c. RE-ENTRY SPEED OUTSIDE LIMITS. SPEED = _____

d. IMPACT VELOCITY IS TOO LARGE. SPEED = _____

e. HEATING CONSTRAINT FROM LIFTOFF TO BOOSTER JETTISON WAS EXCEEDED. QVA = _____

f. EXCESSIVE DYNAMIC PRESSURE DURING POWERED FLIGHT. Q = _____ TIME = _____

g. EXCESSIVE DYNAMIC PRESSURE AT BOOSTER STAGING. Q = _____

h. ELEVATION ANGLE CONSTRAINT WAS EXCEEDED. ANGLE IN DEGREES = _____ TIME = _____

i. LOOK ANGLE CONSTRAINT WAS EXCEEDED. ANGLE IN DEGREES = _____ TIME = _____

j. STAGE II FUEL EXHAUSTION HAS OCCURRED BEFORE VERNIER CUTOFF. FUEL - LBS = _____ LOX - LBS = _____

k. EXCESSIVE Q-ALPHA PRODUCT. Q = ____ ALPHA = _____ TIME = _____

l. EXCESSIVE RADAR SLEW RATE. A DOT = _____ TIME = _____

m. ILLEGAL LAUNCH AZIMUTH. ANGLE IN DEGREES = _____

c. Program Logic. FD P03

(1) Steps 1-5. SW(151) and SW(19) are set ØFF. KØUNT is tested. If less than zero the subprogram continues at step 47, if equal to zero the subprogram continues at step 44, if greater than zero and the maximum value of (qV_a) product during re-entry is less than or equal to the maximum value of (qV_a) between re-entry and detonation, the subprogram continues at step 6. Otherwise statement a is written.

(2) Steps 6-8. If the absolute value of the re-entry point is less than or equal to the maximum re-entry angle above

the local horizontal, and is greater than or equal to the minimum re-entry angle above the local horizontal, the subprogram continues at step 9. Otherwise statement b is written.

(3) Steps 9-11. If the re-entry point speed data is less than or equal to the maximum re-entry air speed, and is equal to or greater than the minimum re-entry air speed, the subprogram continues at step 12. Otherwise statement c is written.

(4) Steps 12-14. If the current detonation point is not higher than 4500 feet and if the impact velocity data at final detonation point is greater than the maximum impact velocity, statement d is written. Otherwise the subprogram continues at the next step.

(5) Steps 15-18. IFLAG is set to identification integer 1603. INTRØG interrogates SW(67) to determine if the gimbal angle has been exceeded. If ØFF, the gimbal angle has not been exceeded and the subprogram continues at step 19. If ON, GIMBEX prints a notification if the largest excessive gimbal angle has been encountered.

(6) Steps 19-20. If the value of the integral of qV_a from liftoff to booster jettison is less than or equal to the maximum of the integral of qV_a from liftoff to staging, the subprogram continues at step 21. Otherwise statement e is written.

(7) Steps 21-22. If the maximum value of q prior to

re-entry vehicle separation is less than or equal to the maximum value of the dynamic pressure q, the subprogram continues at step 23. Otherwise statement f is written.

(8) Steps 23-24. If the value of q at booster jettison is less than or equal to the maximum value of q during staging, the subprogram continues at step 25. Otherwise statement g is written.

(9) Steps 25-26. If the elevation angle nearest VECO is greater than 0.19198622, the subprogram continues at step 27. Otherwise statement h is written.

(10) Steps 27-30. INTRØG interrogates SW(134) to determine if the look angle constraint has been exceeded. If ØFF, the subprogram continues at step 31. If ØN, ARCCØS computes the inverse cosine of the look angle, and statement i is written.

(11) Steps 31-36. INTRØG interrogates SW(12) to determine if Stage II fuel or LOX exhaustion has occurred before vernier cutoff. If ØFF, the subprogram continues at step 38. Otherwise statement j is written. KØUNT is tested. If greater than one or if the usable stage II fuel and LOX remaining is less than or equal to zero, the subprogram continues at step 37. Otherwise the subprogram continues at step 38.

(12) Step 37. SW(151) is set ØN to show that a constraint has been exceeded.

(13) Steps 38-40. INTRØG interrogates SW(68) to determine if the angle of attack has been exceeded. If ØFF, the subprogram continues at step 41. Otherwise statement k is written.

(14) Steps 41-43. INTRØG interrogates SW(50) to determine if the maximum antenna slew rate was exceeded during the current flight. If ØFF, the subprogram continues at step 44. If ØN, statement l is written.

(15) Steps 44-47. The handover launch azimuth is computed modulo 360.0. The azimuth limit is tested against the largest floating point number. If equal to the largest floating point number, control is returned to the user subprogram. If greater than the largest floating point number, the subprogram continues at step 48. If less than the largest floating point number, the launch azimuth is tested against this azimuth lower limit. If equal, the subprogram continues at step 50; if less, control is returned to the user subprogram, if greater, the launch azimuth is tested against this azimuth upper limit. If less than, or equal, the subprogram continues at step 50, if greater, the subprogram continues at step 48.

(16) Steps 48-49. IFLAG is set to identification integer 1603. The subprogram exits to RLLBCK for return to previous check point.

(17) Steps 50-54. SW(19) is set ØN to show that legal launch azimuth has been exceeded. Statement m is written.

The amount by which launch azimuth was exceeded is computed. KØUNT is tested. If KØUNT is less than or equal to one and launch azimuth was not exceeded by more than one, control is returned to the user subprogram. Otherwise the subprogram continues at the next step.

(18) Step 55. SW(151) is set ØN to show that a constraint has been exceeded and the subprogram returns to the user subprogram.

2-120. SUBPROGRAM U11 (DØCNT). DØCNT is the major control subprogram of the TITAN Targeting Program. DØCNT performs initialization of parameters by causing dynamic inputs to be read and stored. After initialization has been completed, DØCNT transfers control to the selected functional control subprogram. When the selected function has been completed, DØCNT terminates the processing of the TTP. The FORTRAN II reference statement is CALL DØCNT.

a. Inputs. This is a control subprogram and there are no inputs as such. SENSE switch 3 is OFF if the TTP is to start at the beginning, or ON if the TTP is to start as determined from the restart card. SW(151) will be ØN if a constraint was exceeded during east simulation of target.

b. Outputs. The outputs are the following statements which are always printed and written:

a. TITAN TARGETING PROGRAM MODEL _____ NOW OPERATING
b. START R/L TAPE UPDATE
c. START R/L TAPE PRODUCTION
d. COMPLETED R/L TAPE OPERATION
e. BEGIN TAPE DATA RECORD ID LISTING
f. TAPE DATA RECORD ID LISTING COMPLETED
g. START M/T TAPE UPDATE
h. START M/T TAPE PRODUCTION
i. COMPLETED M/T TAPE OPERATION
j. START COMMON INPUT TAPE GENERATION
k. COMMON INPUT TAPE COMPLETED

l. STARTING OTC MODE OF OPERATION

m. ALL TARGETS DUMMIES - TARGETING KIT NOT GENERATED

n. TARGETING KIT GENERATED HALT FOR OFFLINE OPERATION

o. COMPLETED OTC MODE OF OPERATION

p. STARTING TOT MODE OF OPERATION

q. COMPLETED TOT MODE OF OPERATION

r. STARTING TAA MODE OF OPERATION

s. COMPLETED TAA MODE OF OPERATION

t. STARTING SIM MODE OF OPERATION

u. COMPLETED SIM MODE OF OPERATION

v. STARTING RSD MODE OF OPERATION

w. COMPLETED RSD MODE OF OPERATION

x. ILLEGAL CONTROL CARD

y. ILLEGAL NUMBER OF IDT CARDS USED

c. Program Logic. FD U11

(1) Steps 1-6. If a constraint was exceeded during last simulation (SW(151) = ØN), control is transferred to step 111, otherwise statement a is written. PRTSSW prints the SENSE switch settings. SW(126) is set ØFF. SENSE switch 3 is tested to determine if this is a restart operation. If ØN, step 18 is modified to continue at step 19, if OFF, step 18 is modified to continue at step 28 and J is set to one.

(2) Steps 7-9. U20 reads a card. INTRØG interrogates the error indicator SW(70) to determine if an error has occurred in U20, (SW(70) = ØN). If ØN, the subprogram con-

tinues at the next step; if ØFF, control is transferred to step 11.

(3) Step 10. ERRPRT prints an indication of the error and operation is halted. If the error is corrected at this point in the subprogram operation, and operation is resumed, control is transferred to step 6.

(4) Steps 11-17. SW(120) is set ØFF. The card, in BCD format in the card image area CDIO-CDI11, is written by U08. INTRØG interrogates SW(70) to determine if an error occurred in U08. If ØN, control is transferred to step 10; if ØFF, CDTYPE determines the type of card. INTRØG interrogates SW(70) to determine if an error occurred in CDTYPE. If ØN, control is transferred to step 10; if ØFF, the subprogram continues at the next step.

(5) Step 18. This step transfers control to step 28, unless the branch has been modified at step 6 to transfer control to step 19.

(6) Steps 20-27. RESCD interprets the restart card. INTRØG interrogates SW(70) to determine if an error occurred in RESCD. If ØN, control is transferred to step 10. Otherwise U04 reads the RLLBCK tape on tape A5. INTRØG interrogates SW(70) to determine if an error occurred in U04. If ON, control is transferred to step 10. If OFF, SW(153)-SW(157) and SW(130) are set ØFF. The control number is written. The B subprograms are verified to be in core by

BAREA. The subprogram exits to RSET4 to transfer to the starting address.

(7) Steps 28-36. INTRØG interrogates SW(116) to determine if this is a remarks card. If ØN, control is transferred to step 7. If ØFF, INTRØG interrogates SW(197) to determine if this is an IDT card. If ØFF, control is transferred to step 37; if ØN, IDTTYP interprets the IDT card. INTRØG interrogates SW(70) to determine if an error occurred in IDTTYP. If ON, control is transferred to step 10. If ØFF, J is stepped by one and then tested to determine if the correct number of IDT cards has been read. If J equals 12, control is transferred to step 82. Otherwise control is transferred to step 7.

(8) Steps 37-54. INTRØG interrogates SW(117) to determine if this is an END card. If ØN, control is transferred to step 126. Otherwise INTRØG interrogates SW(74) to determine if the input is a DOC card. If ØFF, control is transferred to step 173; if ØN, DØCTYP determines the DOC control card type. INTRØG interrogates SW(70) to determine if an error occurred in DØCTYP. If ØN, control is transferred to step 10. If ØFF, INTRØG interrogates SW(18) to determine if the R/L tape is to be processed (SW(18) = ØN). If ØFF, control is transferred to step 62. Otherwise INTRØG interrogates SW(184) to determine if the tape is to be listed only (SW(184) = ØN). If ØN, control is transferred to step 67. Otherwise INTRØG interrogates SW(183). If the tape is

to be updated (SW(183) = ØN), statement b is written; if the tape is to be generated (SW(183) = ØFF), statement c is written. SW(174) is set ØFF. BAREA verifies that the B subprograms for R/L processing are in core. BENTRY transfers control to process RLTAPE. Step 58 is modified to continue at step 59.

(9) Steps 55-57. IDLIST lists the tape record identification. INTRØG interrogates SW(70) to determine if an error occurred in IDLIST. If ØN, control is transferred to step 10. Otherwise the subprogram continues at the next step.

(10) Step 58. This step is modified to continue at step 59, 69, 77, or 87.

(11) Step 59. Statement d is written.

(12) Step 60. IFLAG is set to identification integer 32767.

(13) Step 61. The subprogram exits to HALT for manual intervention.

(14) Steps 62-66. INTRØG interrogates SW(171) to determine if the M/T tape is to be processed (SW(171) = ØN). If ØFF, control is transferred to step 78; otherwise SW(189) is set ØN. INTRØG interrogates SW(184) to determine if tape is to be listed only (SW(184) = ØN). If ØN, the subprogram continues at the next step; otherwise control is transferred to step 70.

(15) Steps 67-68. Statement e is written and step 58 is modified to continue at step 59. Control is transferred to step 55.

(16) Step 69. Statement f is written and control is transferred to step 60.

(17) Steps 70-76. INTRØG interrogates SW(183). If the tape is to be updated (SW(183) = ØN), statement g is written; if the tape is to be generated, statement h is written. SW(137) is set ØFF. BAREA verifies that the B subprograms for M/T tape processing are in core. BENTRY transfers control to process MTTAPE. Step 58 is modified to continue at step 77 and control is transferred to step 55.

(18) Step 77. Statement i is written and control is transferred to step 60.

(19) Steps 78-81. INTRØG interrogates SW(173) to determine if the input common tape is to be processed (SW(173) = ØN). If ØFF, control is transferred to step 92; if ØN, INTRØG interrogates SW(184) to determine if tape is to be listed only (SW(184) = ØN). If ØN, control is transferred to step 67; if ØFF, control is transferred to step 6.

(20) Steps 82-86. Statement j is written and SW(175) is set ØFF. BAREA verifies that the B subprograms for input common tape generation are in core. BENTRY transfers control to process INPTNT. Step 58 is modified to continue at step 87 and control is transferred to step 55.

(21) Steps 87-91. Statement k is written. U04 reads the B2 area from tape A1. INTRØG interrogates SW(70) to determine if an error occurred in U04. If ØN, control is transferred to step 10; otherwise step 18 is modified to continue at step 28. Control is transferred to step 7.

(22) Steps 92-105. LAST is set to zero, and SW(24), SW(130), SW(141)-SW(150) and SW(153)-SW(157) are set ØFF. INTRØG interrogates SW(75) to determine if this is a DOC-OTC card (SW(75) = ØN). If ØFF, control is transferred to step 128. If ØN, ØTCTØT processes the OTC control card. INTRØG interrogates SW(70) to determine if an error occurred in ØTCTØT. If ØN, control is transferred to step 10; if ØFF, SW(31) is set ØN. INTRØG interrogates SW(83) to determine if this is a new target (SW(83) = ØN). If ØN, SW(127) is set ØFF. If ØFF, SW(127) is set ØN. The TGT and MST cards are read by ØTC2, and the paper tape identification is written. Step 108 is modified to continue at step 109.

(23) Steps 106-108. The duplicate target slot number IDPTG is examined to determine if any duplicate (same as) targets have to be set up. If there are any duplicate targets, the contents of TOID, UTDIN, TOGZ, IDGZP, and UBXTR corresponding to the duplicate target are set equal to the original target. Control is modified to continue at step 109 or 135.

(24) Steps 109-114. SW(158) is set ØN. Statement l is written, and BAREA verifies that the B subprograms for

OTC mode of operation are in core. HENTRY transfers control to process ØTCNT. The target flag for each of the ten targets is examined. If it is positive for any of the targets, control is transferred to step 115. If the flag is negative for all targets, all the targets are dummies. Statement m is written, and control is transferred to step 60.

(25) Steps 115-125. SW(128) is set ØN, and SW(75) and SW(157) are set ØFF. BAREA verifies that the B subprograms for OPC mode of operation are in core. HENTRY transfers control to process OPCNT. SW(75) is set ØN, and SW(128) and SW(127) are set ØFF. INGAIN sets up the next address in AGAIN and CKPTCK establishes a check point. After the targeting kit has been produced, statement n is written and control is transferred to step 127. After the target kit has been produced if RESTART verification is requested SW(158) is set ØFF. HENTRY transfers control to process the verification step of OTC. Statement o is written.

(26) Step 126. File A5 is ended.

(27) Step 127. Tape A5 is rewound and control is transferred to step 60.

(28) Steps 128-134. INTRØG interrogates SW(76) to determine if this is a DOC-TOT card (SW(78) = ØN). If ØFF, control is transferred to step 140; if ØN, the TOT control card is processed by ØTCTØT. INTRØG interrogates SW(70) to determine if an error occurred in ØTCTØT. If ØN, control

is transferred to step 10; if ØFF, OTC2 processes TGT and MST cards. Step 108 is modified to continue at step 135 and control is transferred to step 106 to test for duplicate targets.

(29) Steps 135-139. SW(31) is set ØFF and SW(158) is set ØN. Statement p is written, and BAREA verifies that the B subprograms for TOT mode of operation are in core. BENTRY transfers control to process ØTCNT. Statement q is written and control is transferred to step 126.

(30) Steps 140-142. INTRØG interrogates SW(77) to determine if this is a DOC-TAA control card (SW(77) = ØN). If ØFF, control is transferred to step 156. If ØN, J is tested to determine if the correct number of IDT cards have been read. If J equals three, control is transferred to step 145; otherwise the subprogram continues at the next step.

(31) Steps 143-144. Statement y is written. A four is stored in ITYER, and control is transferred to step 10.

(32) Steps 145-155. DØCTAA processes the TAA control card. INTRØG interrogates SW(70) to determine if an error occurred in DØCTAA. If ØN, control is transferred to step 10; otherwise statement r is written. SW(154) is set ØN and SW(175) is set ØFF. BAREA verifies that the B subprograms for M/T and R/L data are in core. BENTRY transfers control to process INPTNT. SW(154) is set ØFF. BAREA verifies that the B subprograms to process TAA mode of opera-

tion are in core. BENTRY transfers control to process TAANT. Statement a is written, and control is transferred to step 126.

(33) Steps 156-170. INTRØG interrogates SW(79) to determine if this is a DOC-SIM control card (SW(79) = ØN). If ØFF, control is transferred to step 171; if ØN, J is tested to determine if the correct number of IDT cards have been read. If J is other than three, control is transferred to step 143. DØCSIM processes the SIM card. INTRØG interrogates SW(70) to determine if an error occurred in DØCSIM. If ØN, control is transferred to step 10; if ØFF, statement t is written. SW(156) is set ØN and SW(175) is set ØFF. BAREA verifies that the B subprograms for M/T and R/L data are in core. BENTRY transfers control to process INPTNT. SW(156) is set ØFF. MSSNT controls the SIM and DEC functions. Statement u is written. File B3 is ended and rewound and control is transferred to step 88.

(34) Steps 171-180. INTRØG interrogates SW(78) to determine if this is a DOC-RSD card (SW(78) = ØN). If ØFF, statement x is written and the subprogram exits to HALT for manual intervention. If ØN, statement v is written. XDEW(1) -XDEW(800) are initialized to zero. SW(155) is set ØFF. BAREA verifies that the B subprograms for RSD mode of operation are in core. BENTRY transfers control to process RSDNT. Statement w is written and control is transferred to step 126.

2-121. SUBPROGRAM U13 (ERRPRT). ERRPRT determines the type of error that has occurred and prints and writes the appropriate error statement. The FORTRAN II reference statement is CALL ERRPRT.

a. Inputs. The inputs are SW(126) set ØN if the entrance was from U07, the contents of the type of error indicator register ITYER, and the identifier register IFLAG of the subprogram in which an error occurred. The card in BCD is in the card image area CDIO-CDI11 and is available for printout.

b. Outputs. The outputs are the following statements printed and written:

a. IFLAG = ________ ILLEGAL ITYER SETTING ITYER = ____
b. IFLAG = ________ TAPE ERROR CHECKSUM FAILURE
c. IFLAG = ________ TAPE ERROR CANNOT FIND RECORD
d. IFLAG = ________ FORMAT DISCREPANCY
e. IFLAG = ________ CARD ERROR
f. IFLAG = ________ DATA STORED PREVIOUSLY
g. IFLAG = ________ SWITCH SETTING ERROR
h. IFLAG = ________ PRINT ERROR
i. IFLAG = ________ TAPE REDUNDANCY ERROR
j. IFLAG = ________ END OF FILE
k. IFLAG = ________ END OF TAPE LOAD TAPE PRESS START KEY
l. IFLAG = ________ MACHINE ERROR
m. IFLAG = ________ THE ABOVE DEC CORRECTION CARD INDICATED IS IN ERROR

c. Program Logic. FD U13

(1) Step 1. ITYER is interrogated. If one or greater, the subprogram continues at step 4. If less than one, the setting of ITYER is in error and the subprogram continues at the next step.

(2) Step 2. Output statement a is written.

(3) Step 3. The program exits to HALT to determine, through manual intervention, if the error can be corrected and processing continued or if the program must stop.

(4) Steps 4-32. The following table indicates the output statement that is written, depending upon the contents of ITYER:

Contents of ITYER	Type of Error	Printout Statement
1	CHECKSUM FAILURE	b
2	CANNOT FIND RECORD	c
3	FORMAT DISCREPANCY	d
4	CARD ERROR	e
5	DATA STORED PREVIOUSLY	f
6	SWITCH SETTING ERROR	g
7	PRINT WHEEL ERROR	h
8	TAPE ERROR	i
9	END OF FILE	j
10	END OF TAPE	k
11	MACHINE ERROR	l
12	DECIMAL CORRECTION CARD ERROR	m

If ITYER contains other than the preceding, the subprogram continues at step 2. If ITYER contains a four or a five, the card image area CDIO-CDI11 is also printed. However if ITYER contains a four and the entrance was from U07 (SW(126) = ØN), nothing is printed, and the subprogram continues at step 3. U08 prints the columns in error. If ITYER contains a seven, U09 prints the columns in error. After the output or outputs are written, the subprogram continues at step 3, except if ITYER contains 11 in which case the subprogram exits to HALT.

2-122. SUBPROGRAM P19 (FUSING). FUSING computes the quantized parameter supplied to the re-entry vehicle fuzing mechanism by means of the OGE console for establishing the time of flight to detonation, and also computes the detonation point errors resulting from quantization of the fuzing parameter. The FORTRAN II reference statement is CALL FUSING.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FVAX (1,1)	2	Current missile velocity relative to local air mass, X component	$\dot{X}_{mr}$	ft/sec
FVAX (1,2)	2	Current missile velocity relative to local air mass, Y component	$\dot{Y}_{mr}$	ft/sec
FVAX (1,3)	2	Current missile velocity relative to local air mass, Z component	$\dot{Z}_{mr}$	ft/sec
GDPSM (1,1)	2	Detonation point position vector, X component	X	feet
GDPSM (1,2)	2	Detonation point position vector, Y component	Y	feet
GDPSM (1,3)	2	Detonation point position vector, Z component	Z	feet
GDPSM (1,7)	2	Time of flight since lift-off to detonation point	t_{det}	seconds
GDPSM (1,8)	2	Angle of velocity vector at detonation point relative to earth above local horizontal	γ_E	degrees
GDPSM (1,9)	2	Magnitude of velocity at detonation point relative to earth	V_E	ft/sec

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GTAUC (1,K,1)	2,9,1	Fuzing model coefficients	C_{ij}	
GRESM	2,9	Re-entry point summary data table		
GTAUC (1,10,1)	2	Reference re-entry velocity	V_{ref}	ft/sec
GTAUC (1,11,1)		Reference re-entry angle	γ_{ref}	degrees
FRTØD	2	Conversion constant: radians to degrees		deg/rad
GFZSM	2	Time since liftoff to first 25g point	t_{25g}	seconds

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
IRFSG (I,1,ISLØT)	2,1,10	Output fuzing parameter for all ten targets. (ISLØT = current target slot number)		
GFZSM (I,11,1)	2	Detonation point time of flight error due to fuzing quantization	t_{detE}	seconds
GFZSM (I,12,1)	2	Horizontal target aim point adjustment (X component) due to fuzing quantization	X_{tap}	feet
GFZSM (I,13,1)	2	Horizontal target aim point adjustment (Y component) due to fuzing quantization	Y_{tap}	feet
GFZSM (I,14,1)	2	Horizontal target aim point adjustment (Z component) due to fuzing quantization	Z_{tap}	feet
GFZSM (I,15,1)	2	Detonation point altitude adjustment due to fuzing quantization		feet

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GFZSM (I,16,1)	2	25g point detection time adjustment factor	τ	seconds

c. Program Logic. FD P19

(1) Steps 1-4. If this run is for a Nike Zeus shot, the fuzing parameter $\hat{i}$ is set to one and the program continues at step 8. Otherwise, expression (1) evaluates the 25g point detonation time adjustment factor τ and expression (2) evaluates $\hat{i}$.

(2) Steps 5-8. The integer value of the fuzing parameter i must be from 1 to 99. If $\hat{i}$ is less than one, i is set equal to one; if $\hat{i}$ is greater than 99, i is set equal to 99; if $\hat{i}$ is not an integer, it is rounded to the nearest whole number.

(3) Steps 9-17. Because of quantization of the fuzing parameter, adjustment is made in the time of flight by expression (3), in the horizontal position of the detonation point by expression (4), and in the altitude of the detonation point.

(4) Steps 18-19. IFLAG is set to identification integer 1619. CUTIE is stepped by one and control is returned to the user subprogram.

d. <u>Expressions</u>.

$$\tau = \sum_{i=0}^{2} \sum_{j=0}^{2} \left[C_{ij} \, (V - V_{ref})^{j} \, (\gamma - \gamma_{ref})^{i} \right] \qquad (1)$$

$$\hat{1} = 1 + 2.45 \left[1.3 \, (t_{det} - t_{25g} + \tau) - 15 \right] \qquad (2)$$

$$\Delta t = (i = \hat{1})/3.185 \qquad (3)$$

$$\vec{H} = \vec{V} \, t_{detE} - \vec{L} \, (\vec{L} \cdot \vec{V} \, t_{detE}) \qquad (4)$$

where

$\vec{H}$ = horizontal target aim point adjustment

$\vec{L}$ = unit vector up representing local vertical

$\vec{V} \, t_{detE}$ = difference between expected detonation point and desired detonation point

WWW.CHROMEHOOVES.NET

2-123. SUBPROGRAM P43 (GCØNST). GCØNST processes the ground guidance complex data and computes the G constants table for simulation. The FORTRAN II reference statement is CALL GCONST (IR, IL.)

a. Inputs. The inputs are IL which defines the selected launch pad and IR which defines the selected radar antenna. The following are also inputs:

COMMON TAG	DIMENSION	ITEM		UNITS
PRWLN	6	Geographic longitudes of radars west of Greenwich	λ_R	degrees
PLWLN	8	Geographic longitudes of launch pads	λ_L	degrees
PRWLT	6,1,1	Geographic latitudes of radars	L_{GR}	degrees
PLWLT	8	Geographic latitudes of launch pads	L_{GL}	degrees
PRWGS	2,1,1	Geoidal separation of radar	G_{SR}	feet
PRWAL	2,1,1	Altitude of radar above geoid	h_{rg}	feet
PLWAL	2,1,1	Altitude of launch pads above geoid	h_L	feet
FRTØD	2	Conversion constant radians to degrees $\left(\frac{180.0}{\pi} = 57.295780\right)$		deg/rad
PLWR	2	Reference azimuth bearing of launch pads	A_{ref}	degrees
PRWAW	2,1,1	West component - gravitational anomaly at radar	W	degrees
PRWAN	2,1,1	North component - gravitational anomaly at radar	N	degrees

COMMON TAG	DIMENSION	ITEM		UNITS
PRMCW	4,1,1	Nominal A_z code wheel readings when antennas are oriented on monolith	A_{MCW}	RAU
PRWMZ	4,1,1	Radar reference monolith azimuths	A_M	degrees
GDRAU	2	Number of degrees per radar angular unit ($\approx 6.1034063 \times 10^{-4}$)		deg/RAU
PRWEL	4,1,1	Nominal E_L code wheel readings when antennas are leveled		RAU
PRRCN	4,1,1	Range calibration numbers for radars A and B	D_{LP}	RAU

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM		UNITS
PRLØN	2	Geographic longitude of this radar	λ'_R	degrees
PLLØN	2	Geographic longitude of this launch pad	λ'_L	degrees
PRLAT	2	Geographic latitude of this radar	L_{GRT}	degrees
PLLAT	2	Geographic latitude of this launch pad	L_{GLT}	degrees
PRCLT	2	Radar geocentric latitude	L_{CR}	degrees
PRDRV	2	Radar distance to center of earth	r_r	feet
PLCLT	2	Launch pad geocentric latitude	L_{CL}	degrees
PLDRV	2	Launch pad distance to center of earth	r_e	feet

COMMON TAG	DIMENSION	ITEM		UNITS
PRSCL	2	Radar sine of geocentric latitude		pure no.
PRCCL	2	Radar cosine of geocentric latitude		pure no.
PRSGL	2	Radar sine of geographic latitude		pure no.
PRCGL	2	Radar cosine of geographic latitude		pure no.
PLSCL	2	Launch pad sine of geocentric latitude		pure no.
PLCCL	2	Launch pad cosine of geocentric latitude		pure no.
PLSRA	2	Launch pad sine of reference bearing		pure no.
GRXYZ	2,3	Inertial coordinates of radar at time of launch	X_R, Y_R, Z_R	feet
XGI	2,24	Matrix - G constants table for tape punching		
PLCRA	2	Launch pad cosine of reference bearing		pure no.

c. Program Logic. FD P43

(1) Steps 1-5. IFLAG is set to identification integer 1643. LGTØLC computes the geocentric latitude of the radar. ELLRAD computes the earth ellipsoid radius at radar. The distance from the radar to the center of the earth is computed by summing the earth ellipsoid radius at the radar point, the geoidal separation of radar, and the altitude of radar above geoid.

(2) Steps 6-8. LGTØLC and ELLRAD compute the geo-

centric latitude of the launch pad and the earth ellipsoid radius at launch pad. The distance from the launch pad to the center of the earth is computed by summing the earth ellipsoid radius at the launch pad, the geoidal separation of launch pad, and the altitude of launch pad above geoid.

(3) Steps 9-24. SINE and CØSINE compute the sine and cosine of the following parameters: radar geocentric latitude, radar geographic latitude, launch pad geocentric latitude, and reference azimuth bearing of launch pad. RØUND rounds the double-precision sine and cosine to single precision.

(4) Steps 25-26. IR is set equal to IL. If IR is equal to one, a computation is to be made, and the subprogram continues at step 27. Otherwise the G constant computations are excluded. CUTIE is stepped by one, and control is returned to the user subprogram.

(5) Steps 27-49. SINE and CØSINE compute the sine and cosine of the geographic latitude of mean radar. RØUND rounds the double-precision sine and cosine to single precision. LGTØLC and ELLRAD compute the geocentric latitude of mean radar and the earth ellipsoid radius at mean radar. The distance from the center of the earth to mean radar (R_R) is computed. SINE and CØSINE compute the sine and cosine of the mean geocentric radar $\not{O}_R$. RØUND rounds these values to single-precision equivalents. R_R sin $\not{O}_R$ and R_R cos $\not{O}_R$ are computed and the north and west component of gravitational

anomaly at the radar are computed. CØSINE and SINE compute the cosine and sine of the current reference azimuth bearing of the launch pad. RØUND rounds the double-precision sine and cosine to single precision. The XGI table of constants is set up. Constants G9 and G13 are examined. If not within the limits ±180 degrees, then 360 degrees are added or subtracted, as required, to bring the constants within the limits.

(6) Steps 50-61. Constant G17 is computed by use of expression (1) as the average range in feet to the missile acquisition point for the two handover complexes. If the nearest complex is used, the indicators are set to compute G17 by expression (2) as the time in cycles for the missile to reach the acquisition altitude. If the farthest complex is used, the indicators are set to compute G19 by expression (2) as the acquisition time. If the acquisition altitude h_{m1} of the nearest complex is zero or less, G18 or G19 is set to zero. The subprogram continues at step 62. If h_{m1} is greater than zero, the initial estimate of time t_1 to rise to acquisition altitude is computed by use of expression (3) with h_{m1}. The nominal time t_{1+1} is computed by use of expression (4). The absolute difference between t_{1+1} and t_1 is compared with 0.001 second. If the difference is greater than 0.001, t_{1+1} is recomputed using $t_1 = t_{1+1}$. The iterative process is repeated until the difference is 0.001 or less. Then t_1 is set equal to t_{1+1} and either constant G18 or G19 is computed depending on the indicator setting. If the constant is zero or less, it is set to zero. The subprogram continues at the next step.

(7) Steps 62-67. The acquisition altitude h_{m2} for the second complex is examined. If zero or less, G18 or G19 is set to zero, and the subprogram continues at step 68. If greater than zero, time t_1 is computed by use of expression (3) with h_{m2}. The nomi al time t_{1+1} is computed by the iterative process of the preceding paragraph until the absolute difference between t_{1+1} and t_1 is less than 0.001. Expression (4) is used with h_{m2}. Constant G18 or G19 is computed by use of expression (5). If the constant is zero or less, it is set to zero. The subprogram continues at the next step.

(8) Steps 68-69. The duplexed values of G17, G18, and G19 are set equal. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$G17 = \frac{1}{2}(\lambda_{R1} + \lambda_{R2}) \quad (1)$$

$$h_m = h_a - h_o \quad (2)$$

$$t_1 = \sqrt{2\, h_m/a_m} \quad (3)$$

$$t_{1+1} = t_1 - \frac{k_1\, t_1^{\,3} + k_2\, t_1^{\,2} - h_{m1}}{3\, k_1\, t_1^{\,2} + 2\, k_2\, t_1} \quad (4)$$

$$G18 \text{ or } G19 = \frac{0.945\, t_1 - 1.440412}{0.390412} - 2.5 \quad (5)$$

where

h_m = distance missile moves from liftoff to acquisition altitude

h_a = acquisition altitude

h_o = missile altitude at liftoff

t_{1+1} = time for nominal missile to rise to acquisition altitude

t_1 = initial estimate of time to rise to acquisition altitude

a_m = average acceleration of nominal missile = 13.43 ft/sec (including gravity)

k_1 = 0.03923249 ft/sec^2
k_2 = 5.921638 ft/sec
= nominal missile acceleration constants

2-124. SUBPROGRAM D48 (GIMBEX). GIMBEX prints a notification of an excessive gimbal angle detected in TAXIAL. If TAXIAL obtains the sine of a gimbal angle which is larger than the maximum allowed, SW(67) is set ØN. When this error is encountered, TAXIAL compares the excessive sine with any previous one encountered and stores the larger of the two as a parameter to GIMBEX. The other two parameters, sine of maximum allowed gimbal angle and time of flight since lift-off, will be set also. TAXIAL then continues normally. Under certain conditions, an excessive sine of a gimbal angle is not considered an error. TARGET determines if GIMBEX should be called to print an error notification. The FORTRAN II reference statement is CALL GIMBEX.

a. Inputs. The inputs as set by TAXIAL are as follows:

COMMON TAG	DIMENSION	ITEM
GXSNT	2	Sine of gimbal angle
GXLGA	2	Maximum allowable sine of a gimbal angle
GXTFL	2	Time of flight since liftoff and occurrence of largest excessive gimbal angle
FRTØD	2	Constant of conversion-radians to degrees

b. Outputs. The output is the following written statement:

GIMBAL ANGLE EXCEEDED LIMIT OF ______ DEGREES. A VALUE OF ______ DEGREES WAS ENCOUNTERED AT ______ SECONDS AFTER LIFT-OFF.

c. Program Logic. FD 048. The input representing the excessive gimbal angle sine is converted to radians by ARCSIN. The maximum allowable sine of gimbal angle is converted to degrees in the same manner. The output statement is printed and written.