The following printed and written statements are also outputs:

WWW.CHROMEHOOVES.NET

a. NEW CARD TYPE ENCOUNTERED BEFORE AN END CARD-CANNOT PROCEED

b. 2ND ID CARD READ, NO DATA FOR PREVIOUS ONE-CANNOT PROCEED

c. UNEVEN NUMBER OF AZIMUTH LIMITS-IMPOSSIBLE TO PROCEED

d. AZIMUTH LIMITS CARD OUT OF ORDER-IMPOSSIBLE TO PROCEED

e. SECOND CARD WITH SAME M CONSTANT-IMPOSSIBLE TO PROCEED

f. LESS THAN 35 M CONSTANTS-IMPOSSIBLE TO PROCEED

g. ERROR READING TAPE A2 AFTER 10 TRIES

c. <u>Program Logic</u>. FD U10

WWW.CHROMEHOOVES.NET

(1) Steps 1-9. The contents of the index registers are saved, and SW(70) and SW(120) are set ØFF. If this is the first pass in reading the card, the subprogram is modified to indicate completion of the first pass. SW(82) is set ØFF so that DECNT will not print all input cards on line. The tape routine is initialized to read missile data. The subprogram continues at step 10. If this is the second pass and if the missile trajectory data was completed (step 4) control is transferred to step 128. If the missile data is incomplete, this data is placed in Common duplexed and written on tape B7.

(2) Steps 10-15. The first (next) card is read and the card image is written. If an error occurred either in

WWW.CHROMEHOOVES.NET

the reading or writing, control is transferred to step 87 for an error indication. If no error occurred, the card type is determined. Control is transferred to step 18 for a missile identification card, to step 36 for an END card, to step 38 for a TRA card, and to step 54 for any other card. The TRA card is for tape updating and replaces the respective GGC identification, data, and END card when the file is to remain unchanged.

(3) Steps 16-25. The card data is stored in duplex for update processing and blanks are stored in five identification words. If tape A2 does not require updating, or if tape A2 does require updating but an end-of-file occurred in step 19, then control is returned to DECNT for reading of DEC cards (step 23). After the cards are read, MTTAPE continues at step 1. If there was no end-of-file, a binary record is read from tape A2. A redundancy in the reading causes the printing and writing of statement g in step 24. ITYER is set to 8 and control is transferred to step 87 for an error indication. If there was no redundancy the subprogram continues.

(4) Steps 26-35. If an end-of-file was read, step 19 is modified to indicate this fact. Identification words are moved to the write area in step 30 and control is returned to DECNT (step 23). If there was no end-of-file, the record checksum is computed and compared with the tape checksum. If the checksums do not agree, ITYER is set to 1 in step 32, IFLAG is set to identification integer 2110 and the subprogram exits to ERRPRT for an error statement printout. If the

checksums agree and if the first card word agrees with the identification word on tape, control is transferred to step 30 for return to DECNT. If the words do not agree, the record is written from old tape A2 to new tape B7 and control is returned to step 19.

WWW.CHROMEHOOVES.NET

(5) Steps 36-43. If tape A2 does not require updating, or if tape A2 requires updating and an end-of-file was reached on this tape, control is transferred to step 44. If there was no end-of-file, a binary record is read from tape A2. A redundancy in the reading causes a transfer to step 24 for an error indication. If there was no redundancy and an end-of-file was reached, control is transferred to step 44. If there was no end-of-file, the record checksum is computed and compared with the tape checksum. Non-agreement transfers control to step 32 for an error indication. If the checksums agree, another binary record is read from tape A2 and the preceding procedure is repeated.

WWW.CHROMEHOOVES.NET

(6) Steps 44-54. Completion of missile data is indicated by modifying step 4. An end-of-file is written on tape B7 and the tape error routine is modified to read azimuth data. SW(71) and SW(199) are set ON; SW(72) is set OFF. The first (next) card is read and the card image is written. An error either in the reading or writing transfers control to step 87 for an error indication. If no error occurred and if all azimuth cards were processed, control is transferred to step 89. If all of the cards were not processed, the card type is de-

WWW.CHROMEHOOVES.NET

termined. Control is transferred to step 55 for an azimuth identification card, to step 73 for an END card, to step 80 for an azimuth data card, and to step 91 for a TRA card. Any other card results in printing and writing statement a, after which control is transferred to step 86 for an error indication.

(7) Steps 55-60. If the core contains a record to print, control is transferred to step 74. Otherwise, the card read in is examined. If this is the second azimuth identification (AZIDT) card (step 56), statement b is printed and written, and control is transferred to step 86 for an error indication. If this is the first AZIDT card, the subprogram is modified to indicate that the next AZIDT card is the second one on a repetition of the preceding procedure. The AZIDT words are stored in duplex for update processing and blanks are stored in four identification words.

(8) Steps 61-72. If tape A2 does not require updating or if an end-of-file was not reached (step 62) control is transferred to step 48 to read another card. If updating is required and an end-of-file was reached, a binary record is read from tape A2. A redundancy returns control to step 24 for an error indication. Otherwise, a check is made for end-of-file. If there was an end-of-file, step 62 is modified to indicate this fact. The azimuth data words are cleared in step 68 and the identification words are moved to the write area. Control is transferred to step 48. If there was no

WWW.CHROMEHOOVES.NET

end-of-file, the checksum of the block in core is computed and compared with the tape checksum. If the checksums do not agree, control is transferred to step 32 for an error indication. If the checksums agree and if the first identification word agrees with the tape, control is transferred to step 68. If the identification word does not agree, the record read from tape A2 is written on tape B7 and control is transferred to step 62.

(9) Steps 73-79. Step 52 is modified to indicate a new card type is required. An uneven number of azimuth limits causes statement c to be printed and written. Control is transferred to step 86. An even number sets the word following the last azimuth limit in the list to the largest positive floating number. The azimuth record is written on tape B7 and the azimuth limits are cleared from the block. Step 56 is restored to indicate only one identification card. Control is transferred to step 52.

(10) Steps 80-88. The azimuth data is converted from BCD to floating point binary. An error in the conversion transfers control to step 87 for an error indication. If no error occurred, the azimuth limits are tested for correct order. If in increasing order, the converted azimuth limits are stored and control is transferred to step 48 to read another card. If not in increasing order, statement d is printed and written, ITYER is set to 4, IFLAG is set to identification integer 2110, and the subprogram exits to

WWW.CHROMEHOOVES.NET

ERRPRT for an error statement printout (step 88).

WWW.CHROMEHOOVES.NET

(11) Steps 89-96. If tape A2 requires updating or if an end-of-file was reached on this tape, control is transferred to step 97 for a change in the tape error routine. If updating is not required and if an end-of-file was not reached, a binary azimuth data record is read from tape A2 in step 91. A redundancy in the reading causes a transfer to step 24 for an error indication. If there was no redundancy and an end-of-file was read, control is transferred to step 97. If an end-of-file was not read, the checksum of the block in core is computed and compared with the tape checksum. Non-agreement transfers control to step 32 for an error indication. If the checksums agree, the azimuth data record read from tape A2 is written on tape B7. Control is returned to step 91 to read another azimuth data record.

WWW.CHROMEHOOVES.NET

(12) Steps 97-106. The tape error routine is changed to read M constants and an end-of-file is written on tape B7. The first (next) card is read (step 99) and the card image is written. An error in either the reading or writing transfers control to step 87 for an error indication. If no error occurred, a check is made in step 103 to determine if an END card was read previously. If an END card was read but either tape updating is not required or an end-of-file was read, then control is transferred to step 125. If an END card was read but both tape updating is required and an end-of-file was not read, then control is transferred to step 135. If an END card

WWW.CHROMEHOOVES.NET

has not yet been read, the card type is determined. Control is transferred to step 107 for an END card, to step 113 for an M constant data card, to step 135 for a TRA card, to step 119 for an M constant identity card, and to step 54 for any other card.

(13) Steps 107-112. Step 103 is modified to indicate that an END card was read. If less than 35 M constants are stored, statement f is printed and written. Control is returned to step 86 for an error indication. Otherwise, the XM storage block is duplexed with M constants and an M constant record is written on tape B7. The XM block is cleared and step 120 is modified to indicate that only one identification card was read. Control is returned to step 103 to test for an END card.

(14) Steps 113-118. The M constant number is converted from BCD to binary. If there is a second card with the same M constant, statement e is printed and written, and control is returned to step 86 for an error indication. Otherwise the M constant data is converted from BCD to binary. An error in the conversion returns control to step 87 for an error indication. If no error occurred, the converted M constants are stored and control is returned to step 99 to read a card.

(15) Steps 119-134. If there is a record in core to print, control is returned to step 108 to check the number of M constants in storage. Otherwise a check is made to determine if an M constants identification (MCIDT) card was

WWW.CHROMEHOOVES.NET

read previously. If this is the second MCIDT card read, control is returned to step 57. If this is the first time an MCIDT card was read, the previous step is modified so that the next pass will indicate a second MCIDT card is being read. The MCIDT data is stored in duplex for processing and blanks are stored in five identification words. If tape A2 does not require updating or if an end-of-file was read (step 125), control is transferred to step 99 to read another card. If updating is required and if an end-of-file was not read, a record is read from tape A2. A redundancy in the reading returns control to step 24 for an error indication. If there was no redundancy, a check is made for end-of-file. If an end-of-file was read, step 125 is modified to indicate this fact. The XM block is cleared and control is returned to step 99. If there was no end-of-file, the record checksum is computed and compared with the tape checksum. Non-agreement returns control to step 32 for an error indication. If the checksums agree, the duplexed M constant identification words are compared. If equal, the XM block is cleared and control is returned to step 99. If not equal, the M constant record is written on tape B7 and control is returned to step 125 to check for end-of-file.

(16) Steps 135-149. A binary record is read from tape A2 for comparison. A redundancy in the reading returns control to step 24 for an error indication. If an end-of-file was not read, the checksum of the block in core is computed and compared with the tape checksum. Non-agreement causes

WWW.CHROMEHOOVES.NET

the M constant data record read from A2 to be written on tape B7. Control is returned to step 135. If an end-of-file was read, the tape error routine is changed to read a delta matrix record. An end-of-file is written after the M constant data on tape B7. The missile identification words are cleared from the delta matrix area. A delta matrix record is written on tape B7 in step 144. The next card is read and the card image is written. An error in either the reading or writing returns control to step 87 for an error indication. If no error occurred, the card type is determined. Control is transferred to step 150 for a delta matrix card, to step 162 for an END card, to step 164 for a TRA card, and to step 54 for any other card.

(17) Steps 150-161. The delta matrix identification cards are stored in duplex for update processing. If the tape does not require updating or if an end-of-file was reached (step 152) control is returned to step 21. If updating is required and if an end-of-file was not read, a record is read from tape A2. A redundancy in the reading returns control to step 24 for an error indication. If there was no redundancy, a check is made for end-of-file. If an end-of-file was read, step 152 is modified to indicate this fact. Control is transferred to step 161. If an end-of-file was not read, the record checksum is computed and compared with the tape checksum. Non-agreement returns control to step 32 for an error indication. If the checksums agree, the identification words are checked. If the first identi-

WWW.CHROMEHOOVES.NET

fication word does not agree with the card word, the record read from tape A2 is written on tape B7. Control is returned to step 152 to check for end-of-file. If the words agree, they are moved to the write area (step 161). Control is transferred to step 21 for return to DECNT.

(18) Steps 162-173. If tape A2 does not require updating or if an end-of-file was read, control is transferred to step 170. If updating is required and if an end-of-file was not read, a record is read from tape A2 (step 164). A redundancy in the reading returns control to step 24 for an error indication. If there was no redundancy, a check is made for end-of-file. If an end-of-file was not read, the record checksum is computed and compared with the tape checksum. Non-agreement transfers control to step 32 for an error indication. If the checksums agree, the delta matrix record read from tape A2 is written on tape B7. Control is returned to step 164 to read the next record. If an end-of-file was read, then an end-of-file is written on tape B7 and this tape is rewound. RTRN4 returns the subprogram along the path established by SAVE4. Return is made to the subprogram which originally called DECNT.

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

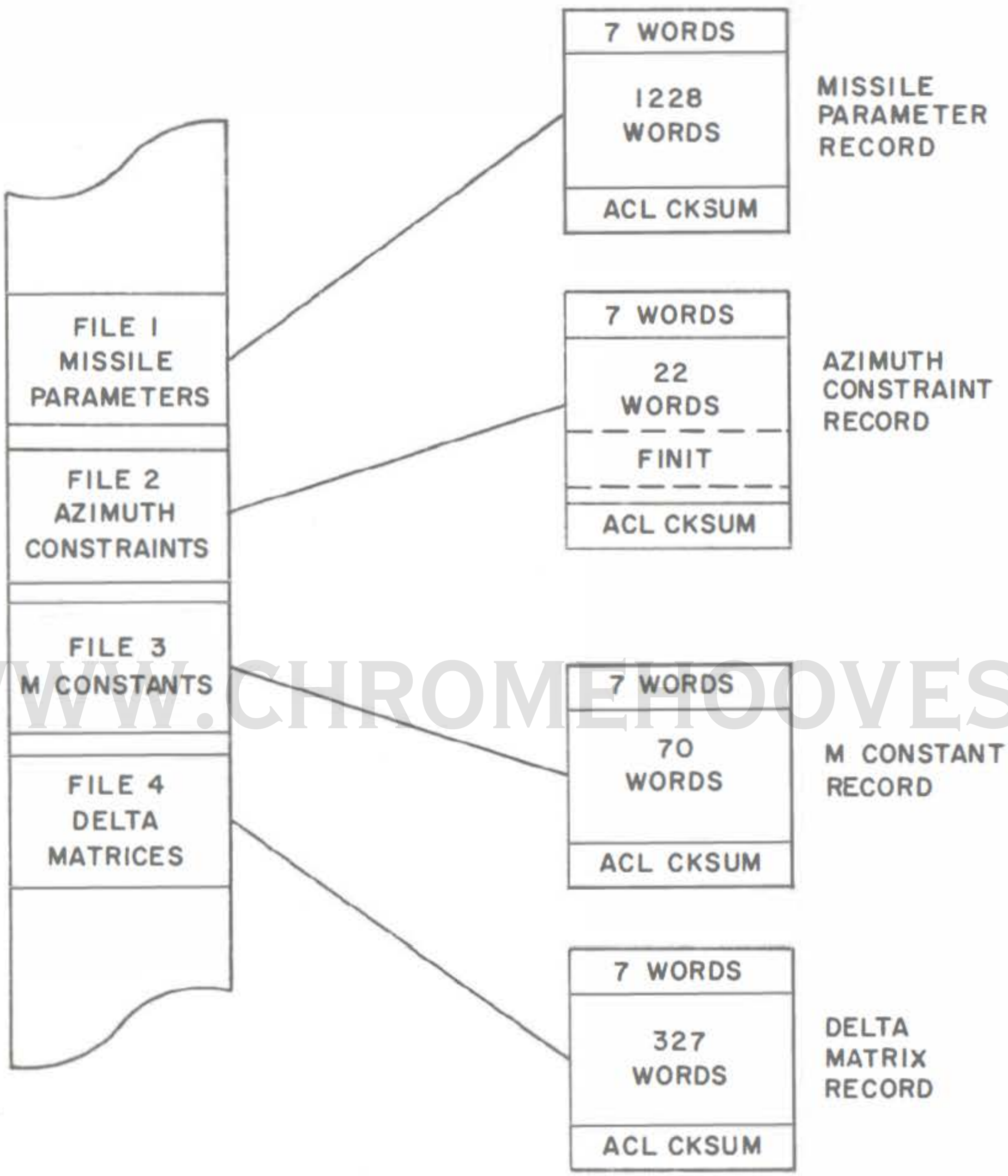

Figure 2-4A. M/T Tape Structure

2-193. SUBPROGRAM U48 (STØRE). STØRE stores the converted value into the Common register specified on the decimal correction card. The FORTRAN II reference statement is CALL STORE, I.

a. Inputs. The inputs are I, the location where the converted input value is stored, UDCAD, address from decimal correction card, and IACTG, the relative address in the block.

b. Output. The output is the value stored in the proper location.

c. Program Logic. The contents of index registers 1, 2, and 4 are saved. The subprogram identification integer 2148 is stored in IFLAG. The converted value is stored in the correct register in Common through address modification. The contents of the index registers are restored and the subprogram exits to the user subprogram.

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

2-194. B7 SUBPROGRAMS.

2-195. The subprograms described in this area are required for range safety data computations. The optional production of such range safety data is a separate function of MSS.

2-196. Subprogram L07 (BENTRY) enables the loader to establish linkage between the B1 and B7 subprograms. This version of BENTRY will be in core only when the RSD mode of operation is requested. The return path of the user subprogram is saved by SAVE4, and RSDNT is called. After RSD has been completed, the subprogram exits to DØCNT through RTRN4.

a.	BENTRY	*L07	Establish B7 Control Area
b.	IIP	P29	Compute and Write Instantaneous Impact Data (Category B)
c.	RSDA	P60	Compute and Write Category A Data
d.	RSDBAA	P05	Convert from Simplexed, Single Precision to Duplexed, Double Precision
e.	RSDE	P91	Split URSD Input Information into Five Distinct Groups for RSD Processing
f.	RSDIME	P38	Range Safety Data Output Timing Control
g.	RSDNT	P52	Control Range Safety Data Extraction
h.	RSDTIM	U29	Range Safety Data Time Control Card Interpretation
i.	URSD	P95	Read One Record from High Density Input Tape B3
j.	URSD1	P96	Control Output Format (Restoring Pages and Spacing)

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

k.	XYZRSD	D26	Convert Inertial Position and Velocity to Range Safety Coordinates

WWW.CHROMEHOOVES.NET

* Subprogram description is in the introductory paragraph of this area.

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

2-197. SUBPROGRAM P29 (IIP). IIP computes the following category B data and writes this data for delayed printout:

a. Geographic latitude and longitude of the instantaneous impact point

b. Down range distance to the impact point in nautical miles

c. Remaining flight time to impact in seconds

d. X_{RSD} and Y_{RSD} range safety coordinates of the impact point in feet and nautical miles

a. Inputs. The inputs are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(1)	Current category A printout time		seconds
XDEW(2)	Current category B printout time		seconds
XDEW(7)	X launch coordinate	X'_{LP}	feet
XDEW(8)	Y launch coordinate	Y'_{LP}	feet
XDEW(9)	Z launch coordinate	Z'_{LP}	feet
FRTØD	Conversion constant: radians to degrees		deg/rad
GMILE	Conversion constant: degrees to nautical miles		NM/deg
ZNØW(1)	Missile position vector	X_C	feet
ZNØW(2)	Missile position vector	Y_C	feet
ZNØW(3)	Missile position vector	Z_C	feet
ZNØW(4)	Missile velocity vector	$\dot{X}_C$	ft/sec
ZNØW(5)	Missile velocity vector	$\dot{Y}_C$	ft/sec
ZNØW(6)	Missile velocity vector	$\dot{Z}_C$	ft/sec

WWW.CHROMEHOOVES.NET

b. Outputs. The outputs are as follows:

WWW.CHROMEHOOVES.NET

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(160)	Down range distance to $\vec{R}_{IIP}$	R	naut mi
GGLAT(1)	Geographic latitude of $\vec{R}_{IIP}$	L_g	degrees
TAPLN(1)	Longitude west of Greenwich of $\vec{R}_{IIP}$	λ_{IP}	degrees
GTMB(1)	Remaining time of flight from cutoff to $\vec{R}_{IIP}$	t_b	seconds
XDEW(50)	X range safety coordinate	X_{RSD}	feet
XDEW(51)	Y range safety coordinate	Y_{RSD}	feet
SW(69)	ØFF - aim point is on side of apogee away from launch pad		

The following written statement also is an output.

WWW.CHROMEHOOVES.NET

IMPACT PØINT - LAT = _____ DEGS LØNG = _____ DEGS
DØWNRANGE DIST TØ IIP = _____ NAUT MILES
CATEGØRY B REM FLIGHT TIME TØ IIP = _____ SEC
IIP RSD CØØRDS X = _____ FEET Y = _____ FEET
CATEGØRY B

c. Program Logic. FD P29

(1) Steps 1-5. The subprogram return path is established by SAVE4 and IFLAG is set to identification integer 1629. Data is initialized for computation of the instantaneous impact point by Herget equations. The latest values of the starting point position vector (X_C, Y_C, Z_C) and starting point velocity vector ($\dot{X}_C$, $\dot{Y}_C$, $\dot{Z}_C$) are taken from the ZNØW registers. The stopping point radius vector magnitude R_H is

WWW.CHROMEHOOVES.NET

established as the distance from the earth's center to the launch pad r_e. The starting point time of flight since lift-off t_c is established as the category B printout time per target.

(2) Steps 6-10. SW(69) is set ØFF to establish the aim point on the side of apogee away from the launch pad. HERGET computes the instantaneous impact point in the inertial coordinate system, its geodetic latitude and longitude, and the remaining time of flight from cutoff to impact. If the printout time per target for category B data is greater than 60 seconds, the stopping point radius vector magnitude is reset to the magnitude of the earth ellipsoid radius and the HERGET solution is recomputed.

(3) Steps 11-15. The current time of flight since liftoff t_f is established as the category A printout time per target, which in turn, is reset to the total time of flight since liftoff at which a point position is valid t_{fp}. The real impact point position vector computed by HERGET (X_{IIP}, Y_{IIP}, Z_{IIP}) is loaded into the ZNØW registers and then converted by XYZRSD to range safety coordinates. The category A printout time per target is reset to the current time of flight since liftoff t_f.

(4) Steps 16-21. Vector computations are performed to determine the angle Ø between the real impact point position vector $\vec{R}_{IIP}$ and the launch pad position vector $\vec{R}'_{LP}$. Expression (1) evaluates the cosine of Ø from the vector magnitudes

and the dot product of the vectors. If the cosine of ϕ is less than 0.707, ARCCØS computes ϕ which represents the earth fixed angular range to the impact point. If the cosine of ϕ is equal to or greater than 0.707, expression (2) evaluates the sine of ϕ from the vector magnitude and the cross product of the vectors and ARCSIN computes ϕ. The down range distance to $\vec{R}_{IIP}$ is determined by converting ϕ from radians to nautical miles.

(5) Steps 22-24. The output statement is written. URSD1 counts four lines and ejects to a new page if necessary. RTRN4 returns the subprogram along the path established by SAVE4.

d. Expressions.

$$\cos \phi = \frac{\vec{R}_{IIP} \cdot \vec{R}'_{LP}}{\left| R_{IIP} \; R'_{LP} \right|} \quad (1)$$

$$\sin \phi = \frac{\vec{R}_{IIP} \times \vec{R}'_{LP}}{\left| R_{IIP} \; R'_{LP} \right|} \quad (2)$$

WWW.CHROMEHOOVES.NET

mod F2

2-198. SUBPROGRAM P60 (RSDA). RSDA computes category A data, and writes this data for delayed printout. The FORTRAN II reference statement is CALL RSDA.

a. Inputs. The inputs are as follows:

COMMON TAG	ITEM	SYMBOL	UNITS
SW(14)	Interpolation indicator--if ØN, bypass		
SW(82)	Test For Printing Option.		
XDEW(109)	X component of missile position vector	X	feet
XDEW(110)	Y component of missile position vector	Y	feet
XDEW(111)	Z component of missile position vector	Z	feet
XDEW(112)	X component of missile velocity vector	$\dot{X}$	ft/sec
XDEW(113)	Y component of missile velocity vector	$\dot{Y}$	ft/sec
XDEW(114)	Z component of missile velocity vector	$\dot{Z}$	ft/sec
XDEW(115)	Time to next normal interruption point		seconds
XDEW(116)	Stage of flight integer	LSEQ	
XDEW(117)	X component of acceleration vector due to gravity	g_x	ft/sec^2
XDEW(118)	Y component of acceleration vector due to gravity	g_y	ft/sec^2
XDEW(119)	Z component of acceleration vector due to gravity	g_z	ft/sec^2
XDEW(120)	Altitude above ellipsoid		feet
XDEW(121)	Time of flight since liftoff	t_f	seconds

WWW.CHROMEHOOVES.NET

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(122)	X component of thrust vector	TX	pounds
XDEW(123)	Y component of thrust vector	TY	pounds
XDEW(124)	Z component of thrust vector	TZ	pounds
XDEW(125)	Missile mass	m	slugs
XDEW(126)	X component of roll vector	ξ_X	
XDEW(127)	Y component of roll vector	ξ_Y	
XDEW(128)	Z component of roll vector	ξ_Z	
XDEW(129)	Sine of gimbal angle		
XDEW(130)	Usable fuel - booster		slugs
XDEW(131)	Usable LOX - booster		slugs
XDEW(132)	Usable fuel - stage II		slugs
FCGR	Conversion constant (=32.174)		ft/sec^2
XDEW(133)	Usable LOX - stage II		slugs
XDEW(134)	Interrupt code number XI		
XDEW(135)	Group ID 3.0		
XDEW(136)	X component of wind velocity vector at missile		ft/sec
XDEW(137)	Y component of wind velocity vector at missile		ft/sec
XDEW(138)	Z component of wind velocity vector at missile		ft/sec
XDEW(139)	X component of drag vector	D_x	pounds
XDEW(140)	Y component of drag vector	D_y	pounds
XDEW(141)	Z component of drag vector	D_z	pounds
XDEW(1)	Time of occurrence of category A printouts	ACURR	seconds
XDEW(7)	X component of launch pad vector	X'_{LP}	feet

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

Pages 2-517 through 2-518 are missing

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(759) to XDEW(791)	Table of interpolated values		
XDEW(29)	Altitude of missile	h_ξ	feet

WWW.CHROMEHOOVES.NET

The following written statements also are outputs:

a. TWTDI = ____ TWDZ = ____ LAUN LAT = _____ DEGS LONG = _____ DEG TARGET-LAT = _____ DEGS LON = _____ DEG.

b. TIME FROM LIFTOFF = _____ SECS/RANGE SAFETY POSITION X = _____ FEET Y = _____ FEET Z _____ FEET CATEGORY A/RANGE SAFETY VELOCITY X = _____ FT/SEC Y = _____ FT/SEC Z = _____ FT/SEC CATEGORY A/SPEED = _____ FT/SEC ANGLE FROM VERT = _____ DEG ALTITUDE _____ FT TOTAL WEIGHT = _____ LBS CATEGORY A/LONG ACCL = _____ FT/SEC - SEC FUEL WT = _____ LBS LOX WT = _____ LBS PROJECTED RANGE = _____ NAUT MILES CATEGORY A///.

WWW.CHROMEHOOVES.NET

c. Program Logic. FD P60

(1) Steps 1-5. IFLAG is set to identification integer 1660. INTRØG interrogates SW(14) to determine if interpolation is to be bypassed. If ØN, ZNØW registers are set up from the XDEW registers for the past integration interval, SW(14) is set ØFF, and the subprogram continues at step 10. If ØFF, the subprogram continues at step 6.

(2) Steps 6-9. The first (next) values of the present, past, and past-past integration intervals are initialized for interpolations. INTERP interpolates the values in the table

WWW.CHROMEHOOVES.NET

as a function of the time ACURR. If all values have not been interpolated, the subprogram continues at step 6. Otherwise the ZNØW registers are set to the interpolated values and the subprogram continues at step 10.

(3) Steps 10-19. XYZRSD converts the position and velocity components from the inertial system, (X, Y, Z) and ($\dot{X}$, $\dot{Y}$, $\dot{Z}$), to the range safety system (X_{RSD}, Y_{RSD}, Z_{RSD}) and ($\dot{X}_{RSD}$, $\dot{Y}_{RSD}$, $\dot{Z}_{RSD}$). Missile speed is computed using expression (1). The Y component of the velocity vector is tested. If negative, the sign of the Y component is changed and the cosine of the re-entry angle computed; if equal to zero, the re-entry angle is set equal to 90 degrees and the subprogram continues at step 19; if positive, the cosine of the re-entry angle is computed and the subprogram continues at step 16. The re-entry angle is computed using ARCCOS and then tested. If negative, 90 degrees is added to the angle prior to setting up the altitude; otherwise, the subprogram goes directly to the altitude computation where missile altitude is set to the altitude above the ellipsoid.

(4) Steps 20-34. The angle θ between the launch pad vector and the inertial missile vector is computed. RSDBAA and SQRØØT compute the magnitude of the missile position vector by use of expression (2). ARCSIN computes the geocentric latitude L_M of the missile by use of expression (3). The inputs to ELLRAD are initialized and ELLRAD computes the earth ellipsoid radius R_{LM} as a function of the geocentric latitude L_M. ELLRAD is initialized with the geocentric lat-

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

itude L_L of the launch pad and then computes the earth ellipsoid radius R_{LL} as a function of the geocentric latitude of the launch pad. Expression (4) evaluates the range in nautical miles to a point on the ellipsoid below the missile from the range safety origin. If re-entry vehicle separation has occurred, λ is set to zero; otherwise, λ is set to one. If the altitude of the missile is less than or equal to 300,000 feet, γ is set to one; otherwise, γ is set to zero. The longitudinal acceleration is computed by use of expression (5), and RSDBAA and SQRØØT compute the total missile weight due to force of gravity by use of expression (6). The subprogram continues at step 35.

(5) Steps 35-51. If this is the booster stage, α is set to one; otherwise, α is set to zero. If re-entry vehicle separation has occurred, β is set to zero; otherwise, β is set to one. Expressions (7) and (8) evaluate the total fuel and LOX. The input to URSD1 is initialized to eight and URSD1 sets up the output format to restore the page if necessary. Statements a and b are written. If SW(82) is ØN after the particular statement (a or b) is written, that statement is printed.

WWW.CHROMEHOOVES.NET

d. Expressions.

$$V = \sqrt{\dot{X}_{RSD}^2 + \dot{Y}_{RSD}^2 + \dot{Z}_{RSD}^2} \quad (1)$$

where

$\dot{X}_{RSD}$, $\dot{Y}_{RSD}$, $\dot{Z}_{RSD}$ are range safety coordinates

WWW.CHROMEHOOVES.NET

$$R_M = \sqrt{X^2 + Y^2 + Z^2} \tag{2}$$

WWW.CHROMEHOOVES.NET

$$L_M = \frac{Z}{\sqrt{X^2 + Y^2 + Z^2}} \tag{3}$$

$$R = (K\theta/2)\ (R_{LM} + R_{LL}) \tag{4}$$

where

θ is in radians

$$\begin{aligned} A = [1/m] \Big[& \xi_x\ (mg_x + \lambda TX + \gamma\ D_x) \\ & + \xi_y\ (mg_y + \lambda TY + \gamma\ D_x) \\ & + \xi_z\ (mg_z + \lambda TZ + \gamma\ D_z) \Big] \end{aligned} \tag{5}$$

$$MG = m \sqrt{{g_x}^2 + {g_y}^2 + {g_z}^2} \tag{6}$$

$$\text{Total Fuel} = (\alpha F_{TB} + \beta F_{TS})\ 32.174 \tag{7}$$

WWW.CHROMEHOOVES.NET

$$\text{Total LOX} = (\alpha L_{TB} + \beta L_{TS})\ 32.174 \tag{8}$$

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

2-201. SUBPROGRAM P05 (RSDBAA). RSDBAA converts simplexed, single-precision quantities to duplexed, double-precision quantities. The FORTRAN II reference statement is CALL RSDBAA.

a. Inputs. The input is the Common storage variable BAA(1) which contains a simplexed, single-precision quantity.

b. Outputs. The outputs are the Common storage variables BAA(2)-BAA(4) which contain duplexed, double-precision information.

c. Program Logic. BAA(2) is set equal to BAA(1) forming a duplexed quantity. BAA(3) and BAA(4) are set to zero. The subprogram returns to the user subprogram.

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

2-205. SUBPROGRAM P91 (RSDE). RSDE splits input information from the array GRASE into five distinct groups for RSD processing. The FORTRAN II reference statement is CALL RSDE.

a. Inputs. The inputs are the values in GRASE.

b. Outputs. The outputs are XDEW(101)-XDEW(157) in five groups, and the setting of IFLAG. These registers are initialized for RSD computations.

c. Program Logic. FD P91

(1) Steps 1-5. IFLAG is set to identification integer 1691 and the internal register NDEW is initialized to one. If the first ID number is one, XDEW(101)-XDEW(107) are initialized by GRASE(1)-GRASE(7) as the first group and NDEW is stepped up by seven. The subprogram continues at step 6.

(2) Steps 6-8. If the first (next) ID number is two, XDEW(108)-XDEW(134) are initialized by GRASE(NDEW)-GRASE (NDEW + 26) as the second group and NDEW is stepped up by 27. The subprogram continues at step 9.

(3) Steps 9-11. If the first (next) ID number is three, XDEW(135)-XDEW(145) are initialized by GRASE(NDEW)-GRASE(NDEW + 10) as the third group, and NDEW is stepped up by 11. The subprogram continues at step 12.

(4) Steps 12-14. If the first (next) ID number is four, XDEW(146)-XDEW(154) are initialized by GRASE(NDEW)-

WWW.CHROMEHOOVES.NET

GRASE(NDEW + 8) as the fourth group, and NDEW is stepped up by nine. The subprogram continues at step 15.

(5) Steps 15-16. If the first (next) ID number is five, XDEW(155)-XDEW(157) are initialized as the fifth group. The subprogram returns to the user subprogram.

WWW.CHROMEHOOVES.NET
WWW.CHROMEHOOVES.NET
WWW.CHROMEHOOVES.NET

2-206. SUBPROGRAM P38 (RSDIME). RSDIME controls the timing of Range Safety Data computations and the corresponding printouts. After certain conditions are set up for a particular target by RSDNT, control is transferred to RSDIME. RSDIME then causes the outputs of categories A and B to be interpolated independently, in accordance with the RSD timing card. The FORTRAN II reference statement is CALL RSDIME.

a. Inputs. The inputs are divided into two types:

(1) RSDNT sets up the following inputs for an entire target run:

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(1)	Current category A printout time ACURR		seconds
XDEW(2)	Current category B printout time BCURR		seconds
XDEW(3)	Time to stop category A printouts ASTØP		seconds
XDEW(4)	Time to stop category B printouts BSTØP		seconds
XDEW(5)	Time difference between successive category A printout time DLTA	ΔtA	seconds
XDEW(6)	Time difference between successive category B printout time DLTB	Δtb	seconds

(2) The following inputs are stored on tape B3 in target files. Each target file consists of a series of records whose contents depend upon the phase of flight. Data in any record is made up from the following groups depending upon the flight phase:

WWW.CHROMEHOOVES.NET

(a) Heading on first record only:

WWW.CHROMEHOOVES.NET

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(101)	Group identification of tape		
XDEW(102)	Target data inventory number TWTDI		BCD
XDEW(103)	Desired ground zero TWDGZ		BCD
XDEW(104)	Geographic latitude of target TWGLT		degrees
XDEW(105)	Longitude of target west of GREENWICH TWLN		degrees
XDEW(106)	Altitude of target above geoid TWAL		feet
XDEW(107)	Launch azimuth	A_L	

(b) All records following the first:

WWW.CHROMEHOOVES.NET

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(109)	X component of missile position vector	X	feet
XDEW(110)	Y component of missile position vector	Y	feet
XDEW(111)	Z component of missile position vector	Z	feet
XDEW(112)	X component of missile velocity vector	$\dot{X}$	ft/sec
XDEW(113)	Y component of missile velocity vector	$\dot{Y}$	ft/sec

WWW.CHROMEHOOVES.NET

WWW.CHROMEHOOVES.NET

COMMON TAG	ITEM	SYMBOL	UNITS
XDEW(114)	Z component of missile velocity vector	$\dot{Z}$	ft/sec
XDEW(115)	Time to next normal interrupt point		seconds
XDEW(116)	Stage of flight integer LSEQ		
XDEW(117)	X component of acceleration vector due to gravity	g_x	ft/sec^2
XDEW(118)	Y component of acceleration vector due to gravity	g_y	ft/sec^2
XDEW(119)	Z component of acceleration vector due to gravity	g_z	ft/sec^2
XDEW(120)	Altitude above ellipsoid		feet
XDEW(121)	Time of flight since liftoff		seconds
XDEW(122)	X component of thrust vector	TX	lbs
XDEW(123)	Y component of thrust vector	TY	lbs
XDEW(124)	Z component of thrust vector	TZ	lbs
XDEW(125)	Current missile mass	m	slugs
XDEW(126)	X component of roll vector	ξ_x	
XDEW(127)	Y component of roll vector	ξ_y	
XDEW(128)	Z component of roll vector	ξ_z	
XDEW(129)	Sine of gimbal angle		
XDEW(130)	Usable fuel (booster)		slugs
XDEW(131)	Usable LOX (booster)		slugs
XDEW(132)	Usable fuel (stage II)		slugs
XDEW(133)	Usable LOX (stage II)		slugs
XDEW(134)	Interrupt code numbers: if one, a simulation interval; if two, an interrupt point		

WWW.CHROMEHOOVES.NET

(c) Present when altitude $\leq$ 300,000 feet:

WWW.CHROMEHOOVES.NET