{m}_{i=8}}{3.6\,\rho_e\,V_e}$$

$$\text{CDLCDP} = \Delta C_{D_P} = 1.11(1.0 + \lambda^2\cos^2\theta)\ C_{D_{P_{\lambda=0}}}\ \Delta\theta\ F_{1_K}\ \sqrt{3}\ M_e$$

$$\text{CDI} = C_{D_I} = \text{CDLCDP} = \Delta C_{D_P}$$

The contribution to the total C_D of C_{D_I} and $C_{D_{f_\infty}}$, CDSUM, is defined as the sum of the turbulent induced and skin friction drag coefficients, LØPT $\geq$ 3, or, if LØPT < 3, when TIMER $\geq$ TZTEST (no fairing is done under these circumstances representing fully turbulent flow). If LØPT < 3 and TIMER < TZTEST the contribution of skin friction and induced C_D's is a value, CDSUM, faired between the laminar and turbulent values. The components of the following fairing equations are defined in the laminar drag calculations of this subroutine, and correspond to laminar values at the last altitude before Z becomes $< Z_{TR}$.

$$PBAR = \bar{P} = \left[C_{fo_{ZTR}} \frac{La}{1.5} \left(1.0 - \frac{t - t_{ZTR}}{t_f}\right)^{1.5} - C_{f_o} \frac{La^{1.6}}{1.8} \left(1.0 - \frac{t - t_{ZTR}}{t_f}\right)^{1.8} \right] \Big/ \left[C_{fo_{ZTR}} \frac{La}{1.5} - C_{f_o} \frac{La^{1.6}}{1.8} \right]$$

$$C_{D_I} = C_{D_{I_{LAM}}} \bar{P} + (1.0 - \bar{P}) C_{D_{I_{TURB}}}$$

$$C_{D_{f_{\infty}\,BL,\,TURB,WB}} = C_{D_{f_{\infty}\,BL,\,LAM,\,WB}} * \bar{P} + (1.0 - \bar{P}) * C_{D_{f_{\infty}\,BL,\,TURB,\,WB}}$$

$$C_{D_{f_{\infty}\,BL,\,TURB,\,NB}} = C_{D_{f_{\infty}\,BL,\,LAM,\,NB}} * \bar{P} + (1.0 - \bar{P}) * C_{D_{f_{\infty}\,BL,\,TURB,\,NB}}$$

Then the contribution to total drag is

$$\text{CDSUM} = C_{D_{SUM}} = C_{D_I} + C_{D_{f_\infty}} \text{ BL, TURB, WB}$$

Tests of T, time for one cycle, are made with quantities TCRIT and TECØN to determine whether the pressure drag component of total drag coefficient should be corrected by the averaged angle of attack effect or by the ratio $(C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}})$.

If TCRIT $\leq$ T or if TCRIT > T, but T < TECØN, then

$$C_D = \text{total drag coefficient} = C_{D_{SUM}} + C_{D_B} + C_{D_{P_\alpha}}$$

otherwise,

$$C_D = C_{D_{SUM}} + C_{D_B} + C_{D_{P_{\alpha=0}}} (1.0) + \frac{\text{GRATE}}{T})$$

Then the return to the calling subroutine is executed.

Laminar Flow Calculations

The ratio of reference enthalpy to sharp cone edge enthalpy, h^*/h_e, is computed prior to evaluating the sharp cone, laminar, no blowing skin friction drag.

$$\text{HSTAHE} = \frac{h^*}{h_e} = 0.5 + 0.5 \frac{C_{P_W} T_{W_{i=8}}}{C_{P_e} T_e} + 0.0935 (\gamma - 1.0) M_e^2$$

$$\text{CDFINF}(1, 2, 2) = C_{D_{f_\infty}} \text{ sh, LAM, N.B.}$$

$$= \frac{1.53}{\tan\theta} \sqrt{\frac{P_e}{P_\infty \, Rey_\infty \, L}} \left(\frac{V_e}{V}\right)^{1.5} \left(\frac{h^*}{h_e} \frac{C_{P_e} \, T_e}{0.2398 \, T_\infty}\right)^{-0.185}$$

Correcting for the effects of blowing

$$CF0 = C_{f_0} = \frac{1.15}{1.53} \tan\theta \, C_{D_{f_\infty \text{ sh, lam, N.B.}}} * \frac{\rho_{\infty 1} \, V^2}{\rho_e \, V_e^2}$$

$$CDFINF(1, 2, 1) = C_{D_{f_{\infty \text{ sh, lam, WB}}}} = C_{D_{f_{\infty \text{ sh, lam, N.B.}}}} \Big/ \left[1.0 + \frac{2.0 \, \dot{m}_{i=8}}{\rho_e \, V_e \, C_{f0}}\right]$$

and then for bluntness with blowing

$\lambda \geq 0.2 \quad N = 472$

$\lambda < 0.2 \quad N = 457$

$$CDFINF(2, 2, 1) = C_{D_{f_{\infty \text{ bl, LAM, WB}}}} = C_{D_{f_{\text{ sh, LAM, WB}}}} * \left[\sum_{JJ=0}^{2} \sum_{II=0}^{4} A_{N+5JJ+II} \, \lambda^{JJ} \, (\log_{10} Rey_\infty \, L)^{II}\right]$$

and without blowing

$$\text{CDFINF}(2,2,2) = C_{D_f}\Big|_{\text{bl, lam, N.B.}}$$

$$= \text{CDFINF}(2,2,1) * \frac{\text{CDFINF}(1,2,2)}{\text{CDFINF}(1,2,1)} \quad .$$

In laminar flow, the induced drag coefficient has three components - the induced pressure (DELCDP), pressure induced skin friction (DECDFP), and the transverse curvature induced (DECFTC) components. These are determined as indicated below from the wall and recovery enthalpies and the edge properties:

$H_W = \bar{H}_W$, or if no mass loss is being calculated

$H_W = (0.24\ T_{w_{i=8}})/33.86$

$h_r = 0.9\ h_s$

$\text{TEM} = \rho_e\ V_e\ C_{f_0}$

$$\text{DTHE1} = \Delta\theta_1 = C_{f_0}\left(\frac{2.0\ \dot{m}_{i=8}}{\text{TEM}} + \frac{1.0}{1.0 + 1.25\ \dot{m}_{i=8}/\text{TEM}}\right)/\left(2\sqrt{3}\right)$$

$$\text{DTHE2} = \Delta\theta_2 = \Delta\theta_2\ (A_{283} + A_{284}\ M_e^2 + \frac{h_W}{h_r}\ (A_{285} + A_{286}\ M_e^2 +$$

$$\frac{h_W}{h_r}\ (A_{287} + A_{288}\ M_e^2 + \frac{h_W}{h_r}\ (A_{289} + A_{290}\ M_e^2))))$$

$$\text{DTHE} = \Delta\theta = \Delta\theta_2 + \frac{\dot{m}_{i=8}}{3.0\ \rho_e\ V_e}$$

$$\text{DELCDP} = \Delta C_{D_P} = 1.33(1.0 - \lambda^2 \cos^2\theta)\, C_{D_{P_{\lambda=0}}}\, \Delta\theta\, F_{1_K}\, \sqrt{3}\, M_e$$

$$\text{DECDFP} = \left(\Delta C_{D_{f_\infty}}\right)_P = 1.5\,\Delta\theta(1.0 - \lambda^2 \cos^2\theta)\, C_{D_f}\, F_{1_K}$$

$$\left[-0.823 + 0.524\, \frac{T_{w_{i=8}}}{T_e} + \frac{0.438\, M_e}{\sqrt{L}}\right]$$

$$\text{DINF} = d_\infty = 0.058 + \frac{0.968\, T_{w_{i=8}}}{T_\infty\, M_\infty^2}$$

$$\text{DECFTC} = \left(\Delta C_{D_{f_\infty}}\right)_{TC} = 1.5\,(0.517 + 0.913\, \frac{T_{w_{i=8}}}{T_e} + 0.0484\, M_e^2)$$

$$((1 - \lambda^2 \cos^2\theta)\,\Delta\theta\, C_{D_{f_\infty}})$$

$$/\,(M_e^2 \tan\theta\, d_\infty \sqrt{3L})$$

$$\text{CDI} = C_{D_I} = \Delta C_{D_P} + \left(\Delta C_{D_{f_\infty}}\right)_P + \left(\Delta C_{D_{f_\infty}}\right)_{TC}$$

If the trajectory option code, LØPT, is less than 3, the quantities needed for the fairing equation with turbulent flow are defined as:

$$CF0ZTR = C_{f_{0_{ZTR}}} = C_{f_0}\Big|_{LAM}$$

$$TZTR = t_{Z \sim ZTR}$$

$$TF = t_f = 1.0 \; / \; (0.48667 \times 10^{-4} \; V \; |\sin \gamma_f|)$$

$$CDILAM = C_{D_I}\Big|_{LAM}$$

$$TZTEST = t_{Z \sim ZTR} + t_f$$

If $L\emptyset PT \geq 3$, the above definitions are bypassed. Next $\bar{\chi}$ is tested, if $\geq \chi_{LOW}$ control passes to statement 33 where the equation for fairing into the strong interaction flow is evaluated. If $\bar{\chi}$ is $< \chi_{LOW}$, indicating fully laminar flow, the angle of attack cycle time, T, is tested against TCRIT and TEC$\emptyset$N. This test is made in order to determine whether the angle of attack effects on C_{D_P} are to be found from the averaged effect or from the ratio $C_{D_{P_\alpha}} / C_{D_{P_{\alpha=0}}}$. If TCRIT $\leq$ T, or if TCRIT > T, but T < TEC$\emptyset$N, the ratio correction is used and the total drag coefficient is obtained from

$$CD = C_D = C_{D_B} + C_{D_I} + C_{D_{f_{\infty \text{ bl, lam, WB}}}} + C_{D_P}$$

before returning to calling subroutine.

If, on the other hand, TCRIT > T, but T ≥ TECØN, the averaged effect is used and total drag coefficient is

$$CD = C_D = C_{D_B} + C_{D_I} + C_{D_{f_{\infty \text{ bl, lam, WB}}}} + C_{D_{P_{\alpha = 0}}} \frac{(1.0 + GRATE)}{T}$$

Then the return to calling subroutine is executed.

Statement 33 begins the evaluation of the total drag coefficient in the fairing region, i. e.

$$CDLAM = C_{D_{LAM}} = C_{D_B} + C_{D_I} + C_{D_P} + C_{D_{f_{\infty \text{bl, lam, WB}}}}$$

$$CD = C_D = C_{D_S} \left(\frac{\bar{\chi} - \chi_{LOW}}{\chi_{UP} - \chi_{LOW}} \right) + C_{D_{LAM}} \left(1.0 - \frac{\chi - \chi_{LOW}}{\chi_{UP} - \chi_{LOW}}\right.$$

before returning to the calling subroutine.

5. Other Information

A. SUBRØUTINE DRAGCØ is called in by SUBRØUTINE VIXEN or by SUBRØUTINE DEREQ.

B. SUBRØUTINE DRAGCØ calls the library functions

1. EXIT
2. DEXP
3. DSIN
4. DCØS
5. DSQRT
6. DLØG
7. DLØG10
8. DTANH
9. FDXPD
10. FDXPI

3.1.8 Derivatives of the Translational Parameters

SUBROUTINE TEQUAT calculates the derivatives of the translational trajectory parameters with the assistance of SUBROUTINE MATMPY which performs matrix multiplication.

SUBROUTINE TEQUAT (DERIV)

1. Purpose

Subroutine TEQUAT calculates the derivatives of the particle trajectory parameters and of the thrusting parameters indicated below:

DERIV(1) = $\dot{V}$

DERIV(2) = $\dot{\gamma}_f$

DERIV(3) = dt/dz = d(time) / d (altitude)

DERIV(4) = $\dot{X}_r$

DERIV(14) = $\dot{Y}_r$

DERIV(15) = $\dot{\psi}_a$

DERIV(16) = $\dot{W}_{th}$

Name	Symbol	Occur/Noccur Number	Source of Input	De
ALPRIM	α'	003	ROTATE, DEREQ or VIXEN	instantane
AREF	A_{REF}	001	PRELIM	instantane
CAPG	G	019	SR2490 or READIT	gravitatio
CD	C_D	016	DEREQ or DRAGCØ	total drag
CN	C_n	203	ROTATE	normal fo
FTMAT, 3	F_{X_T}, F_{Y_T}, F_{Z_T}	-	MATMPY	componen system
GAMF	γ_f	026	DEREQ	instantane
ISP	$I_{S.P.}$	222	READIT	input spec
M	m	037	PRELIM	instantane
PHI	ϕ	044	DEREQ	Euler angl
PSI	ψ	045	DEREQ	Euler angl
PSIALP	ψ_a	200	DEREQ	thrust mis
QD	q_D	051	PRELIM	dynamic p
RE	R_e	063	SR24990 or READIT	radius of
TBMAT, 3	T_{X_B}, T_{Y_b}, T_{Z_B}	3643-3645	PRELIM	thrusting

ce of ut	Description	Units
ATE, DEREQ VIXEN	instantaneous angle of attack	radians
LIM	instantaneous reference area - area of base	ft^2
490 or DIT	gravitational acceleration - preset 32.21852	ft/sec^2
EQ or GCØ	total drag coefficient based on A_{ref}	-
ATE	normal force coefficient	-
TMPY	components of body force in trajectory coordinate system	lb.
EQ	instantaneous flight path angle	radians
DIT	input specific impulse of thrust	sec.
LIM	instantaneous mass of vehicle	slug.
EQ	Euler angle, Φ	rad.
EQ	Euler angle, Ψ	rad.
EQ	thrust misalignment angle	rad.
LIM	dynamic pressure	lb/ft^2
4990 or DIT	radius of the earth	ft.
LIM	thrusting force components in body coordinate system	lb.

B

2. Input (Concl'd)

Name	Symbol	Occur/Noccur Number	Source of Input	D
THEALP	θ_a	071	DEREQ	Euler ang
THINF	Th	213	PRELIM	thrusting
TTMAT, 3	T_{X_T}, T_{Y_T}, T_{Z_T}	3565-3567	MATMPY	thrusting
V	V	082	DEREQ	velocity
Z	Z	091	DEREQ	altitude

3. Output

CMAT, 9	C_{xx}, C_{xy}, C_{xz}, C_{yx}, C_{yy}, C_{yz}, C_{zx}, C_{zx}, C_{zz}	-		matrix fo body coor coordinate
DERIV, 16		-		derivative by ADM4R
FBMAT, 3	F_{x_B}, F_{y_B}, F_{z_B}	-		componen system
TBMAT, 3	T_{x_B}, T_{y_B}, T_{z_B}	3643-3645		componen system

Description	Units
Euler angle, Ⓗ	radians
thrusting force in vacuum	lb.
thrusting force components in trajectory coordinate system	lb.
velocity	ft/sec
altitude	ft.
matrix for converting components in body coordinate system to trajectory coordinate system	-
derivatives of the 16 quantities being integrated by ADM4RK, see VIXEN	-
components of body force in body coordinate system	lb.
components of thrusting force in body coordinate system	lb

4. Numerical Procedure

The subroutine may be divided into three main sections: (1) calculation of components of thrusting force in trajectory coordinate system, (2) calculation of body force components in trajectory coordinate system, (3) calculation of the derivatives.

The calculation of the components of the thrusting force begins with the evaluation of the sines and cosines of the Euler angles - θ_a, Ψ, and ϕ. The nine components of the 3 x 3 C matrix for converting components in a body coordinate system to those in a trajectory coordinate system are then defined as follows:

C_{xx} = CMAT(1,1) = $\cos \theta_a \cos \Psi$

C_{xy} = CMAT(2, 1) = $\cos \theta_a \sin \Psi$

C_{xz} = CMAT(3, 1) = $- \sin \theta_a$

C_{yx} = CMAT(1, 2) = $\cos \Psi \sin \theta_a \sin \phi - \sin \Psi \cos \phi$

C_{yy} = CMAT(2, 2) = $\sin \Psi \sin \theta_a \sin \phi + \cos \Psi \cos \phi$

C_{yz} = CMAT(3, 2) = $\cos \theta_a \sin \phi$

C_{zx} = CMAT(1, 3) = $\cos \Psi \sin \theta_a \cos \phi + \sin \Psi \sin \phi$

C_{zy} = CMAT(2, 3) = $\sin \Psi \sin \theta_a \cos \phi - \cos \Psi \sin \phi$

C_{zz} = CMAT(3, 3) = $\cos \theta_a \cos \phi$

SUBROUTINE MATMPY is called in to multiply the conversion matrix, CMAT, by the components of the thrust vector in the body coordinate system, TBMAT, to obtain the components of the thrust vector in the trajectory coordinate system, TTMAT.

The components of the velocity vector in the body coordinate system are then determined using the relations:

$$\dot{X}_B = C_{xx} \; V$$

$$\dot{Y}_B = C_{yx} \; V$$

$$\dot{Z}_B = C_{zx} \; V$$

These values are used in conjunction with the calculated values for the normal force and for the axial force coefficient,

$$F_N = C_N \; q_D \; A_{ref}$$

$$C_x = (-C_D + C_N \sin \alpha') / \cos \alpha'$$

to calculate the components of the body forces in the body coordinate system as indicated below:

$$F_{x_B} = FBMAT(1) = q_D \; A_{ref} \; C_x$$

$$R\emptyset\emptyset T = \sqrt{\dot{Y}_B^2 + \dot{Z}_B^2}$$

If $R\emptyset\emptyset T < 1.0 \times 10^{-10}$, set $F_{y_B} = 0.0$ and $F_{z_B} = 0.0$; otherwise,

$$F_{y_B} = \text{FBMAT}(2) = -F_N\left(\dot{Y}_B \Big/ \sqrt{\dot{Y}_B^2 + \dot{Z}_B^2}\right)$$

$$F_{z_B} = \text{FBMAT}(3) = -F_N\left(\dot{Z}_B \Big/ \sqrt{\dot{Y}_B^2 + \dot{Z}_B^2}\right) .$$

SUBROUTINE MATMPY is again employed using FBMAT and the conversion matrix, CMAT, as input to obtain the components of the body forces in the trajectory coordinate system, FTMAT.

The body forces and thrusting forces are then used in the determination of the trajectory-related derivatives using the following relations:

$$\text{DERIV}(1) = \dot{V} = -\frac{q_D C_D A_{ref}}{m} - G\left(\frac{R_e}{R_e + Z}\right)^2 \sin\gamma_F + \frac{T_{x_T}}{m}$$

$$\text{DERIV}(2) = \dot{\gamma}_f = \frac{\cos\gamma_f}{V}\left(\frac{V^2}{R_e + Z} - \frac{G R_e^2}{(R_e + Z)^2}\right) - \left(\frac{F_{z_T} + T_{z_T}}{mV}\right)$$

$$\text{DERIV}(3) = \frac{dt}{dz} = 1.0 / V \sin\gamma_f$$

$$\text{DERIV}(4) = \dot{X}_r = \cos\psi_a \; V \cos\gamma_f \; \frac{R_e}{R_e + Z}$$

$$\text{DERIV}(14) = \dot{Y}_r = \sin\psi_a \; V \cos\gamma_f \; \frac{R_e}{R_e + Z}$$

$$\text{DERIV}(15) = \dot{\psi}_a = -(F_{y_T} + T_{y_T}) / (mV \cos\gamma_f)$$

$$\text{DERIV}(16) = \dot{W}_{TH} = -\frac{Th}{I_{sp}}$$

These values are then transferred to the calling subroutine as arguments of SUBRØUTINE TEQUAT.

5. Other Information

A. SUBROUTINE TEQUAT is called by SUBROUTINE DEREQ.

B. SUBROUTINE TEQUAT calls in SUBROUTINE MATMPY.

C. The following library functions are utilized by SUBROUTINE TEQUAT:

1. DSIN
2. DCØS
3. DSQRT

SUBROUTINE MATMPY(A, B, C)

1. Purpose

SUBROUTINE MATMPY performs the multiplication of a 3 by 3 matrix B by a 3 component column matrix C to obtain the 3 component matrix A.

2. Input

Name	Symbol	Description
B, 9	B_{11}, B_{12}, B_{13}, B_{21}, B_{22}, B_{23}, B_{31}, B_{32}, B_{33}	square 3 x 3 matrix
C,	C_1, C_2, C_3	3 component column matrix

3. Output

Name	Symbol	Description
A, 3	A_1, A_2, A_3	matrix resulting from multiplication of B x C

4. Numerical Procedure

SUBROUTINE MATMPY utilizes two nested DØ loops to perform the matrix multiplication which is the evaluation of

$$A_i = \sum_{j=1}^{3} B_{ij} \times C_j \qquad \text{for } i = 1, 2, 3$$

The inner DØ loop performs the summation in j, then the outer DØ loop changes the integer i to obtain the value for A_i for each i to replace the initial value $A_j = 0.0$.

5. Other Information

A. SUBROUTINE MATMPY is called in by SUBROUTINE TEQUAT.

B. SUBROUTINE MATMPY calls in no other subroutines or internal functions.

3.2 Wake Calculations

SUBROUTINE WAKE performs preliminary calculations utilized by subroutine FLOWF and RCSEC. SUBROUTINE FLOWF performs flow field calculations and uses SUBROUTINE AR2DIM and FUNCTION AR3DIM to interpolate in 2 and 3 dimensional tables respectively. SUBROUTINE RCSEC determines the peak wake radar cross section and utilizes SUBROUTINE FIB1, a one variable Fibonacci search which optimizes FUNCTION FUN1, in determining the wake length.

3.2.1 Preliminary Calculations

SUBROUTINE WAKE

1. Purpose

SUBROUTINE WAKE calculates some quantities needed by SUBROUTINE FLOWF and RCSEC, then calls these subroutines to calculate the vehicle wake length and cross section at each of three radar frequencies.

2. Input

Unless otherwise specified, all numbers in COMMON LOCATION refer to positions in the ØCCUR array. See part 5 for tables used.

Name	Common Location	Source of Input	Description
A, 514	301-814	ZPRS	preset constants
AREF	001	PREL.M	reference area, ft^2
BZ	FRLTNK	INTERP	scale height (1000 ft.)
B1		FLØWF	exponential decoy constant (1/meters)
CD	016	DRAGCØ or DEREQ	total drag coefficient
CDB	099	DRAGCØ	base drag coefficient
CDI	100	DRAGCØ	total induced drag coefficient
CDP	098	DRAGCØ	pressure drag coefficient
CPE	017	PRELIM	specific heat at constant pressure at the edge of boundary layer of sharp cone, Btu/lbm - ^{o}R
D	021	PRELIM	base diameter in feet

2. Input (Cont'd)

Name	Common Location	Source of Input	Description
ENET		FLØWF	transition point electron density, e/cc
GAMF	026	VIXEN or DEREQ	flight path angle in radians
IDBL	IØCCUR(314)	ZREADX	code which determines units of radar cross section, see ZREADX
IØP, 90	IØCCUR(1-90)	ZREADX	option parameter
LPLØT	IØCCUR(302)	VIXEN	index of altitude at which VIXEN has just produced output
ME	036	PRELIM	Mach number at the edge of the boundary layer
MINF	035	PRELIM or VIXEN	Mach number in the free stream
MØPT	NØCCUR(03)	READIT	option parameter
MUINF	034	PRELIM	free stream viscosity, lbm/ft-sec
PE	048	PRELIM	pressure at edge of sharp cone boundary layer, lb/ft^2
PINF	049	PRELIM	free stream pressure, lb/ft^2
RHØE	061	PRELIM	density at the edge of the boundary layer, lbm/ft^3
RHØINI	056	PRELIM	free stream density in lbm/ft^3
RNNX	052	DEREQ or CHNTBL	nose radius, same as Rn, ft.
TE	079	PRELIM	temperature at edge of sharp cone boundary layer, ^oR

2. Input (Concl'd)

Name	Common Location	Source of Input	Description
THETA	076	CHNTBL	cone half angle, radians
TINF	072	PRELIM	free stream temperature, ^{o}R
V	082	DEREQ or VIXEN	velocity, ft/sec
VE	083	PRELIM	velocity at edge of sharp cone boundary layer, ft/sec
WDØT	086	TØMALØ	weight loss due to ablation, lb/sec
XBAR	090	PRELIM	interaction parameter, $\bar{\chi}$
XLØW	238	MAIN or READIT	the value of $\bar{\chi}$ which is the upper limit of fully laminar flow regime
ZNX	091	VIXEN or DEREQ	altitude, same as Z, ft.
ZPLØT, 160	PCCUR (161-320)	VIXEN	output altitudes, ft.
ZTR	092	PRELIM	transition altitude, ft.

3. Output

Name	Common Block	Description
AMABLD	FRLTNK	mass ablated (lbm/sec)
AMBLSD	FRLTNK	mass swallowed by boundary layer (lbm/sec)
AMINFC	FRLTNK	cone Mach number
AMU	FRLTNK	upstream Mach number

3. Output (Cont'd)

Name	Common Block	Description
AMUU	FRLTNK	free stream viscosity, $\frac{\text{lbm}}{\text{ft.-sec.}}$
BZ	FRLTNK	See Input
B1		See Input
CDTA	FRLTNK	total drag times area, ft^2
CDVA	FRLTNK	viscous drag times area, ft^2
C1	CRCSEC	preset constant
C2	CRCSEC	preset constant
C3	CRCSEC	preset constant
C4	CRCSEC	preset constant
C5	CRCSEC	preset constant
DB	FRLTNK	base diameter (ft)
DDW	CRCSEC	wake diameter (meters)
ENET	-	See Input
F	CRCSEC	frequency (cycles/sec)
GAMMAE	FRLTNK	entry angle (rad)
HINFC	FRLTNK	cone enthalpy, ft^2/sec^2
HH	CRCSEC	altitude (1000 ft.)
HU	FRLTNK	upstream static enthalpy, ft^2/sec^2
PHI	CRCSEC	look angle (degrees)

3. Output (Cont'd)

Name	Common Block	Description
PICIPU	FRLTNK	ratio of cone to upstream pressure
PU	FRLTNK	upstream pressure (lb/ft^2)
RHØIC	FRLTNK	cone density (lbm/ft^3)
RHØU	FRLTNK	upstream density (lbm/ft^3)
RN	FRLTNK	nose radius (ft)
SC	FRLTNK	length of conical frustum (ft)
SIGMDS	CRCSEC	noise level to which wake length is measured (square meters)
TAU	CRCSEC	pulse length (μ sec)
THETAC	FRLTNK	cone angle (rad)
UNIFC	FRLTNK	cone velocity (1000 ft/sec)
UU	FRLTNK	upstream velocity (1000 ft/sec)
WLIP, 160	PCCUR(1281)	wake length at radar frequency 1, meters
WL2P, 160	PCCUR(1441)	wake length at radar frequency 2, meters
WL3P, 160	PCCUR(1601)	wake length at radar frequency 3, meters
WR1P, 160	PCCUR(801)	cross section at radar frequency 1, see IDBL
WR2P, 160	PCCUR(961)	cross section at radar frequency 2, see IDBL
WR3P, 160	PCCUR(1121)	cross section at radar frequency 3, see IDBL
XBZ	CRCSEC	scale height
Z	FRLTNK	altitude (1000 ft)

3. Output (Concl'd)

Name	Common Location	Description
ZBLT	FRLTNK	boundary layer transition altitude (1000 ft)
ZME	CRCSEC	Mach number (upstream)

4. Numerical Procedure

WAKE first tests IWPRNT, If IWPRNT equals zero, the WRITE statement following the test is omitted and statement 10 is executed. If IWPRNT equals one, the WRITE statement is executed and statement 10 follows. The test at 10 determines whether the conditions are satisfied to permit the wake length and cross section calculations at the given altitude. If they are, control passes to 30. If they are not, the three wake lengths are set equal to zero and control is returned to VIXEN.

At 30, the value of BZ is found by linear interpolation in the tables of BETAZ vs. WKALT at the altitude ZPLØT.

In the sequence of statements beginning at 30 and ending at 102, quantities required by RCSEC and FLØWF are calculated. After 102 FLØWF is called to get B1 and ENET which are needed by RCSEC. The next two statements produce printed output, if IWPRNT equals one. Then, if IØP(67) plus IØP(76) equal zero go to 40. Otherwise, the next four statements calculate quantities needed by RCSEC for calculating the vehicle wake length and cross section at radar frequency one. Next, RCSEC is called to perform the calculations, if WL1P (LPLØT) is not equal to zero, control passes to 35. If it equals zero, set WR1P (LPLØT) equal

to zero, then change it to one if IDBL equals four. This is done because it may be necessary to plot the log of WR1P.

At 35, IDBL equals three, WR1P is converted to decibels.

The sequence of statements starting at 40 and ending just before 50 provides for the calculation of vehicle wake length and cross section at peak frequency three.

At 50, control is returned to VIXEN.

5. Other Information

A. SUBROUTINE WAKE is called by SUBROUTINE VIXEN only.

B. SUBROUTINE WAKE calls

1. SUBROUTINE FLØWF
2. SUBROUTINE RCSEC

C. SUBROUTINE WAKE calls in the IBM supplied routine DLØG10.

D. Tables of 10 values each used in this subroutine which are transmitted from SUBROUTINE ZREADX through labeled common CWAKE are:

BETAZ, scale height (1000 ft)
PHI1, look angle for first radar frequency in degrees
PHI2, look angle for second radar frequency in degrees
PHI3, look angle for third radar frequency in degrees

which are a function of altitude, WKALT.

3.2.2 Flow Field Calculations

SUBROUTINE FLØWF (ENET, B1, IND)

1. Purpose

SUBROUTINE FLØWF calculates the electron density transition, n_{et} (e/cc) and the decay rate, b_1.

Name	Symbol	Source of Input	
AKW	K_w	ZREADX	heatshield con
AMABLD	$\dot{m}^*_{ABL}$	WAKE	mass ablated
AMBLSD	$\dot{m}^*_{B.L.S.}$	WAKE	mass swallow
AMINFC	$M_{\infty c}$	WAKE	cone Mach. nu
AMU	M_u	WAKE	upstream Mac
AMUU	μ_u	WAKE	free stream vi
BZ	β_z	WAKE	scale height
B21	b_{21}	ZREADX	scaling consta
B22	b_{22}	ZREADX	scaling consta
B23	b_{23}	ZREADX	scaling consta
CDTA	$C_{DT}A$	WAKE	total drag tim
CDVA	$C_{DV}A$	WAKE	viscous drag
CRHØW	$C_{\rho W}$	ZREADX	heatshield spe
C(59)	C_{59}	ZREADX	preset constan
C(60)	C_{60}	ZREADX	preset constan
C(67)	C_{67}	ZREADX	preset constan
C(69)	C_{69}	ZREADX	preset conetan

Description	Units
heatshield conductivity	Btu/ft - ^{o}R-Hr
mass ablated	lbm/sec
mass swallowed by boundary layer	lbm/sec
cone Mach. number	-
upstream Mach. number	-
free stream viscosity	lbm/(ft-sec)
scale height	1000 ft.
scaling constant	
scaling constant	
scaling constant	
total drag times area	ft^2
viscous drag times area	ft^2
heatshield specific heat	Btu/lb - ^{o}R
preset constants	
preset constants	
preset constants	
preset constants	

B

2. Input (Cont'd)

Name	Symbol	Source of Input	
C(83) - C(93)	$C_{83} - C_{93}$	ZREADX	preset cor
C(100)	C_{100}	ZREADX	preset cor
C(115) - C(125)	$C_{115} - C_{125}$	ZREADX	preset cor
C(130) - C(136)	$C_{130} - C_{136}$	ZREADX	preset cor
C(159)	C_{159}	ZREADX	preset cor
C(160)	C_{160}	ZREADX	preset cor
C(164)	C_{164}	ZREADX	preset cor
C(165)	C_{165}	ZREADX	preset cor
C(169)	C_{169}	ZREADX	preset cor
DB	D_B	WAKE	base diam
DELWH	δ_{WH}	ZREADX	heatshield
DHCHEM	Δh_{chem}	ZREADX	chemical e
GAMMAE	γ_e	WAKE	entry angl
HINFC	h_{∞_c}	WAKE	cone entha
HU	h_u	WAKE	upstream
IND		WAKE	IND = 1: A IND ≠1: N
PICIPU	$\frac{P_{\infty_c}}{P_u}$	WAKE	Ratio of co
PU	P_u	WAKE	upstream p
RHØ	ρ_u	WAKE	upstream d

ɔut	Description	Units
	preset constants	
	preset constants	
	preset constants	
	preset constants	
	preset constants	
	preset constants	
	preset constants	
	preset constants	
	preset constants	
	base diameter	ft.
	heatshield thickness	in.
	chemical enthalpy of heatshield	ft^2/sec^2
	entry angle	deg.
	cone enthalpy	ft^2/sec^2
	upstream static enthalpy	ft^2/sec^2
	IND = 1: All calculated quantities are printed; IND ≠1: No print out	
	Ratio of cone to upstream pressure	
	upstream pressure	lb/ft^2
	upstream density (identical to RHØU)	lbm/ft^3

B

2. Input (Cont'd)

Name	Symbol	Source of Input	D
RHØIC	$\rho_{\infty c}$	WAKE	cone densit
RHØSL	ρ_{SL}	ZREADX	sea level d
RHØU	ρ_u	WAKE	upstream d
RHØW	ρ_w	ZREADX	heatshield
RN	R_N	WAKE	nose radius
RTØ	RT_o	ZREADX	reference e
SC	S_c	WAKE	length of c
TABL	T_{ABL}	ZREADX	ablation of
THETAC	θ_c	WAKE	cone angle
UINFC	$U_{\infty c}$	WAKE	cone veloci
UU	U_u	WAKE	upstream v
V	U_u	WAKE	upstream v
Z	Z	WAKE	altitude
ZBLT	Z_{BLT}	WAKE	boundary la
***** TABULAR INPUTS **********			**********
D	n_e	ZREADX	electron de enthalpy an seed (Table
EMCTBL	M_c	ZREADX	argument i
ENTABL	n_c	ZREADX	equilibrium
ERNRTB	ρ	ZREADX	density arg
ERNTBL	n_e	ZREADX	equilibrium table (Tabl

rce of Input	Description	Units
WAKE	cone density	lbm/ft^3
ZREADX	sea level density	lbm/ft^3
WAKE	upstream density	lbm/ft^3
ZREADX	heatshield density	lbm/ft^3
WAKE	nose radius	ft.
ZREADX	reference enthalpy	ft^2/sec^2
WAKE	length of conical frustum (along cone)	ft.
ZREADX	ablation of temperature of heatshield	^oK
WAKE	cone angle	deg.
WAKE	cone velocity	1000ft/sec
WAKE	upstream velocity	1000 ft/sec
WAKE	upstream velocity (identical to UU)	1000 ft/sec
WAKE	altitude	1000 ft.
WAKE	boundary layer transition altitude	1000 ft.
*********	**********	
ZREADX	electron density as a function of normalized enthalpy and air density for 1000 PPM sodium seed (Table D)	electrons/cc
ZREADX	argument in table E , Mach number on cone	
ZREADX	equilibrium electron density table (Table B)	electrons/cc
ZREADX	density argument in Table F	lbm/ft^3
ZREADX	equilibrium normal shock electron density table (Table F)	electrons/cc

B

2. Input (Concl'd)

Name	Symbol	Source of Input	D
ERNUTB	V or U	ZREADX	velocity argum
ETABL	M	ZREADX	Table of M vs.
HSTABL	h/RT_o	ZREADX	non-dimensiona
RSTABL	ρ/ρ_o	ZREADX	non-dimensiona
THTTBL	θ_c	ZREADX	cone angle, argu
XDTABL	ρ/ρ_o (ATM)	ZREADX	air density, argu
YDTABL	h/RT_o	ZREADX	normalized enth
ZDTABL	M_{RAT}	ZREADX	argument in Ta rate to the ma

nput	Description	Units
)X	velocity argument in Table F	kfps
)X	Table of M vs. M_c and θ_c (Table E)	
) X	non-dimensional enthalpy, argument in Table B	
)X	non-dimensional density, argument in Table B	
)X	cone angle, argument in Table E (cone half angle)	deg.
)X	air density, argument in Table D	
)X	normalized enthalpy, argument in Table D	
)X	argument in Table D, the ratio of ablated mass loss rate to the mass loss rate in the boundary layer	

B

3. Output

Name	Symbol	Description	Units
AI	A_I	additive term for second entropy layer shock angle	
AKE	K_E	$G_{t_{chem}}$ factor	
AKV	K_V	heating constant	
AMBLDS	$\dot{m}^*_{BL}$	mass flow boundary layer	lbm/sec
AMNDS	$\dot{m}^*_N$	mass flow nose	lbm/sec
AMRAT	M_{rat}	ratio ablation to boundary layer air mass flow	
AMS	M_s	Mach number shoulder	
AMSS2D	$\dot{m}^*_{\Sigma S2}$	mass flow 2nd entropy layer	lbm/sec
AM2	M_2	mass number 2nd entropy layer	lbm/sec
AM2C	M_{2C}	Mach number cone - 2nd entropy layer	
B1	b_1	decay rate	
B5	b_5	net scaling factor	
B11	b_{11}	electron decay rate factor	
B35	b_{35}	N_{SRN} factor	
CDSS2A	$C_{D\Sigma S2}A$	drag times area for 2nd entropy layer	ft^2
ENBL	N_{BL}	number of electrons leaving boundary layer	e/sec
ENEBL	n_{eBL}	electron density in the boundary layer	e/cc
ENEN	n_{en}	electron density at the neck	e/cc
ENEQH	n_{eQH}	equilibrium electron density	e/cc

3. Output (Cont'd)

Name	Symbol	Description	Units
ENERN	n_{eRN}	electron density at the nose cap	e/cc
ENET	n_{et}	transition electron density	e/cc
ENSRN	N_{SRN}	number of electrons e^- at shoulder	e/cc
ENRN	N_{RN}	number of electrons produced by nose cap	e/cc
ENS	N_s	number of electrons entering wake neck	e/cc
GT	G_t	function aero	
GTCHEM	G_{tchem}	function chemical	
HC	h_c	enthalpy cone	ft^2/sec^2
HN	h_n	enthalpy neck	ft^2/sec^2
HNIHS	h_n/H_s	ratio neck enthalpy to total shoulder enthalpy, $H_s=h_s + U^2/2$	
HS	h_s	enthalpy shoulder	ft^2/sec^2
HW	h_w	enthalpy wall	ft^2/sec^2
H2	h_2	enthalpy 2nd entropy layer	ft^2/sec^2
H2C	h_{2C}	enthalpy cone - 2nd entropy layer	ft^2/sec^2
PC	P_c	pressure cone	lb/ft^2
P2C	P_{2c}	pressure cone - 2nd entropy layer	lb/ft^2

3. Output (Cont'd)

Name	Symbol	Description	Units
RES	R_{es}	Reynolds number shoulder	
REUD	R_{eud}	Reynolds number upstream	
RETS	$R_{e\theta s}$	shoulder Reynolds number based on theta	
RHØC	ρ_c	density cone	lbm/ft^3
RHØS	ρ_s	density shoulder	lbm/ft^3
RHØ2	ρ_2	density 2nd entropy layer	lbm/ft^3
RHØ2C	ρ_{2c}	density cone - 2nd entropy layer	lbm/ft^3
SCHEM	S_{chem}	chemical length	ft.
SRAT	S_{rat}	ratio boundary swallowing to nose mass flow	
THTBLS	θ_{BLS}	initial wake momentum thickness based on shoulder conditions	ft.
THETSI	$\theta_{s\infty}$	shock angle universal	deg.
THETSS	θ^*_s	shoulder wake momentum thickness	ft.
THETS2	θ_{S2}	shock angle max.	deg.
THTS2B	$\overline{\theta_{S2}}$	average shock angle for second entropy layer	rad.
TW	T_W	temperature wall	^oK
TWC	T_{WC}	temperature wall cone	^oK
TWSP	T_{WSP}	temperature wall sphere	^oK

3. Output (concl'd)

Name	Symbol	Description	Units
UC	U_c	velocity cone	1000ft/sec
US	U_s	v elocity shoulder	1000ft/sec
U2	U_2	velocity of second entropy layer	1000 ft/sec
U2C	U_{2c}	velocity cone - 2nd entropy layer	1000 ft/sec

4. Numerical Procedure

The Fortran listing of SUBROUTINE FLØWF is numbered in the following manner: At the end of each line is written FLØWF i where i refers to the number of each line.

The actual calculations of this subroutine begin on line 185. The following is a line by line account of the equations calculated in the subroutine.

Lines 185, 186

$$R_{eud} = \frac{10^3 \rho_u U_u D_B}{\mu_u}$$

$$K_v = \frac{K_w U_u \sin \gamma_e}{\rho_w C_{\rho w} \delta^2_{wH}}$$

Lines 187 - 197

$$T_{wc} = \begin{cases} 278 + \dfrac{C_{60}(K_v Z)^{C_{59}}}{\left(\dfrac{S_c e^{Z/\beta_z}}{\sin^2 \theta_c}\right)^{1/2}} \left(\dfrac{K_w}{\delta_{WH} U_u^3}\right) & \text{if } Z \geq Z_{BLT} \\ 278 + \dfrac{C_{60}(K_v Z_{BLT})^{C_{59}}}{\left(\dfrac{S_c e^{Z_{BLT}/\beta_z}}{\sin^2 \theta_c}\right)^{1/2}} \left(\dfrac{K_w}{\delta_{WH} U_u^3}\right) + \dfrac{C_{160}\left[K_v (Z_{BLT} - Z)\right]^{C_{159}}}{\left(\dfrac{S_c e^{Z/\beta_z}}{\sin^2 \theta_c}\right)^{.8}} \left(\dfrac{K_w}{\delta_{WH} U_u^3}\right) & \text{if } Z < Z_{BLT} \end{cases}$$

Line 198 - 199

$$T_{WSP} = 278 + \frac{C_{60}(K_v Z)^{C_{59}}}{\frac{K_w D_B^{1/2} e^{(Z/4\beta_z)}}{U_u^{3/2} \delta_{WH}}}$$

Line 200

$$\dot{m}^*_{BL} = \dot{m}^*_{BLS} + \dot{m}^*_{ABL}$$

Lines 201 - 204

$$T_W = \begin{cases} T_{ABL}; & (T_{ABL} \leq T_{WC} \text{ and } 2R_N < .99 D_B) \text{ or } (T_{ABL} \leq T_{WSP} \text{ and } 2R_N \geq .99 D_B) \\ T_{WC}; & 2R_N < .99 D_B \text{ and } T_{WC} < T_{ABL} \\ T_{WSP}; & 2R_N \geq .99 D_B \text{ and } T_{WSP} < T_{ABL} \end{cases}$$

Lines 205, 206

$$h_W = 1.087(10^4)\ T_W$$

$$C_{D\Sigma\ S2} A = C_{DT} A - C_{DV} A$$

Lines 207 - 211

The subroutine prints out the quantities: R_{eud}, K_v, T_{WC}, T_{WSP}, $\dot{m}^*_{BL}$, T_W, h_W, $C_{D\Sigma S2}A$ if the subroutine argument IND equals 1.

Line 212

$$M_{RAT} = \frac{\dot{m}^*_{ABL}}{\dot{m}^*_{BL}}$$

Lines 213-217

$$\theta_{s\infty} = \begin{cases} \sin^{-1}\left(\frac{1}{M_u}\right) & \text{if } \frac{P_{\infty c}}{P_u} \leq \frac{19}{6} \\ \sin^{-1}\left[\frac{1}{M_u}\left[\left(\frac{6}{7}\frac{P_{\infty c}}{P_u} - \frac{13}{7}\right)\right]^{1/2}\right] & \text{if } \frac{P_{\infty c}}{P_u} > \frac{19}{6} \end{cases}$$

Lines 218 - 226 evaluate the following group of equations:

$$\theta_{s\infty} = \text{Max}\,(\theta_{s\infty}, \theta_c)$$

$$\dot{m}^*_{\Sigma S2} = 2000 \rho_u U_u R_N^2 \cot^2 \theta_{s\infty}$$

$$U_{2c} = U_u - \frac{500 \rho_u U_u^2 C_{D\Sigma S2}A}{\dot{m}^*_{\Sigma s2}}$$

$$U_{2c} = \text{Max}\,(U_{2c}, 1)$$

$$A_I = .286 + 1.029\ M_u^2 \left[1 - \left(\frac{U_{2c}}{U_u}\right)^2\right]$$

$$\theta_{s2} = \sin^{-1}\left(\frac{1}{M_u}\left[\frac{A_I + (2.86 + A_I)^{1/2}}{2}\right]^{1/2}\right)$$

$$\theta_{s2} = \text{Max}\,(\theta_{s2}, \theta_c)$$

$$\bar{\theta}_{s2} = .5\,(\theta_{s2} + \theta_{s\infty})$$

$$b_{35} = C_{119}\ U_u^{\ C_{120}}$$

Lines 227-231

The subroutine prints out the quantities: M_{RAT}, $\theta_{s\infty}$, $\dot{m}^*_{\Sigma s2}$, U_{2c}, A_I, θ_{s2}, $\bar{\theta}_{s2}$, b_{35} if IND = 1.

Line 232

$$\dot{m}^*_N = \pi R_N^2\ C_{116}\ \rho_u\ U_u$$

Lines 233 - 236

$$S_{RAT} = \begin{cases} 1 & \text{if } \dot{m}^*_{BLS} > \dot{m}^*_N \\ \dfrac{\dot{m}^*_{BLS}}{\dot{m}^*_N} & \text{if } \dot{m}^*_{BLS} < \dot{m}^*_N \end{cases}$$

Line 237

Using ρ and V in SUBROUTINE AR2DIM, this line interpolates linearly in the Equilibrium Normal Shock Electron Density Table to obtain n_{eRN}

Lines 238 - 241 determine the following quantities:

$$N_{RN} = \frac{(30.48)^3 \dot{m}^*_N S_{RAT} n_{eRN}}{\rho_u}$$

$$P_{2C} = \frac{(7 M_u \sin^2 \tilde{\theta}_{s2} - 1) P_u}{6}$$

$$h_{2c} = \frac{h_u P_{2C}(P_{2C} + 6P_u)}{P_u(6P_{2C} + P_u)}$$

$$\rho_{2c} = \frac{\rho_u(6 P_{2C} + P_u)}{P_{2C} + 6 P_u}$$

Lines 242-254

If $\dot{m}^*_{BLS} \geq \dot{m}^*_{\Sigma S2}$, the following quantities are defined.

$$\rho_c = \rho_{\infty c}$$

$$h_c = h_{\infty c}$$

$$P_c = P_{\infty c}$$

$$U_c = U_{\infty c}$$

$$S_{chem} = \frac{\rho_{2C}^{C_{117}} S_c \dot{m}^*_{\Sigma S2}}{U_{2C} \dot{m}^*_{BLS}} + \frac{\rho_{\infty C}^{C_{117}} S_c(\dot{m}^*_{BLS} - \dot{m}^*_{\Sigma S2})}{U_{\infty C} \dot{m}^*_{BLS}}$$

If $\dot{m}^*_{\Sigma S2} > \dot{m}^*_{BLS}$, the following quantities are defined.

$$\rho_c = \rho_{2C}$$

$$h_c = h_{2C}$$

$$P_c = P_{2C}$$

$$U_c = U_{2C}$$

$$S_{chem} = \frac{\rho_{2C}^{C_{117}} \; S_c \; \dot{m}^*_{BLS}}{U_{2C} \; \dot{m}^*_{\Sigma S2}}$$

Lines 255 - 257

$$N_{SRN} = \frac{C_{122} \, N_{RN} \, \dot{m}^*_N \left[1 - C_{121} + (1 + N_{RN} \, S_{chem}) \, C_{121} \, e^{\left(-b_{35} - S_{chem}\right)}\right]^{C_{124}}}{\dot{m}^*_N + C_{123} \, S_{chem} \, N_{RN} \, T_W}$$

Lines 258-262

The subroutine prints out the quantities:

$\dot{m}^*_N$, S_{RAT}, n_{eRN}, N_{RN}, P_{2C}, h_{2C}, ρ_{2C}, ρ_c if IND = 1

Lines 263 - 269

n_{eQH} depends on $\frac{\rho_c}{\rho_{SL}}$, $\frac{h_c}{RT_o}$, M_{RAT} and is interpolated from one of the following tables:

TABLE B if $M_{RAT} = 0$

TABLE D_1 if $M_{RAT} = 0.01$

TABLE D_2 if $M_{RAT} = 0.1$

TABLE D_3 if $M_{RAT} = 1.0$

by using FUNCTION AR3DIM and SUBROUTINE AR2DIM.

Lines 270-271

$$n_{eBL} = n_{eQH}\left[1 - e^{\left(-b_{22}\, S_{chem}^{C_{118}}\right)}\right]\left[b_{21} + b_{23}\left(\frac{U_c - 22}{22}\right)\right]$$

Line 272

$$N_{BL} = \frac{(30.48)^3\; n_{eBL}\; \dot{m}^*_{BL}}{P_C}$$

Lines 273 - 277

The subroutine prints out the quantities:

h_c, P_c, U_C, S_{chem}, N_{SRN}, N_{eQH}, $n_{e\,BL}$, N_{BL} if IND = 1

Lines 278 - 280

$$N_s = N_{BL} + N_{SRN}$$

$$M_{2C} = \frac{U_{2C}\; M_u}{U_u}\left(\frac{h_u}{h_{2C}}\right)^{1/2}$$

$$M_{2C} = \text{Max}\,(M_{2C}, 1)$$

Lines 281-282

M_2 depends on M_{2C}, θ_c and is obtained by using the linear interpolation SUBROUTINE AR2DIM in Table E.

Line 283

M_{∞} depends on $M_{\infty c}$, θ_c and is obtained by using the linear interpolation SUBROUTINE AR2DIM on Table E.

Lines 284-291 compute the values:

$$\rho_{\infty} = \rho_{\infty c} \left[\frac{1 + .2M^2_{\infty c}}{1 + .2 M^2_{\infty}} \right]^{2.5}$$

$$h_{\infty} = h_{\infty c} \left[\frac{1 + .2 M^2_{\infty c}}{1 + .2 M^2_{\infty}} \right]$$

$$U_{\infty} = \frac{U_{\infty c} M_{\infty}}{M_{\infty c}} \left[\frac{h_{\infty}}{h_{\infty c}} \right]^{1/2}$$

$$\rho_2 = \rho_{2C} \left[\frac{1 + .2 M^2_{2C}}{1 + .2 M^2_2} \right]^{2.5}$$

$$h_2 = h_{2C} \left(\frac{1 + .2 M^2_{2C}}{1 + .2 M^2_2} \right)$$

$$U_2 = \frac{U_{2C} M_2}{M_{2C}} \left[\frac{h_2}{h_{2C}} \right]^{1/2}$$

Lines 292-301

$$\left.\begin{aligned} M_s &= M_\infty \\ \rho_s &= \rho_\infty \\ h_s &= h_\infty \\ U_s &= U_\infty \end{aligned}\right\} \quad \text{if } \dot{m}^*_{BLS} \geq \dot{m}^*_{\Sigma S2}$$

$$\left.\begin{aligned} M_s &= M_2 \\ \rho_s &= \rho_2 \\ h_s &= h_2 \\ U_s &= U_2 \end{aligned}\right\} \quad \text{if } \dot{m}^*_{BLS} < \dot{m}^*_{\Sigma S2}$$

Line 302

$$R_{es} = \frac{\rho_s \; M_s \; R_{eud} \; C_{169}}{\rho_u \; M_u} \left[\frac{h_s}{h_u}\right]^{C_{67}}$$

Lines 303-307

The subroutine prints out the quantities:

N_s, M_{2C}, M_2, ρ_2, h_2, U_2, M_s, ρ_s if IND = 1

Lines 308-312

$$C_s = C_{164} + C_{165} \sin^{1/4}(\theta_c) \, R_{es}^{-3/8} \, M_s^{-.5}$$

$$h_n = h_w + C_s \left[h_s - h_w + .5\,(10^6)\, U_s^2 \right] \quad - \Delta h_{chem}$$

$$h_n / H_s = \frac{2\, h_n}{2h_s + 10^6\, U_s^2}$$

$$n_{eN} = \frac{\rho_s\, N_s\, h_s}{(30.48)^3\, \dot{m}^*_{BL}\, h_n}$$

Lines 313-317

The subroutine prints out the quantities:

h_s, U_s, R_{es}, h_n, h_n / H_s, N_s, n_{eN} if IND = 1

Line 318

$$\theta_{BLS} = \left[\frac{C_{DT}\, A\, \rho_u}{2\pi \rho_s} \right]^{1/2} \frac{U_u}{U_s}$$

Lines 319-323

$$\theta^*_s = \begin{cases} \theta_{BLS} & \text{if } \dot{m}^*_{BLS} > \dot{m}^*_{\Sigma S2} \\ \left[\dfrac{\rho_u}{2\pi \rho_s} \left(C_{DV}\, A + \dfrac{\dot{m}^*_{\Sigma S2}\, C_{D\,\Sigma S2}\, A}{\dot{m}^*_{BLS}} \right) \right]^{1/2} \left[\dfrac{U_u}{U_s} \right] & \text{if } \dot{m}^*_{\Sigma S2} > \dot{m}^*_{BLS} \end{cases}$$

Line 324

$$R_{e\theta s} = \frac{\theta_s^* R_{es}}{D_B}$$

Lines 325-329

$$G_t = \left[1 - \frac{h_n(1 - \frac{h_n}{h_s})}{2(10^6)\ C_{90} \rho_s\ \theta_s^*\ U_s^2\ R_{e\theta s}\ (1 + h_n/H_S)}\right]^{-1} \quad \text{if } Z > Z_{BLT}$$

$$G_t = 0 \quad \text{if } Z \leq Z_{BLT}$$

Lines 330 - 335 compute the parameters

$$b_5 = C_{83}\left[1 + C_{84}\ M_{RAT}^{C_{85}} + C_{86}\ M_{RAT}^{C_{87}} + C_{88}\left(\frac{\rho_s}{\rho_{SL}}\right)^{C_{89}}\right]$$

$$b_{11} = C_{91} + C_{92}\ M_{RAT}^{C_{93}}$$

$$b_1 = \frac{\mu_u\left[C_{115} + b_{11}\left(\frac{\rho_s}{\rho_{SL}}\right)^{C_{125}}\right]}{10^6\ U_s^2\ \theta_{BLS}}$$

$$K_e = (\theta_s^*)^{-2} - \frac{7.5\,(10^{28})\,\rho_s\ C_{100}}{n_{eN}}$$

Lines 336 - 340

The subroutine prints out the quantities:

θ_{BLS}, θ_s^*, $R_{e\theta s}$, G_t, b_5, b_{11}, b_1, K_e **if IND = 1**

Lines 341 - 344

$$G_{t_{chem}} = \left[1 + \frac{h_n}{4\, C_{90}\, h_s\, K_e\, \rho_s\, \theta^*_s\, R_{e\theta s}} \right]^{-1} \qquad Z > Z_{BLT}$$

$$G_{t_{chem}} = 1 \quad \text{if} \quad Z \leq Z_{BLT}$$

Lines 345 - 348

$$n_{et} = \frac{10^{-3}\, \mu_u\, {}^{C_{134}}\, b_S\, {}^{C_{133}}\, \rho_5\, G_t}{\theta^*_s\, U_s\, {}^{C_{134}}} \left[1 - \exp\left(- |h_n| \left(C_{135} + C_{136} |h_n|^{C_{130}} + C_{131} |h_n|^{C_{132}} \right) \right) \right] + 1.11(10^{27}) \rho_s\, C_{100} + n_{eN}\, G_{tchem}$$

Lines 349 - 352

The subroutine prints out the quantities:

$G_{t_{chem}}$, n_{et} if IND = 1

Lines 353-355

The return to the calling subroutine is accomplished.

5. Other Information

A. SUBROUTINE FLØWF is called by SUBROUTINE WAKE.

B. SUBROUTINE FLØWF calls in SUBROUTINE AR2DIM and FUNCTION AR3DIM.

C. The following internal functions are called:

1. DEXP
2. DSIN
3. DCØTAN
4. DARSIN
5. DMAX

TABLE B

η_e AS A FUNCTION OF NORMALIZED DENSITY AND ENTHALPY

h/RT_o \ ρ/ρ_o	10^2	1.0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	0
0	0	0	0	0	0	0	0	0	0
30	0.72×10^{9}	0.72×10^{7}	0.45×10^{7}	0.19×10^{7}	0.58×10^{6}	0.92×10^{5}	0.52×10^{4}	0.80×10^{3}	↑
50	0.25×10^{14}	0.25×10^{12}	0.48×10^{11}	0.43×10^{10}	0.15×10^{8}	0.16×10^{8}	0.94×10^{6}	0.42×10^{5}	↑
70	0.20×10^{15}	0.20×10^{13}	0.91×10^{12}	0.60×10^{11}	0.28×10^{10}	0.16×10^{9}	0.60×10^{7}	0.32×10^{6}	↑
90	0.78×10^{16}	0.78×10^{14}	0.56×10^{13}	0.37×10^{12}	0.23×10^{11}	0.14×10^{10}	0.12×10^{9}	0.25×10^{7}	↑
110	0.35×10^{17}	0.35×10^{15}	0.29×10^{14}	0.27×10^{13}	0.26×10^{12}	0.32×10^{11}	0.31×10^{10}	0.32×10^{9}	↑
130	0.13×10^{18}	0.13×10^{16}	0.14×10^{15}	0.13×10^{14}	0.13×10^{13}	0.10×10^{12}	0.82×10^{10}	0.71×10^{9}	↑
150	0.34×10^{18}	0.34×10^{16}	0.34×10^{15}	0.30×10^{14}	0.24	0.18	0.14×10^{11}	0.11×10^{10}	↑
170	0.62×10^{18}	0.62×10^{16}	0.54×10^{15}	0.46×10^{14}	0.35	0.27	0.19	0.15	↑
190	0.90×10^{18}	0.90×10^{16}	0.80×10^{15}	0.68×10^{14}	0.50	0.36	0.27	0.18	↑
210	0.14×10^{19}	0.14×10^{17}	0.12×10^{16}	0.86×10^{14}	0.62	0.44	0.31	0.23	↑
230	0.17×10^{19}	0.17	0.14	0.10×10^{15}	0.76	0.53	0.40	0.27	↑
250	0.21	0.21	0.17	0.13	0.94×10^{13}	0.60	0.44	0.31	↑
270	0.25	0.25	0.21	0.15	0.10×10^{14}	0.74	0.51	0.42	↑
290	0.36	0.36	0.24	0.17	0.12	0.89	0.59	0.52	↑
310	0.38	0.38	0.28	0.20	0.14	0.97×10^{12}	0.68	0.60	↑
330	0.47	0.47	0.33	0.26	0.16	0.11×10^{13}	0.88	0.78	↑
350	0.56	0.56	0.42	0.28	0.19	0.13	0.95×10^{11}	0.90×10^{10}	↑
370	0.63	0.63	0.46	0.34	0.24	0.15	0.12×10^{12}	0.12×10^{11}	↑
390	0.70	0.70	0.57	0.38	0.28	0.23	0.16	0.14	↑
410	0.86	0.86	0.72	0.52	0.33	0.26	0.20	0.17	↑
430	0.20×10^{19}	0.20×10^{17}	0.83	0.62	0.45	0.35	0.25	0.24	↑
450	0.12×10^{20}	0.12×10^{18}	0.94×10^{16}	0.72×10^{15}	0.60	0.39	0.28	0.29	↑
470	0.14×10^{20}	0.14×10^{18}	0.14×10^{17}	0.10×10^{16}	0.74×10^{14}	0.45×10^{13}	0.42×10^{12}	0.33×10^{11}	0

KJTE: This table used when η_{Equil} is to be computed.

TABLE D

ELECTRON DENSITY AS A FUNCTION OF

NORMALIZED ENTHALPY AND AIR DENSITY

FOR 1000 ppm SODIUM SEED, M_{RAT} 0.0

$$n_e = n_e \left\{ \frac{h}{RT_o}, \rho/\rho_o \right\} \text{ [electrons/cc]}$$

$\frac{h}{RT_o}$ / ρ/ρ_o [ATM]		Greater than 0.1	0.1	0.03	0.01	Less than 0.01
Less than	15	0.0	0.0	0.0	0.0	0.0
	15	0.70×10^{1}	0.50×10^{1}	0.32×10^{2}	0.95×10^{2}	0.90×10^{2}
	20	0.38×10^{3}	0.38×10^{3}	0.12×10^{4}	0.30×10^{4}	0.30×10^{4}
	40	0.32×10^{10}	0.32×10^{10}	0.11×10^{10}	0.20×10^{9}	0.20×10^{9}
	60	0.25×10^{12}	0.25×10^{12}	0.78×10^{11}	0.05×10^{11}	0.05×10^{11}
	80	0.32×10^{13}	0.32×10^{13}	0.10×10^{13}	0.14×10^{12}	0.14×10^{12}
	100	0.17×10^{14}	0.17×10^{14}	0.65×10^{13}	0.20×10^{13}	0.20×10^{13}
	120	0.84×10^{14}	0.84×10^{14}	0.31×10^{14}	0.78×10^{13}	0.78×10^{13}
	140	0.24×10^{15}	0.24×10^{15}	0.87×10^{14}	0.22×10^{14}	0.22×10^{14}
	160	0.44×10^{15}	0.44×10^{15}	0.16×10^{15}	0.38×10^{14}	0.38×10^{14}
Greater than	160	0.44×10^{15}	0.44×10^{15}	0.16×10^{15}	0.38×10^{14}	0.38×10^{14}

TABLE D

ELECTRON DENSITY AS A FUNCTION OF NORMALIZED ENTHALPY AND AIR DENSITY FOR 1000 ppm SODIUM SEED, $M_{RAT} = 0.01$

$$n_e \quad n_e \left\{ \frac{h}{RT_o}, \rho/\rho_o \right\} \text{ [electrons/cc]}$$

ρ/ρ_o [ATM] / $\frac{h}{RT_o}$		Greater than 0.1	0.1	0.03	0.01	Less than 0.01
Less than	15	0.0	0.0	0.0	0.0	0.0
	15	1.0×10^{5}	1.0×10^{5}	5.5×10^{4}	2.0×10^{4}	2.0×10^{4}
	20	3.5×10^{7}	3.5×10^{7}	1.9×10^{7}	7.0×10^{6}	7.0×10^{6}
	40	2.0×10^{11}	2.0×10^{11}	1.1×10^{11}	4.0×10^{10}	4.0×10^{10}
	60	1.3×10^{12}	1.3×10^{12}	2.3×10^{11}	1.6×10^{11}	1.6×10^{11}
	80	2.0×10^{12}	2.0×10^{12}	4.5×10^{11}	2.0×10^{11}	2.0×10^{11}
	100	4.5×10^{12}	4.5×10^{12}	2.0×10^{12}	6.0×10^{11}	6.0×10^{11}
	120	1.4×10^{13}	1.4×10^{13}	5.0×10^{12}	1.4×10^{12}	1.4×10^{12}
	140	2.4 ↑	2.4 ↑	9.0×10^{12}	2.0 ↑	2.0 ↑
	160	3.0 ↓	3.0 ↓	1.5×10^{13}	3.0 ↓	3.0 ↓
greater than	160	4.0×10^{13}	4.0×10^{13}	2.5×10^{13}	4.0×10^{12}	4.0×10^{12}

TABLE D

ELECTRON DENSITY AS A FUNCTION OF

NORMALIZED ENTHALPY AND AIR DENSITY

FOR 1000 ppm SODIUM SEED, $M_{RAT} = 0.1$

$$n_e = n_e \left\{ \frac{h}{RT_o}, \rho/\rho_o \right\} \text{ [electrons/cc]}$$

$\frac{h}{RT_o}$ \ ρ/ρ_o [ATM]		Greater than 0.1	0.1	0.03	0.01	Less than 0.01
Less than	15	0.0	0.0	0.0	0.0	0.0
	15	2.4×10^{5}	2.4×10^{5}	1.2×10^{5}	4.5×10^{4}	4.5×10^{4}
	20	1.5×10^{8}	1.5×10^{8}	7.1×10^{7}	2.7×10^{7}	2.7×10^{7}
	40	7.6×10^{11}	7.6×10^{11}	3.7×10^{11}	1.4×10^{11}	1.0×10^{11}
	60	5.6×10^{11}	5.6×10^{12}	2.4×10^{12}	7.1×10^{11}	7.1×10^{11}
	80	1.3×10^{13}	1.3×10^{13}	5.8	1.7×10^{12}	1.7×10^{12}
	100	1.6	1.6	6.2	1.7	1.7
	120	2.1	2.1	7.7×10^{12}	2.0	2.0
	140	3.0	3.0	1.1×10^{13}	2.7	2.7
	160	4.0	4.0	1.5×10^{13}	3.5	3.5
Greater than	160	4.5×10^{13}	4.5×10^{13}	2.0×10^{13}	4.0×10^{12}	4.0×10^{12}

TABLE D

ELECTRON DENSITY AS A FUNCTION OF

NORMALIZED ENTHALPY AND AIR DENSITY

FOR 1000 ppm SODIUM SEED, $M_{RAT} = 1.0$

$$n_e = n_e \left\{ \frac{h}{RT_o}, \rho/\rho_o \right\} \text{ [electrons/cc]}$$

$\frac{h}{RT_o}$ \ ρ/ρ_o [ATM]		Greater than 0.1	0.1	0.3	0.01	Less than 0.01
Less than	15	0.0	0.0	0.0	0.0	0.0
	15	7.9×10^{5}	7.9×10^{5}	5.2×10^{5}	1.7×10^{5}	1.7×10^{5}
	20	2.3×10^{8}	2.3×10^{8}	1.5×10^{8}	5.0×10^{7}	5.0×10^{7}
	40	2.0×10^{12}	2.0×10^{12}	1.3×10^{12}	4.3×10^{11}	4.3×10^{11}
	60	1.8×10^{13}	1.8×10^{13}	8.0×10^{12}	2.5×10^{12}	2.5×10^{12}
	80	6.0×10^{13}	6.0×10^{13}	3.0×10^{13}	1.1×10^{13}	1.1×10^{13}
	100	1.4×10^{14}	1.4×10^{14}	5.7 ↑	1.5 ↑	1.5 ↑
	120	1.3 ↑↓	1.3 ↑↓	5.2 ↕	1.3 ↕	1.3 ↕
	140	1.2 ↕	1.2 ↕	4.7 ↕	1.2 ↕	1.2 ↕
	160	1.5 ↓	1.5 ↓	5.0 ↓	1.3 ↓	1.3 ↓
greater than	160	2.0×10^{14}	2.0×10^{14}	6.0×10^{13}	2.0×10^{13}	2.0×10^{13}

TABLE E

TABLE OF M VERSUS M_C AND θ_C

MC \ θ_C	0	2	6	10	14	18	22	26	30	45	45
1.0	1.0	1.13	1.29	1.43	1.57	1.70	1.84	2.00	2.13	2.76	100
3.0	3.0	3.11	3.33	3.58	3.85	4.15	4.49	4.88	5.32	7.79	100
6.0	6.0	6.30	7.00	7.84	8.89	10.2	12.0	14.6	18.4	100	100
9.0	9.0	9.64	11.2	13.4	16.7	21.8	31.6	53.0	100	100	100
12.0	12.0	13.1	16.2	21.0	29.8	51.0	100	100	100	100	100
15.0	15.0	16.8	22.0	33.4	58.0	100	100	100	100	100	100
18.0	18.0	20.6	29.0	49.2	100	100	100	100	100	100	100
21.0	21.0	24.6	37.7	80.0	100	100	100	100	100	100	100
24.0	24.0	28.8	48.6	100	100	100	100	100	100	100	100
27.0	27.0	33.4	63.0	100	100	100	100	100	100	100	100
30.0	30.0	38.0	81.0	100	100	100	100	100	100	100	100
> 30.0	100	100	100	100	100	100	100	100	100	100	100

TABLE F

EQUILIBRIUM NORMAL SHOCK ELECTRON DENSITY

ρ lb/ft^3	Velocity 10kfps	14kfps	18kfps	22 kfps	26 kfps	30 kfps	
.0804	1.4×15^{12}	5×10^{15}	9.1×10^{16}	3.6×10^{17}	1×10^{18}	2×10^{18}	
.01171	2.3×10^{12}	8.1×10^{14}	1.4×10^{16}	5.4×10^{16}	1.5×10^{17}	2.2×10^{17}	
.001068	6×10^{11}	1.1×10^{14}	1.9×10^{15}	7×10^{15}	1.8×10^{16}	2.8×10^{16}	
.00033	1.2×10^{10}	9.2×10^{12}	1.7×10^{14}	7.4×10^{14}	1.8×10^{15}	3×10^{15}	
1.112×10^{-4}	6.7×10^{8}	7.7×10^{11}	2×10^{13}	8.8×10^{13}	2.1×10^{14}	3.5×10^{14}	
4.261×10^{-5}	4.3×10^{7}	1×10^{11}	3.5×10^{12}	1.5×10^{13}	4×10^{13}	7×10^{13}	
1.7×10^{-5}	2×10^{10}	2×10^{10}	7.2×10^{11}	3.1×10^{12}	7.2×10^{12}	1×10^{13}	
2.26×10^{-6}	1.15×10^{6}	4×10^{9}	2.1×10^{11}	7×10^{11}	1.3×10^{12}	2×10^{12}	
FOR: $\rho \leq 2.26 \times 10^{-6}$	Use Value for			$= 2.26 \times 10^{-6}$			
$\rho \geq .0804$	Use Value for			$= .0804$			
$V < 10$	Use Linear Interpolation to $n_e = 10^2$						
$V \geq 30$	Use Value for $V = 30$ kfps						

FUNCTION AR3DIM (D, N20, M20, N10, RØW, H, EM, XYZTBL)

1. Purpose

FUNCTION AR3DIM is the result of a table look-up employing linear interpolation to determine the value of a dependent variable which is a function of three independent variables, i.e. f(X, Y, Z)

2. Input

*indicates integer quantity

Name	Description
D(N20, M20, N10)	the dependent variable table
EM	the value of the independent variable Z for which f is desired
H	the value of the independent variable Y for which f is desired
M20 *	the number of Y values used in forming the table
N10 *	the number of Z values used in forming the table
N20 *	the number of X values used in forming the table
RØW	the value of the independent variable X for which f is desired
XYZTBL (11, 3)	the table containing the three independent variables; the second integer indicates the related independent variable in the following manner: 1 indicates X tabular values 2 indicates Y tabular values 3 indicates Z tabular values

3. Output

Name	Description
AR3DIM	the value of the dependent variable f resulting from 3 dimensional linear interpolation

4. Numerical Procedure

FUNCTION AR3DIM utilizes a DØ loop to test the values in the XYZTBL(K, 3) array, Z tabular values, starting with the second value until it reaches the first value which either exceeds or equals the input EM, the specified Z value. This value is XYZTBL(I, 3). If this criterion is not satisfied, the last value in the array is used as the appropriate value in subsequent calculations.

SUBROUTINE AR2DIM is called for the (I-1) value in the table of Z values to interpolate for the value of f at the specified X and Y; this is called ANS1. SUBROUTINE AR2DIM is called in a second time to perform the same function for the Ith value in the table of Z values; this result is designated ANS2. These quantities are then used to define AR3DIM from the following equation:

$$AR3DIM = \frac{ANS1(XYZTBL(I,3)-EM) + ANS2(EM-XYZTBL(I-1,3))}{XYZTBL(I,3) - XYZTBL(I-1,3)}$$

5. Other Information

A. FUNCTION AR3DIM is utilized by SUBROUTINE FLOWF.

B. FUNCTION AR3DIM calls SUBROUTINE AR2DIM.

SUBROUTINE AR2DIM (N, M, X, XTB, Y, YTB, Z, ZTB)

1. Purpose

SUBROUTINE AR2DIM performs a linear interpolation in a two-dimensional table for Z as a function of X and Y.

2. Input

*indicates integer quantity

Name		Description
M	*	number of Y values used in forming the table
N	*	number of X values used in forming the table
X		the value of the first independent variable for which the value of Z is desired
XTB(N)		tabular values of the independent variable X
Y		the value of the second independent variable for which the value of Z is desired
YTB(M)		tabular values of the independent variable Y
ZTB(M, N)		tabular values of the dependent variable, Z

3. Output

Z	the value of the dependent variable Z at the point X, Y obtained by a double linear interpolation in the table of ZTB as function of XTB and YTB

4. Numerical Procedure

SUBROUTINE AR2DIM utilizes a DØ loop to test the values in the XTB array starting with the second value until it reaches the first one which either equals or exceeds the input value X. This appropriate XTB value, XTB(I+1) together with the preceding tabular value XTB(I) is then used to define the quantity P_2

$$P_2 = \frac{X - XTB(I)}{XTB(I+1) - XTB(I)}$$

An identical procedure is undertaken with the Y variable and the YTB table in order to define

$$P_1 = \frac{Y - YTB(J)}{YTB(J+1) - YTB(J)} .$$

These quantities are then used in the calculation of Z from the equation

$$Z = (1.0 - P_1 - P_2 + P_1 * P_2)\ ZTB(J, I) + (P_2 - P_1 * P_2)\ ZTB(J, I+1)$$

$$+ (P_1 - P_1 * P_2) * ZTB(J+1, I) + P_1 * P_2 * ZTB(J+1,\ I+1)$$

5. Other Information

A. SUBROUTINE AR2DIM is called from FUNCTION AR3DIM.

B. SUBROUTINE AR2DIM calls no other subprograms.

C. If either X or Y should exceed respectively all the tabular inputs for XTB and YTB the last value in the effected table is used in the definitions.

3.2.3 Radar Cross Section Calculations

RCSEC

1. Purpose

To compute radar cross section and pulse shapes from the given observing radar and the transition point electron density of the vehicle and also compute wake length.

2. Input

Name	Symbol	Source of Input	Desc
B1	b_1	SUBROUTINE WAKE	exponential dec
ZNET	ηe_t	SUBROUTINE WAKE	transition point
NØPT*	NØPT	SUBROUTINE WAKE	NØPT = 0, co NØPT = 1, co
F	f	SUBROUTINE WAKE	frequency
TAU	τ	SUBROUTINE WAKE	pulse length
SIGMDS	σ_{MDS}	SUBROUTINE WAKE	noise level to w
PHI	ϕ	SUBROUTINE WAKE	look angle
XBZ	B_z	SUBROUTINE WAKE	scale height
ZME	M_e	SUBROUTINE WAKE	Mach number (
HH	h	SUBROUTINE WAKE	altitude
DDW	D_w	SUBROUTINE WAKE	wake diameter
C1	C_1	SUBROUTINE WAKE	input constant
C2	C_2	SUBROUTINE WAKE	input constant
C3	C_3	SUBROUTINE WAKE	input constant
C4	C_4	SUBROUTINE WAKE	input constant
C5	C_5	SUBROUTINE WAKE	input constant
BZERØ	b_0	SUBROUTINE WAKE	scaling constan
B2	b_2	SUBROUTINE ZREADX	scaling constan

	Description	Units
AKE	exponential decay constant	1/meters
AKE	transition point electron density	e/cc
AKE	NØPT = 0, compute SIGP only NØPT = 1, compute SIGP and WAKEL	dimensionless
AKE	frequency	CPS
AKE	pulse length	μ sec.
AKE	noise level to which wake length is measured	$(\text{meters})^2$
AKE	look angle	degrees
AKE	scale height	1000 ft.
AKE	Mach number (upstream)	dimensionless
AKE	altitude	1000 ft.
AKE	wake diameter = base diameter	meters
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	scaling constant	dimensionless
READX	scaling constant	dimensionless

B

2. Input (Cont'd)

Name	Symbol	Source of Input	Descri
B3	b_3	SUBROUTINE ZREADX	scaling cons
BTWEN	b_{20}	SUBROUTINE ZREADX	scaling cons
B24	b_{24}	SUBROUTINE ZREADX	scaling cons
DX	Δx	SUBROUTINE ZREADX	step in axial
ZNUS	γ_{SL}	SUBROUTINE ZREADX	sea level col
CNE	NET	SUBROUTINE ZREADX	transition el production t
DSB	Δr_{2B}	SUBROUTINE ZREADX	additional R production t
X2BØD	$X_{2B\emptyset D}$	SUBROUTINE ZREADX	2 body overd
X3B	X_{3B}	SUBROUTINE ZREADX	station wher dominate th
IND2*	IND2	SUBROUTINE ZREADX	IND2 = 0; no IND2 = 1; int
NSTWL*	NSTWL	SUBROUTINE ZREADX	maximum nu wake length

t	Description	Units
EADX	scaling constant	dimensionless
EADX	scaling constant	dimensionless
EADX	scaling constant	dimensionless
EADX	step in axial coordinate	meters
EADX	sea level collision frequency	CPS
EADX	transition electron density when non-linear production terms are considered in turbulent wake	e/cc
EADX	additional RCS due to consideration on non-linear production terms in turbulent wake	$(\text{meters})^2$
EADX	2 body overdense length	meters
EADX	station where linear production terms first dominate the non-linear production terms	meters
EADX	IND2 = 0; no output generated by this routine IND2 = 1; intermediate steps printed out	dimensionless
EADX	maximum number of steps used to compute wake length	dimensionless

B

2. Input

Name	Symbol	Source of Input	D
B1	b_1	SUBROUTINE WAKE	exponential
ZNET	η_{e_t}	SUBROUTINE WAKE	transition p
NØPT*	NØPT	SUBROUTINE WAKE	NØPT = 0, NØPT = 1,
F	f	SUBROUTINE WAKE	frequency
TAU	τ	SUBROUTINE WAKE	pulse length
SIGMDS	σ_{MDS}	SUBROUTINE WAKE	noise level
PHI	ϕ	SUBROUTINE WAKE	look angle
XBZ	B_z	SUBROUTINE WAKE	scale height
ZME	M_e	SUBROUTINE WAKE	Mach numbe
HH	h	SUBROUTINE WAKE	altitude
DDW	D_w	SUBROUTINE WAKE	wake diame
C1	C_1	SUBROUTINE WAKE	input consta
C2	C_2	SUBROUTINE WAKE	input consta
C3	C_3	SUBROUTINE WAKE	input consta
C4	C_4	SUBROUTINE WAKE	input consta
C5	C_5	SUBROUTINE WAKE	input consta
BZERØ	b_0	SUBROUTINE WAKE	scaling cons
B2	b_2	SUBROUTINE ZREADX	scaling cons

	Description	Units
AKE	exponential decay constant	1/meters
AKE	transition point electron density	e/cc
AKE	NØPT = 0, compute SIGP only NØPT = 1, compute SIGP and WAKEL	dimensionless
AKE	frequency	CPS
AKE	pulse length	μ sec.
AKE	noise level to which wake length is measured	$(\text{meters})^2$
AKE	look angle	degrees
AKE	scale height	1000 ft.
AKE	Mach number (upstream)	dimensionless
AKE	altitude	1000 ft.
AKE	wake diameter = base diameter	meters
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	input constant	dimensionless
AKE	scaling constant	dimensionless
READX	scaling constant	dimensionless

B

2. Input (Cont'd)

Name	Symbol	Source of Input	Descrip
B3	b_3	SUBROUTINE ZREADX	scaling const
BTWEN	b_{20}	SUBROUTINE ZREADX	scaling const
B24	b_{24}	SUBROUTINE ZREADX	scaling const
DX	Δx	SUBROUTINE ZREADX	step in axial
ZNUS	γ_{SL}	SUBROUTINE ZREADX	sea level coll
CNE	NET	SUBROUTINE ZREADX	transition ele production te
DSB	$\Delta\sigma_{2B}$	SUBROUTINE ZREADX	additional RC production te
X2BØD	$X_{2B\emptyset D}$	SUBROUTINE ZREADX	2 body overd
X3B	X_{3B}	SUBROUTINE ZREADX	station where dominate the
IND2*	IND2	SUBROUTINE ZREADX	IND2 = 0; no IND2 = 1; int
NSTWL*	NSTWL	SUBROUTINE ZREADX	maximum nu wake length

	Description	Units
DX	scaling constant	dimensionless
DX	scaling constant	dimensionless
DX	scaling constant	dimensionless
DX	step in axial coordinate	meters
DX	sea level collision frequency	CPS
DX	transition electron density when non-linear production terms are considered in turbulent wake	e/cc
DX	additional RCS due to consideration on non-linear production terms in turbulent wake	$(\text{meters})^2$
DX	2 body overdense length	meters
DX	station where linear production terms first dominate the non-linear production terms	meters
DX	IND2 = 0; no output generated by this routine IND2 = 1; intermediate steps printed out	dimensionless
DX	maximum number of steps used to compute wake length	dimensionless

B

3. Output

Name	Symbol	Description	Units
SIGP	σ_p	radar cross section	(meters)^2
WAKEL	L_w	wake length	meters
LP*	LP	set to 6 if number of steps required to compute wake length exceeds NSTWL	dimensionless

4. Numerical Procedure

Compute ρ = atmospheric density

$$\rho = R\emptyset = e^{-\frac{h}{BZ}}$$

Compute $\eta_{e_{CR}}$ = electron density at which plasma becomes overdense

$$\eta_{e_{CR}} = ZNCR = \left(\frac{ft^2}{10^8}\right)\left[1 + \rho^2\left(\frac{\nu_{SL}}{f^2}\right)^2\right]^{C_5}$$

Compute X_t = transition pt. onset of overdense region

$$X_t = XTT = \frac{C_1}{\rho}\left(1 - \frac{C_2}{M_e{}^{C_3}}\right) + C_4$$

Test on ϕ: If $\phi = 0.0$, change to 0.01

If $\phi > 90.0$, change to $180.0 - \phi$

Compute $X_{\emptyset D}$ = onset of constant multiple scattering region

$$X\emptyset D = \frac{1}{b_1}\left[\ln\left(\frac{\eta_{e_t}}{\eta_{e_{CR}}}\right)^2\right]; \quad X2B\emptyset D \geq X3B$$

$$X\emptyset D = X2B\emptyset D \quad ; \quad X2B\emptyset D < X3B$$

Compute $X_{\emptyset DP}$ which replaces XØD in all equations

$$X\emptyset DP = X\emptyset D; \quad X\emptyset D \geq 0$$

$$X\emptyset DP = 0.0; \quad X\emptyset D < 0$$

Compute $L_{\emptyset D}$ = smaller of length from XT to XØD and the range resolution

$$L_{\emptyset D} = CL\emptyset D = \text{minimum of } \left[\frac{3\tau \times 10^2}{2\cos\phi}\right] \text{and } X\emptyset DP$$

Compute X_{SS} = pt. of onset of single scattering region

$$XSS = \frac{1}{b_1}\left[\ln\left[\frac{\eta e_t}{\eta e_{CR}\sin^2\phi}\right]^2\right]$$

Compute X_{SSP} which replaces XSS in all equations

$$XSSP = X_{SS}\ ;\quad XSS \geq 0$$

$$XSSP = 0\ ;\quad XSS < 0$$

Compute L_{SS} = smaller of distance to XSS and the range resolution

$$L_{SS} = CLSS = \text{minimum of } \left(\frac{3\tau \times 10^2}{2\cos\phi}\right) \text{and } X_{SSP}$$

Compute D_w^* = surface scattering correction term for base diameter

$$D_w^* = DWST = \left[\frac{(K\, b_{24}\, D_w)^3}{2\sin^2\phi + (K\, b_{24}\, D_w)^3}\right] D_w$$

$$\text{where } K = 2\pi f^2 \times 10^9$$

Compute X_{MS} = pt. of onset of variable multiple scattering region

$$XMS = X\emptyset DP;\ b_{20} = 0$$

$$XMS = \frac{2}{b_1 b_{20}} \ln\left[\left[\frac{b_o\left(\frac{f^2}{10^8}\right)^2 D_w^4}{\left[1 + b_2\left(\frac{f^2}{10^8}\right) D_w^2\right]^{b_3}}\right]^{b_{20}} \left(\frac{\eta e_t}{\eta e_{CR}}\right) \sin\phi^{(4-2b_{20})}\right];\quad b_{20} \neq 0$$

$$XMSP = XMS;\ XMS \geq X\emptyset DP$$

$$XMSP = X\emptyset DP;\ XMS < X\emptyset DP$$

$$XLM\emptyset D = \text{MAXIMUM}\,(XMSP,\ X\emptyset DP)$$

$$XLMSD = \text{MINIMUM}\,(XLM\emptyset D,\ \frac{150\tau}{\cos\phi})$$

Compute L_{MS} = smallest of half the length from XØDP to XMSP or half of the range resolution minus XØDP

$$LMS = XLMS = 0.0;\quad XMSP \leq X\emptyset DP$$

$$XLMS = \text{MINIMUM}\,(\frac{1}{2}\,(\frac{150\tau}{\cos\phi} - X\emptyset DP),\ \frac{1}{2}\,(XMSP - X\emptyset DP))$$

Compute σ_p = radar cross section

Intermediate steps:

$$E_1 = e^{-b_1 L\emptyset D};\quad E_2 = e^{-b_1 LSS};\quad E_3 = e^{-b_1 \frac{150\tau}{\cos\phi}}$$

$$TRM1 = \left[L\emptyset D\ D_w^{*}\ \sin^2\phi + \frac{D_w^{*}}{b_1}\left(\frac{\eta e_t}{e_{CR}}\right)^2 \left[(E_1 - E_3)\ \frac{\sin^2\phi}{16} + \frac{b_o\left(\frac{f^2}{10^8}\right)^2 D_w^4}{\left[1 + b_2\left(\frac{f^2}{10^8}\right) D_w^2\right]^{b3}}\ (E_2 - E_3) \right] + LMS\ D_w^{*}\ \sin^2\phi \right]$$

$$E_1 = e^{(-4b_1 L\emptyset D)} - e^{(-b_1 \frac{600\tau}{\cos\phi})}$$

$$\sigma_p = \text{SIGP} = \text{TRM1} + \frac{D_w^* \sin^2\phi}{4b_1} \left(\frac{\eta e_t}{\eta e_{CR}}\right)^8 E_1$$

$$+ 2 D_w^* \left[\frac{b_o \left(\frac{f^2}{10^8}\right)^2 D_w^4}{\left[1 + b_2 \left(\frac{f^2}{10^8}\right) D_w^2\right]^{b_3}}\right] \frac{(\sin^2\phi)^{(2-b_{20})}}{b_1 \; b_{20}} \qquad b_{20} \neq 0$$

$$\left(\frac{\eta e_t}{\eta e_{CR}}\right)^{b_{20}} \left(e^{\frac{-b_1 b_{20} L_{MSD}}{2}} - e^{\frac{-b_1 b_{20} L_{SS}}{2}}\right)$$

$$\text{SIGP} = \text{TRM1} + \frac{D_w^* \sin^2\phi}{4b_1} \left(\frac{\eta e_t}{\eta e_{CR}}\right)^8 E_1$$

$$+ D_w \left[\frac{b_o \left(\frac{f^2}{10^8}\right)^2 D_w^4}{\left[1 + b_2 \left(\frac{f^2}{10^8}\right) D_w^2\right]^{b_3}}\right] (\text{LSS-LMSD}) \sin^4\phi \qquad b_{20} = 0$$

If NØPT = 0, return to calling program (WAKE)

If NØPT = 1, compute wake length (WAKEL)

Compute wake length

The σ vs. X curve has a maximum or peak and is a monotonically increasing and decreasing curve for X < XPEAK and X > XPEAK respectively. Therefore, starting at X_t, and increasing X by ΔX, SIG is found for each value of X using function FUN1 until XPEAK is reached.

Using the values of X on either side of the peak as arguments, the subroutine FIB1 is called to determine exact σpeak and corresponding XPEAK. If σPEAK $\leq \sigma_{MDS}$, WAKEL is set to 0.0 and control is returned to the calling subroutine WAKE. Otherwise, routine starts at XPEAK and moves along in steps of ΔX, computing σ for each X, until $\sigma \leq \sigma_{MDS}$. It then calls subroutine FIB1 using as arguments the value of X just found and the previous value, to determine X_{NL}

WAKEL = XNL - XPEAK

RETURN TO CALLING PROGRAM WAKE

5. Other Information

A. SUBROUTINE RCSEC is called by SUBROUTINE WAKE

B. SUBROUTINE RCSEC calls in function FUN1 and SUBROUTINE FIB1

C. SUBROUTINE RCSEC calls the library functions

1. DSIN
2. DCØS
3. DLØG
4. DMIN1
5. DMAX1
6. DEXP
7. DABS
8. DFLØAT

SUBROUTINE FIB1 (AA, BB, NF, NMIMAX, ACCUR, NFUNC, XMIMAX, YMIMAX)

1. Purpose

SUBROUTINE FIB1 utlizes a Fibonacci search technique to find the maximum or minimum of a one variable unimodal function within a defined region (AA, BB).

2. Input

*indicates integer quantity

Name	Source of Input	Description
AA	RCSEC	one boundary of the defined region within which search will take place
ACCUR	RCSEC	desired accuracy
BB	RCSEC	one boundary of the defined region within which search will take place
FUN1	Function FUN1	the value of the function being optimized at the current evaluation point within the search interval
NF	RCSEC*	code number of the function to be optimized
NFUNC	RCSEC*	the number of values of the function to be utilized in the optimization
NMIMAX	RCSEC*	integer code; value of +1 calls for maximizing calculation; value of -1 calls for minimization

3. Output

Name	Description
NF*	see input
XL	the smaller of the two evaulation points within the current search interval (A,B)
XMIMAX	the optimum value of independent variable
XR	the larger of the two evaluation points within the current search interval (A, B)
YMIMAX	XMIMAX value of independent variable

4. Numerical Procedure

SUBROUTINE FIB1 is a duplicate of SUBROUTINE MIMAX with the following exceptions:

(1) where MIMAX calls in FUNCTION FMIMAX, FIB1 calls instead FUNCTION FUN1 at locations following statements 50, 60, 80, 120, and 140.

(2) the common block END containing the variable ITERM is not used in FIB1

(3) ITERM is set equal to zero at the beginning of FIB1

(4) the Fibonacci number E is dimensioned for 40 elements in FIB1 instead of the 100 elements specified in MIMAX.

5. Other Information

A. SUBROUTINE FIB1 is called by SUBROUTINE RCSEC only.

B. SUBROUTINE FIB1 calls FUNCTION FUN1.

C. SUBROUTINE FIB1 calls the IBM routines DSQRT and FDXPI (exponentiation).

FUNCTION FUN1

1. **Purpose:**

To compute pulse shape- radar return as a function of distance behind the leading edge of the radar pulse.

2. Input

Name	Symbol	Source of Input	
XX	X	SUBROUTINE RCSEC	axial coordi
N*	N	SUBROUTINE RCSEC	N = 1; =
			N = 2; = (
XTT	X_t	SUBROUTINE RCSEC	transition pe
WS1	$\frac{150\tau}{\cos\phi}$	SUBROUTINE RCSEC	range resol
XØDP	$X_{\emptyset DP}$	SUBROUTINE RCSEC	see RCSEC
DWST	D_w^*	SUBROUTINE RCSEC	surface scat
SPH2	$\sin^2\phi$	SUBROUTINE RCSEC	intermediate
XLMS	L_{ms}	SUBROUTINE RCSEC	see RCSEC
XMSP	X_{msp}	SUBROUTINE RCSEC	see RCSEC
XSSP	X_{ssp}	SUBROUTINE RCSEC	see RCSEC
B1	b_1	SUBROUTINE RCSEC	exponential
ZCR2	$(\frac{\eta et}{\eta eCR})^2$	SUBROUTINE RCSEC	intermediate
ZCR4	$(\frac{\eta et}{\eta eCR})^4$	SUBROUTINE RCSEC	intermediate
XB20	b_{20}	SUBROUTINE RCSEC	scaling cons
DWBST1	$(Dw^* \sin^2\phi)/b_1$	SUBROUTINE RCSEC	intermediate
SPH4	$\sin^4\phi$	SUBROUTINE RCSEC	intermediate

	Description	Units
CSEC	axial coordinate	meters
CSEC	N = 1; = SIG N = 2; = (SIGMDS - SIG)^2	dimensionless
CSEC	transition point onset of overdense region	meters
CSEC	range resolution	μ sec.
CSEC	see RCSEC	meters
SEC	surface scattering correction term for base diameter	meters
CSEC	intermediate calculation	dimensionless
CSEC	see RCSEC	meters
CSEC	see RCSEC	meters
CSEC	see RCSEC	meters
CSEC	exponential decay constant	1/meters
CSEC	intermediate calculation	dimensionless
CSEC	intermediate calculation	dimensionless
CSEC	scaling constant	dimensionless
CSEC	intermediate calculation	dimensionless
CSEC	intermediate calculation	dimensionless

B

2. Input (Concl'd)

Name	Symbol	Source of Input	Description
B0	b_o	SUBROUTINE RCSEC	scaling consta
DDW	D_w	SUBROUTINE RCSEC	wake diameter
SIGMDS	σ_{MDS}	SUBROUTINE RCSEC	noise level to
WS4	$\left(\frac{\eta_{et}}{\eta_{eCR}}\right)^{b_{20}}$	SUBROUTINE RCSEC	intermediate
IND2*	IND2	SUBROUTINE RCSEC	IND2 = 0; no IND2 = 1; inte

3. Output

FUN1	σ : For N = 1 $\sigma_{MDS}-\sigma)^2$: For N=2	pulse shape

Description	Units
scaling constant	dimensionless
wake diameter = base diameter	meters
noise level to which wake length is measured	$(\text{meters})^2$
intermediate calculation	dimensionless
IND2 = 0; no output generated IND2 = 1; intermediate steps printed out	dimensionless
pulse shape	$(\text{meters})^2$

B

4. Numerical Description

Given a value of X, compute FUN1 where

$$FUN1 = \sigma \qquad N = 1$$

$$FUN1 = (\sigma_{MDS} - \sigma)^2 ; \quad N = 2$$

and $\sigma = \sigma_1 + \sigma_2 + \sigma_3 + \sigma_4$

Compute Y_1 = region affected by overdense scattering

$Y_1 = Y1$ = minimum of the following:

1. $\frac{150\tau}{\cos\phi}$
2. $X_{\phi DP}$
3. $(X - X_\tau)$
4. $X_\tau + X_{\phi DP} + \frac{150\tau}{\cos\phi} - X$

Compute σ_1 = portion of pulse shape due to overdense return

$$\sigma_1 = SIG1 = 0 \quad : \quad (X - X_\tau) < 0$$

$$= 0 \quad : \quad \left[(X - X_\tau) - \frac{150\tau}{\cos\phi} - X_{\phi DP}\right] > 0$$

$$= Y_1 D_w^* \sin^2\phi : \quad (X - X_\tau) \geq 0 \text{ and } \left[(X - X_\tau) - \frac{150\tau}{\cos\phi} - X_{\phi DP}\right] \leq 0$$

Compute Y_2 = region affected by first part of multiple scattering

$Y_2 = Y2$ = minimum of the following:

1. $2 L_{ms}$
2. $\frac{150\tau}{\cos\phi}$
3. $(X - X_\tau) - X_{\phi DP}$
4. $X_\tau + X_{MSP} + \frac{150\tau}{\cos\phi} - X$

Compute σ_2 = portion of pulse shape due to constant portion of multiple scattering region

$$\sigma_2 = SIG2 = 0 \quad : \quad (X - X_\tau) < X_{\phi DP}$$

$$= 0 \quad : \quad \left[(X - X_\tau) - \frac{150\tau}{\cos\phi} - X_{MSP}\right] > 0$$

$$= \frac{1}{2} Y_2 D_w^* \sin^2\phi : (X - X_\tau) \geq X_{\phi DP}$$

$$\text{and } \left[(X - X_\tau) - \frac{150\tau}{\cos\phi} - X_{MSP}\right] \leq 0$$

Compute Y_{31} = upper limit of region affected by second part of multiple scattering

$$Y_{31} = Y31 = X - \frac{150\tau}{\cos\phi} - X_\tau : \quad \frac{150\tau}{\cos\phi} < \left[X - X_\tau - X_{MSP}\right]$$

$$= X_{MSP} \quad : \quad \frac{150\tau}{\cos\phi} \geq \left[X - X_\tau - X_{MSP}\right]$$

Compute Y_{32} = lower limit of region affected by second par of multiple scattering

$$Y_{32} = Y32 = X - X_\tau \quad : \quad X \leq \left[X_\tau + X_{SSP}\right]$$

$$= X_{SSP} \quad : \quad X > \left[X_\tau + X_{SSP}\right]$$

Compute σ_3 = portion of pulse shape due to variable portion of multiple scattering region

$$\sigma_3 = SIG3 = 0 \quad : \quad (X - X_\tau) < X_{MSP}$$

$$= 0 \quad : \quad (X - X_\tau) > \left(\frac{150\tau}{\cos\phi} + X_{SSP}\right)$$

$$\sigma_s = \left[\left(\frac{D_w^* \sin^2\phi}{b_1} \right) \left(EX1 + EX2 \right) + \left(Y_{32} - Y_{31} \right) \sin^4\phi \, b_o \, D_w \right] : \quad (X - X_T) \geq X_{MSP}$$

$$(X - X_T) \leq \left(\frac{150\tau}{\cos\phi} + X_{SSP} \right)$$

$$b_{20} = 0.0$$

$$= \left[\frac{(D_w^* \sin^2\phi)}{b_1} \left(EX1 + EX2 \right) + 2 \left(\frac{\eta_{eT}}{\eta_{eCR}} \right)^{b_{20}} \left(e^{\frac{-b_1 b_{20} Y_{31}}{2}} - e^{\frac{-b_1 b_{20} Y_{32}}{2}} \right) \right. \quad : \quad (X - X_t) \geq X_{MSP}$$

$$\left. \frac{b_o \sin^2\phi}{b_1 b_{20}} \right)^{(2 - b_{20})} D_w^* \Bigg] \qquad (X - X_T) \leq \left(\frac{150\tau}{\cos\phi} + X_{SSP} \right)$$

$$b_{20} \neq 0$$

where $EX1 = \dfrac{\left[e^{-b_1 Y_{31}} - e^{-b_1 Y_{32}} \right]}{16} \left(\dfrac{\eta_{eT}}{\eta_{eCR}} \right)^2$

$$EX2 = \frac{\left[e^{-4b_1 Y_{31}} - e^{-4b_1 Y_{32}} \right]}{4} \left(\frac{\eta_{eT}}{\eta_{eCR}} \right)^8$$

Compute Y_{41} = upper limit of region affected by single scattering

$$Y_{41} = Y41 = X_{SSP} \quad : \frac{150\tau}{\cos\phi} \geq (X - X_\tau) - X_{SSP}$$

$$= (X - X_\tau) - \frac{150\tau}{\cos\phi} \quad : \frac{150\tau}{\cos\phi} < (X - X_\tau) - X_{SSP}$$

Compute Y_{42} = lower limit of region affected by single scattering

$$Y_{42} = Y42 = (X - X_\tau)$$

Compute σ_4 = portion of return due to single scattering region

$$\sigma_4 = SIG4 = 0$$

$$= \left[\frac{1}{4}\left(\frac{D_w^* \sin^2\phi}{b_1}\right) EX1 \left(\frac{\eta_{eT}}{\eta_{eCR}}\right)^8 + \right. \qquad : (X - X_\tau) > X_{SSP}$$

$$\left. \frac{D_w}{b_1} \left[\frac{\sin^2\phi}{16} \frac{D_w^*}{D_w} + b_o\right] \left(\frac{\eta_{eT}}{\eta_{eCR}}\right)^2 EX2\right]$$

where

$$EX1 = \left[e^{-4b_1 Y_{41}} - e^{-4b_1 Y_{42}}\right]$$

$$EX2 = \left[e^{-b_1 Y_{41}} - e^{-b_1 Y_{42}}\right]$$

5. Other Information

A. FUNCTION FUN1 is called by SUBROUTINE FIB1.

B. FUNCTION FUN1 calls the library functions:

1. DMIN1
2. DEXP
3. DABS

3.3 Miscellaneous

Miscellaneous calculations are performed by subroutines POLCAL and MISC.

SUBROUTINE MISC(N, X, VAL)

1. Purpose

At present MISC can call SUBROUTINE PØLCAL to calculate the free space radar cross section of a vehicle. However, the first three arguments of PØLCAL have been generalized so that any three elements of the OCCUR array can be used. It calculates the elements of the GD array of generalized differences between up to twenty pairs of elements of the OCCUR array.

MISC is intended, in general, to perform any tasks which do not seem to fit readily into any of the other subroutines. It is expected to grow in the future.

2. Input

Unless otherwise specified, all numbers in the COMMON LOCATION refer to positions in the ØCCUR array.

Name	Source	Common Location	
ACØE, 140	ZREADX	PCCUR (5881)	coefficients for
LAMDA1	READIT	137	the input initial
LAMDA2	READIT	143	the input initial
LAMD1F	CHNTBL	151	the value of blu
LA1	READIT	138	the input initial
LA2	READIT	144	the input initial
LA1F	CHNTBL	146	the value of axi
PI	SR2490	42	
RB1	READIT	136	the input initial
RB2	READIT	142	the input initial
RB1F	CHNTBL	147	the value of bas
RN1	READIT	135	the input initial
RN2	READIT	141	the input initial
RN1F	CHNTBL	169	the value of nos
THETA1	READIT	134	the input initial
THETA2	READIT	140	the input initial
THET1F	CHNTBL	150	the last value o
W1	READIT	133	the input initial
W2	READIT	139	the input initial
W1F	CHNTBL	170	the last value o
XCØM, 200	ZREADX or ZPRM	IXCØM	miscellaneous
ZTURN	READIT or SR2490	145	altitude at whic

The following integer quantities are also input:

ICØM, 200	ZREADX or ZPRM	IXCØM	integer option
IGDH, 20	ZREADX or ZPRM	IGDHL	input integer c
IGDL, 20	ZREADX or ZPRM	IGDHL	input integer c

in addition, any value in the ØCCUR array may also be input

ON refer

ion	Description	Units
	coefficients for calculating free space radar cross sections	-
	the input initial bluntness ratio for the first configuration	-
	the input initial bluntness ratio for the second configuration	-
	the value of bluntness ratio just before shape change	-
	the input initial vehicle axial length for first configuration	ft.
	the input initial vehicle axial length for second configuration	ft.
	the value of axial length just before shape change	ft.
		-
	the input initial value of base radius for first configuration	ft.
	the input initial value of base radius for second configuration	ft.
	the value of base radius just before shape change	ft.
	the input initial value of nose radius for the first configuration	ft.
	the input initial value of nose radius for second configuration	ft
	the value of nose radius just before shape change	ft.
	the input initial half cone angle for the first configuration	degrees
	the input initial half cone angle for second configuration	degrees
	the last value of cone half angle before shape change	degrees
	the input initial weight for the first configuration	lb.
	the input initial weight for the second configuration	lb.
	the last value of weight before shape change	lb.
	miscellaneous input quantities	-
	altitude at which vehicle shape change occurs	ft.
	integer option codes	-
	input integer code which specifies ØCCUR locations	-
	input integer code which specifies ØCCUR locations	-

B

3. Output

Name	Common Block	Description
ACØE	PCCUR(5881)	See Input
DELRCS	ØCCUR(3964)	delta of free space radar cross section
GD, 20	ØCCUR(3921 - 3940)	generalized differences

4. Numerical Procedure

As a first step, MISC tests the product of the seventh, eight and ninth elements of the IC∅M array. If this product equals zero, ANS is set equal to zero and statement 40 is executed next. If the product is not equal to zero, the six statements starting at 30 use the seventh, eighth and ninth elements of the IC∅M array to determine which three elements of the ∅CCUR array will be used as the first three arguments of P∅LCAL. Then P∅LCAL is called to calculate the free space radar cross section.

At 40, DELRCS is evaluated, then in the D∅ loop ending at 111 the generalized differences between specified pairs of elements in the ∅CCUR array are calculated and these values are saved in the GD array. The IGDH and IGDL arrays are input integers which specify locations in the ∅CCUR array. If an element of either of these arrays is zero, statement 112 is executed and no more differences are saved.

The four statements following statement 11 calculate the initial volume of the vehicle first configuration. Then RATI∅ (7) is defined as the ratio of the input initial weight of the first configuration to this volume.

Next ZTURN is tested to determine whether a shape change is expected. If not, control passes to 200. Otherwise, V2, the initial volume of the second configuration is calculated and RATI∅(8) is defined as the ratio of the initial input weight of the second configuration to this volume. The next six statements subtract the values of six vehicle characteristics just before shape change from the corresponding values of these characteristics after shape change and store these differences in the DIF array. The six statements following calculate the corresponding ratios if the value of RN1F or LAMD1F after shape change is not zero and save these in the RATI∅ array.

5. Other Information

A. MISC is called by FEV

B. MISC calls PØLCAL

SUBROUTINE PØLCAL(X, Y, Z, NAX, NAY, NAP, ITP, ACØE, ANS)

1. Purpose

SUBROUTINE PØLCAL evaluates the polynomial

$$ANS = \sum_{K=1}^{NAZ} \sum_{J=1}^{NAY} \sum_{I=1}^{NAX} \left[A_{((K-1)*NAY*NAX + (J-1)*NAX + I)} * X^{(I-1)} * Y^{(J-1)} * Z^{(K-1)} \right]$$

2. Input

*indicates integer quantity

Name	Source	Description
ACØE, 140	MISC	the polynomial coefficients , A's
ITP*	MISC	test parameter
NAX*	MISC	1 + order of X in polynomial
NAY*	MISC	1 + order of Y in polynomial
NAZ*	MISC	1 + order of Z in polynomial
X	MISC	first unknown of polynomial
Y	MISC	second unknown of polynomial
Z	MISC	third unknown of polynomial

3. Output

Name	Description
ANS	value of polynomial