2-106. SUBPROGRAM D15 (SQRØØT). SQRØØT evaluates the square root of a number greater than or equal to zero. The FORTRAN II reference statement is CALL SQRØØT (A,X).

a. Inputs. The input is the duplexed double-precision argument A. If A is actually single precision, the least significant registers of A contain zeros.

b. Outputs. The output is the argument X which is the duplexed double-precision square root of A.

c. Program Logic. FD D15

(1) Steps 1-4. The duplexed input is interrogated for a positive, negative, or zero value. If the input is positive, the subprogram continues at step 5. If the input is negative, SQRØØT exits to the error indication NEGSQR. If the input is zero, the output registers are set to zero and the subprogram performs the normal return procedure.

(2) Steps 5-11. The square root function on the FORTRAN II system library tape is used to obtain the first square root approximation X_0. Constants are set for use in expression (1) and the subprogram then evaluates the double-precision square root based on the first approximation. Expression (1) evaluates the square root by calling FDP, FAD and FMP to perform the double-precision divide, add, and multiply functions. The approximate square root is placed in the designated register. CUTIE is stepped by one and control is

returned to the user subprogram.

d. Expressions.

$$X = \frac{1}{2}\left(X_0 + \frac{A}{X_0}\right) \tag{1}$$

2-107. SUBPROGRAM P46 (SWAP). SWAP controls the time sharing of subprograms in core. The FORTRAN II reference statement is CALL SWAP.

a. Inputs. The inputs are the settings of the following switches:

SWITCH	ITEM (switch in ØN state)
SW(131)	Closed loop simulation is to be performed
SW(132)	Open loop simulation is to be performed
SW(151)	A constraint has been exceeded

b. Outputs. The outputs are the settings of the following switches depending on the status of core:

WWW.CHROMEHOOVES.NET

SWITCH	CONDITION	ITEM
SW(17)	ØN	B Common in core
	ØFF	C2 subprograms in core
SW(33)	ØN	B subprograms in core
	ØFF	C1 subprograms in core

c. Program Logic. FD P46

(1) Steps 1-14. INTRØG interrogates SW(131) to determine if CLØØP is requested. If ØFF, control is transferred to step 15. If ØN, U04 reads the C1 subprograms into core from tape A1. INTRØG interrogates SW(70) to determine if an error occurred in U04. If ØN, control is transferred to step 27. If ØFF, SW(33) is set ØFF to indicate that C1 subprograms are in core. U04 reads the C2 subprograms into

WWW.CHROMEHOOVES.NET

core from tape A1. INTRØG interrogates SW(70) to determine if an error occurred in U04. If ØN, control is transferred to step 27. If ØFF, SW(17) is set ØFF to indicate C2 subprograms are in core. Switches SW(153)-SW(157) are set ØFF to indicate that the subprograms for the ØTC, TØT, TAA, RSD, functions are not in core. Tape A1 is rewound. CLØØP performs closed loop missile flight simulation. SW(151) is examined to determine if a constraint has been exceeded (SW(151) = ØN). If ØN, control is transferred to step 25. Otherwise the subprogram continues at the next step.

(2) Steps 15-21. INTRØG interrogates SW(132) to determine if ØLØØP is to be called (SW(132) = ØN). If ØFF, control is transferred to step 25. If ØN, INTRØG interrogates SW(33) to determine if the B subprograms are in core (SW(33) = ØN). If ØFF, control is transferred to step 22. If ØN, U04 reads the C1 subprograms into core. INTRØG interrogates SW(70). If ØN, control is transferred to step 27. If ØFF, SW(33) is set ØFF to indicate that C1 subprograms are in core.

(3) Steps 22-24. SW(131) is set ØFF to indicate that CLØØP is to call ØLØØP. Tape A1 is rewound. ØLØØP performs open loop missile flight simulation.

(4) Step 25. BAREA verifies that the B subprograms and B Common are in core.

(5) Step 26. CUTIE is stepped by one and the subpro-

 Changed 31 May 1962

gram exits to the user subprogram.

(6) Step 27. The subprogram exits to BADPT for manual intervention.

2-108. SUBPROGRAM D11 (TANGNT). TANGNT computes the tangent of any angle expressed in radians. The FORTRAN II reference statement is CALL TANGNT (BA,DX).

a. Inputs. The input is the duplexed argument BA, the positive or negative angle expressed in radians whose tangent is to be evaluated. BA is expressed in single-precision floating point form of dimension two.

b. Outputs. The output is the duplexed argument DX which is the tangent BA expressed in single-precision floating point form of dimension two.

c. Program Logic. FD D11

(1) Steps 1-2. The input angle is tested. If BA $\geq$ 0.032, the subprogram continues at step 4; if BA $<$ 0.032, DX is set equal to $(BA)^3/3 + BA$ and dual computation is performed; if BA $<$ 0.0031 DX is set equal to input.

(2) Step 3. CUTIE is stepped by one, and control is returned to the user subprogram.

(3) Steps 4-6. SINE and CØSINE compute sin BA and cos BA. Cos BA is tested. If cos BA = 0, DX is set equal to maximum (FINIT,BA); if cos BA $\neq$ 0, DX is set equal to sin BA/ cos BA. The subprogram continues at step 3.

2-109. SUBPROGRAM D19 (VCDØTP). VCDØTP computes the acute angle between two input vectors and the dot product of the vectors. The FORTRAN II reference statement is CALL VCDØTP (BVECT QVECT).

a. Inputs. The inputs are the duplexed vectors BVECT and QVECT. Each of these vectors is specified by three components. BVECT and QVECT must be defined in both VECMAG and the user subprogram by Dimension statements as requiring six registers each.

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM
GVMG1	2	Magnitude of vector, $\|V1\|$
GVMG2	2	Magnitude of vector, $\|V2\|$
GCØSV	2	Cosine of angle between the two vectors
GNGLV	2	Angle in radians between the two vectors

WWW.CHROMEHOOVES.NET

c. Program Logic. FD D19

(1) Step 1. The dot product of the two vectors is computed.

$$V1 \cdot V2 = (V1_x)(V2_x) + (V1_y)(V2_y) + (V1_z)(V2_z)$$

WWW.CHROMEHOOVES.NET

(2) Steps 2-4. VECMAG computes the magnitude of each input vector in duplexed form. The cosine of the angle between the vectors is determined from the vector magnitudes and the dot product of the vectors.

$$\cos \lambda = \frac{V1 \cdot V2}{|V1| \quad |V2|}$$

ARCCØS computes the angle between the vectors.

(3) Step 5. CUTIE is stepped by one and control is returned to the user subprogram.

2-110. SUBPROGRAM D18 (VECMAG). VECMAG computes the magnitude of a vector expressed in components of a rectangular coordinate system. The FORTRAN II reference statement is CALL VECMAG (B, DA, DB, DC).

a. Inputs. The input is the duplexed argument B which is a vector comprised of X, Y, and Z components in a rectangular coordinate system. B is of dimension six.

b. Outputs. The major output is the duplexed argument DC, which represents the magnitude of the duplexed input vector in the same physical units as the input vector components. The other outputs are the duplexed parameters DB and DA. DB defines the square of DC. DA defines the square root of the sum of the squares of the X and Y components of vector B.

c. Program Logic. FD D18

(1) Steps 1-3. VECMAG sums the squares of the X and Y coordinates of vector B, using slightly different algorithms on the duplexed inputs to produce a duplexed result. SQRØØT obtains the square root of each sum of (X^2 plus Y^2) as the duplexed argument DA of the call statement for VECMAG.

(2) Steps 4-8. The duplexed argument DB is derived by adding each Z^2 to the corresponding duplexed sum of X^2 plus Y^2. SQRØØT obtains the duplexed argument DC as the square root of the argument DB. CUTIE is stepped by one and control is returned to the user subprogram.

2-111. SUBPROGRAM C55 (XPRØD). XPRØD computes the cross product of two 3-dimensional vectors. The FORTRAN II reference statement is CALL XPRØD (BA, BB, DC).

a. Inputs. The inputs are arguments BA and BB which are duplexed in double-precision floating point form. If BA and BB are actually single precision, their least significant registers contain zeros.

b. Outputs. The output is the argument DC of the above statement. DC is the duplexed double-precision cross product of the two 3-dimensional quantities.

c. Program Logic. FD C55

(1) Steps 1-6. FMP performs double-precision multiplication to obtain products in the following order:

$$A_y \ B_z$$
$$A_z \ B_y$$
$$A_z \ B_x$$
$$A_x \ B_z$$
$$A_x \ B_y$$
$$A_y \ B_x$$

where x, y, and z are coordinates of the input vectors BA and BB.

(2) Steps 7-9. FSB performs double-precision subtraction of the corresponding products:

$A_y B_z - A_z B_y$

$A_z B_x - A_x B_z$

$A_x B_y - A_y B_x$

(3) Step 10. CUTIE is stepped by one and control is returned to the user subprogram.

2-112. SUBPROGRAM D27 (XYZGEØ). XYZGEØ converts position vectors, from the inertial earth-centered rectangular coordinate system to geocentric latitude, longitude, and altitude form. The FORTRAN II reference statement is CALL XYZGEØ.

a. Inputs. The inputs are duplexed and expressed in single-precision floating point form. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
PSPPS	6	Missile position vector	$\vec{r}_m$	feet
GTMFL	2	Time of flight t_{fp} since missile launch to time of validity of missile position	$\vec{r}_m$	seconds
PRWLN	2	Longitude of guidance radar west of Greenwich	λ_R	degrees
GØMGA	2	Earth's rate of rotation	Ω	rad/sec

b. Outputs. The outputs are duplexed and in single-precision floating point form. The outputs for each of the duplexed sets of inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GMLAT	2	Geocentric latitude of point	L_{CP}	degrees
GMLØN	2	Longitude of point west of Greenwich	λ'_p	degrees
GMRAD	2	Radius of earth ellipsoid at this geocentric latitude	r_{ep}	feet

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GMDST	2	Radius vector magnitude of point relative to center of earth	r_p	feet
GMALT	2	Geometric altitude of point above earth ellipsoid	h'_p	feet

c. Program Logic. FD D27

(1) Steps 1-5. LØCALT obtains the missile parameters: radius of earth ellipsoid, radius vector magnitude, geometric altitude, sine of geocentric latitude, and projection of missile radius vector. The first three parameters are moved to registers specified as outputs of XYZGEØ. The projection of the missile radius vector is used in the development of Δ. If this value is zero, the registers designated to contain the result of expression (1) are set to zero, otherwise expression (1) is evaluated. In both cases, ARCSIN computes the arcsine of this quantity. ARCSIN also obtains the geocentric latitude from its sine which was obtained from LØCALT.

(2) Steps 6-7. The rotation of the earth since lift-off is computed as the product of time of flight since lift-off and the rate of rotation of the earth Ωt_{fk}. The quantities so far computed are converted from radians to degrees.

(3) Steps 8-20. The evaluation of Δ depends on the sign of the X and Y missile position coordinates.

SIGN		EXPRESSION USED
X	Y	
+	+	4
-	-	2
-	+	3

For conditions other than the above, Δ is set to the arcsine of the value set in registers designated to contain the results of expression (1). Expression (5) evaluates the missile longitude relative to the guidance radar. A dual computation is performed for these values.

(4) Steps 21-28. The longitude of the missile west of Greenwich is computed dependent upon the sign of the longitude relative to the radar site. Expression (5), (7), or (8) is used when longitude relative to the radar site is zero, positive, or negative, respectively. A dual computation is performed to obtain duplexed outputs for these values. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$\frac{|Y|}{\sqrt{X_m^{\ 2} + Y_m^{\ 2}}} \qquad (1)$$

$$\Delta = 180.0 - \text{arcsine} \frac{|Y|}{\sqrt{X_m^{\ 2} + Y_m^{\ 2}}} \qquad (2)$$

$$\Delta = 180.00 + \arcsin \frac{|Y|}{\sqrt{X_m^2 + Y_m^2}} \quad (3)$$

$$\Delta = 360.0 - \arcsin \frac{|Y|}{\sqrt{X_m^2 + Y_m^2}} \quad (4)$$

$$\Omega t_{fp} + \Delta + \lambda_R \quad (5)$$

$$|(\Omega t_{fp} + \Delta + \lambda_R)/360| \quad (6)$$

$$\Omega t_{fp} + \Delta + \lambda_R + 360 |(\Omega t_{fp} + \Delta + \lambda_R)/360| \quad (7)$$

$$\Omega t_{fp} + \Delta + X_R - 360 |(\Omega t_{fp} + \Delta + \lambda_R)/360| \quad (8)$$

2-113. B1 SUBPROGRAMS.

2-114. The Subprograms described in this area are mainly of a control nature or perform computations other than those performed during missile flight simulations. The subprograms are as follows:

a.	BØØSHL	P27	Booster Shell Nominal Impact Point
b.	CDCHK	U07	Check for All TGT, GGC, Inputs
c.	CDTYPE	U19	Determine Card Type
d.	CKPTCK	D40	Establish a Check Point
e.	CLØØP	**M02	Dummy CLØØP
f.	CNSTRN	P03	Check for Exceeded Constraints
g.	DØCNT	U11	Dynamic Operational Control
h.	ERRPRT	U13	Utility Subprograms Error Print
i.	FUSING	P19	Determine Fuzing Parameters
j.	GCØNST	P43	Convert and Tabulate GGC Data
k.	GIMBEX	D48	Print excessive Gimbal Angle
l.	GMGPHT	D37	Geopotential to Geometric Altitude Conversion
m.	HALT	D38	System Halt
n.	HERGET	P33	Herget Solution of Impact Point
o.	IDLIST	U38	List Record Identification on Tape
p.	INITAL	P35	Initialize for Flight Simulation
q.	LAZMTH	D21	Estimate Launch Azimuth

r.	METDTA	U39	Read, Modify, and Store MET Data
s.	MISDIS	P25	X,Y,Z Components of Miss Distance
t.	MSSNT	P78	Missile System Simulation Control
u.	SPRINT	S01	FORTRAN Library Print Subprogram
v.	TRAIL	P49	Target Aim Point Trail Adjustment Estimation
w.	TRGTRB	D16	Inertial Range and Target Bearing
x.	TRJPAR	P01	Delta Matrix Search
y.	U09	U09	Print Card Columns in Error
z.	U20	U20	Read One Card
a'.	U40	U40	Latitude Longitude Conversion
b'.	U41	U41	Decimal Information Conversion
c'.	WRTTP	U37	Write Binary Tape

** This subprogram has been written to form a linkage between the A and C1 areas and performs no function as a subprogram.

2-115. SUBPROGRAM P27 (BØØSHL). BØØSHL uses a Herget solution for booster shell impact point determination. The FORTRAN II reference statement is CALL BØØSHL.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	UNITS
TAPLN	2	Target aim point longitude west of Greenwich	degrees
TAPLT	2	Target aim point geocentric latitude.	degrees
GBTSM	2,15	Booster jettison point summary data table	

b. Outputs. The Outputs are as follows:

COMMON TAG	DIMENSION	ITEM	UNITS
FSPPS	2,3	Current single-precision missile position vector	feet
GCØPS	2,3	Starting point position vector for Herget computations	
GCØVL	2,3	Starting point velocity vector for Herget computations	
GCØTM	2	Starting point time of flight since liftoff for Herget computations	
GCRAD	2	Earth ellipsoid radius as a function of GCLAT	feet
GTAPR	2	Stopping point radius vector magnitude of Herget computations	

COMMON TAG	DIMENSION	ITEM	UNITS
SW(69)	1	If ØN, aim point is on same side of apogee as launch pad	

c. Program Logic.

(1) The values of target aim point longitude west of Greenwich and target aim point geocentric latitude are saved.

(2) The starting point position vector and the current single-precision missile position vector are set to their corresponding values from GBJSM table. The missile velocity vector components from GBJSM table are multiplied by 0.94, and set to the corresponding starting point velocity vector components.

(3) The starting point time of flight is duplexed and set to 119, and the stopping point radius vector magnitude is duplexed and set to the duplexed earth ellipsoid radius as a function of GCLAT.

(4) SW(69) is set ØFF to show that the aim point is not on the same side of apogee as launch pad.

(5) HERGET computes target aim point, altitude, and the geographic longitude and latitude.

(6) The values of target aim point longitude west of Greenwich and target aim point geocentric latitude are

restored.

(7) Control is returned to the user subprogram.

2-116. SUBPROGRAM U07 (CDCHK). CDCHK verifies that TGT cards or GGC cards for the function requested have been processed. The FORTRAN II reference statement is CALL CDCHK. The FAP reference instruction is TSX U07, 4.

a. Inputs. The input is SW(125) set ØN for TGT card input or SW(71) set ØN for GGC card input. If the function being processed is OTC, SW(75) is ØN; if TOT, SW(76) is ØN. UTGIT is the log of the TGT input cards and UGCIT is the log of the GGC input cards. IDPTG indicates the target slot being processed.

b. Outputs. This is a check program and there is no output as such. If an error occurs, a statement is printed and written indicating the card which is missing for either target, launch pad, or radar. The output statements are:

a. _____ TARGET _____ CARD MISSING

b. LAUNCH PAD _____ CARD MISSING

c. RADAR _____ CARD MISSING

c. Program Logic. FD U07

(1) Steps 1-10. The contents of index registers 1, 2, and 4 are saved. ITYER is set to zero, and work area CERR and SW(126) are set ØFF. For GGC card input, control is transferred to step 19. For TGT card input a check is made for the type of function to be performed. If TOT is required, the index registers are set to check for four UTGIT registers and 12 TGT cards. If OTC is required, the index

registers are set to check for three UTGIT registers and nine TOT cards. The subprogram continues at step 11 after the index registers are set. If the input is neither GGC nor TGT cards or if the function is neither TOT nor OTC, then CERR is set ØFF and control is transferred to step 24 for return to the user subprogram.

(2) Steps 11-18. If all of the target switches are ØFF, control is transferred to step 24. If one of the target switches is ØN, IDPTG is examined to determine if the target slot being processed corresponds to the target switch which is set ØN. If they do not agree, the target switches are examined to find the next target switch ØN. If they agree, the target number is stored in statement a. UTGIT is examined to determine if all the target data cards necessary for the particular target and function were processed. Missing card types, if any, are stored in statement a. An HBC card check is made although this card was deleted. The purpose is to provide proper indexing into the table. Work area CERR is set ØN. Statement a is printed and written. After examining UTGIT for all the necessary cards, control is transferred to step 24.

(3) Steps 19-28. If the function is for launcher GGC cards, a check is made for missing launcher cards, if any. If the function is not for launcher GGC cards, a check is made for missing radar cards, if any. If all cards are not present, control is transferred to step 22 for determination

of the missing launcher card(s) or to step 34 for determination of the missing radar card(s). If all cards are present, UGCIT is cleared (step 22) and the contents of the index registers are restored. The contents of CERR are stored in error indicator SW(70) which is interrogated. If ØFF, control is returned to the user subprogram. If ØN, ITYER and IFLAG are set, SW(126) is set ØN, and control is returned to the user subprogram.

(4) Steps 29-33. A check is made in successive passes for launcher cards LON, LAT, ALT, GSP, RAZ, and IDT. The BCI identification of each missing card is placed in statement c. CERR is set ØN and statement c is printed and written for each missing card. After checking for all cards, control is returned to step 22.

(5) Steps 34-37. A check is made in successive passes for radar cards IDT, LON, LAT, ALT, GSP, MC, FVC, MONA, MONB, MCWA, MCWB, RCNA, and RCNB. The BCI identification of each missing card is placed in statement b. This statement is printed and written for each missing card. After checking for all cards, control is returned to step 22.

2-117. SUBPROGRAM U19 (CDTYPE). CDTYPE determines the type of control card present and sets the corresponding switch ØN. The remaining control card switches are set ØFF. The FORTRAN II reference statement is CALL CDTYPE.

a. Inputs. The input is a control card in BCD format in the card image area CDIO-CDI13. The identification of the control card is in columns 8-10.

b. Outputs. The outputs are the switch for the corresponding card set ØN and all other control card switches set ØFF. The switches for the control cards are as follows:

SWITCH	CONTROL CARD
SW(74)	DOC - Dynamic Operation Control
SW(111)	DEC - Decimal Correction
SW(71)	GGC - Ground Guidance Complex
SW(114)	TGT - Target
SW(112)	MET - Meteorological
SW(118)	RSD - RSD Timing
SW(197)	IDT - Identification
SW(~~117~~ 105)	END - End
SW(116)	REM - Remarks
SW(~~117~~ 115)	TRA - Transfer
SW(119)	OCT - Octal Correction
SW(84)	RES - Restart

c. Program Logic. FD U19

(1) Steps 1-3. The contents of the index registers are saved. IFLAG is set to the identification integer 2119 and the error indicator switch SW(70) is set ØFF.

(2) Steps 4-39. The control card type (columns 8-10 of the input card) is compared against the subprogram stored control card types. Each control card switch is set ØFF until a match results, at which time the switch corresponding to that control card type is set ØN and the remaining switches are set ØFF. If no matching type results, the subprogram continues at step 40. The contents of the index registers are restored and the subprogram exits to the user subprogram.

(3) Steps 40-42. Columns 8-10 are indicated by setting to one the proper bits in CLEL, storing a four in ITYER, and setting SW(70) ØN. The contents of the index registers are restored and the subprogram exits to the user subprogram.

2-118. SUBPROGRAM D40 (CKPTCK). CKPTCK establishes a re-entry check point to which the program has been run successfully. All check points are established within the B area subprograms. The FORTRAN II reference statement is CALL CKPTCK (N).

a. Inputs. The inputs are the number of the check point (agreement N) and the contents of LAST, the number of the last successfully passed check point.

b. Outputs. The output is the updating of LAST and and DMPCT (Common table tape dump counter). OFTEN (the number of RLLBCK errors since last check point) and CUTIE (the successful computation unit tally) are set to zero. The Common Area is written on tape 5 and one or more of the following statements is printed and written:

(a) CHECK POINT ____ REACHED THROUGH IMPROPER SEQUENCING

(b) CHECK POINT ____ REACHED SUCCESSFULLY

(c) CKPTCK FAILED TO DUMP COMMON ON TAPE 5

c. Program Logic. FD D40

(1) Steps 1-4. The number N of this check point is compared with the previous check point in LAST. If N is greater than LAST, the subprogram continues at step 5. If N is less than LAST, output statement a is printed and written. IFLAG is set to identification integer 440 and

the subprogram exits to RLLBCK for return to the previous check point.

(2) Steps 5-14. LAST is set to N. CUTIE and ØFTEN are set to zero and DMPCT is stepped by one. IRECR and IFILE are set to N and five, respectively. U03 writes the Common Area on tape 5. INTRØG interrogates SW(70) to determine if an error occurred in U03. If ØN, the subprogram continues at step 15. If ØFF, INTRØG interrogates SW(123) to determine if a tape redundancy occurred. If ØN, (SW(123) = ØN) the subprogram continues at step 15. If ØFF, statement b is printed and written. The subprogram exits to the user subprogram.

(3) Steps 15-16. Statement c is printed and written and the subprogram exits to HALT for manual intervention.

2-119. SUBPROGRAM P03 (CNSTRN). CNSTRN tests constraint data against actual computed values, and prints a statement if the computed values are outside the limits of the constraints. Switches are used for testing the constraints with the computed list. The FORTRAN II reference statement is CALL CNSTRN.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	UNITS
FQVAX	2	Maximum value of q(FVA) product during re-entry	lb/ft-sec
GLQVR	2	Constraint - maximum value of (qV_a) between re-entry and detonation	lb/ft-sec
GRESM	2,9	Re-entry point summary data table	
GLXRA	2	Constraint - maximum re-entry angle above local horizontal	degrees
GLNRA	2	Constraint - minimum re-entry angle above local horizontal	degrees
GLXRS	2	Constraint - maximum re-entry air speed	ft/sec
GLNRS	2	Constraint - minimum re-entry air speed	ft/sec
TWDA	2	Current target desired detonation point above target	feet
GDPSM	2,9	Final detonation point	
GLMVL	2	Constraint - maximum impact velocity below 4500 ft altitude (=1500)	ft/sec
GELVE	2	Elevation angle nearest VECO	radians

COMMON TAG	DIMENSION	ITEM	UNITS
GNQVA	2	Actual value of integral of qV_a from liftoff to booster jettison	lb/ft
GLQ B	2	Constraint - maximum of integral of qV_a from liftoff to staging	lb/ft
GAQMX	2	Maximum value of q during flight prior to re-entry vehicle separation	lb/ft^2
GLQMX	2	Constraint - maximum value of dynamic pressure of q (=768)	lb/ft^2
GAQST	2	Value of Q at booster jettison point	lb/ft^2
GLQST	2	Constraint - maximum value of q during staging	lb/ft^2
SW(50)	1	If ØN, $\dot{A}$ exceeded maximum antenna slew rate during current flight	
SW(68)	1	If ØN, angle of attack has been excessive at least once during this flight	
SW(12)	1	If ØN, Stage II fuel or LOX exhaustion has occurred before vernier cutoff	
SW(134)	1	If ØN, look angle constraint exceeded	
SW(67)	1	If ØN, gimbal angle exceeded	
GMXQT	2	Time of flight since liftoff of occurrence of GAQMX	seconds
OLØØK	2	Sine of smallest too-small look angle occurring after angle has become at least +5°	pure no.
FRTØD	2	Conversion constant: Radians to degrees (180.0/π = 57.295780)	deg/rad

COMMON TAG	DIMENSION	ITEM	UNITS
GLØKT	2	Time of flight from lift-off to GLØØK	seconds
KØUNT	1	Counter used for TARGET	
FUFS	2,1	Usable Stage II fuel remaining	slugs
FULS	2,1	Usable Stage II LOX remaining	slugs
GXLPØ	2	Dynamic pressure at time of GXLPH	lb/ft^2
GXLPH	2	Largest excessive angle of attack so far	degrees
GXLPT	2	Time of flight since lift-off of occurrence of GXLPH	seconds
GLAZM	2	Launch azimuth	degrees
PAZLM	22	Azimuth limits data table	
FINIT	2,2	Largest positive floating point number expressible in memory	
FCGR	2	Conversion constant: mass in slugs to weight at sea level in pounds force (=32.174)	lbs/slug

WWW.CHROMEHOOVES.NET

b. Outputs. The outputs are: SW(151), set ØFF if no constraints were exceeded or set ØN if constraints were exceeded; and SW(19), set ØFF if there was no illegal launch azimuth or set ØN if there was illegal launch azimuth. The following written statements are also outputs:

a. RE-ENTRY HEATING CONSTRAINT WAS EXCEEDED.
QVA = ______ TIME = ____

b. RE-ENTRY ANGLE OUTSIDE LIMITS. ANGLE = ____

WWW.CHROMEHOOVES.NET