AD/A-002 853

A STRUCTURAL WEIGHT ESTIMATION PROGRAM (SWEEP) FOR AIRCRAFT. VOLUME II- PROGRAM INTEGRATION AND DATA MANAGEMENT MODULE. PART 2: DATA MANAGEMENT MODULE

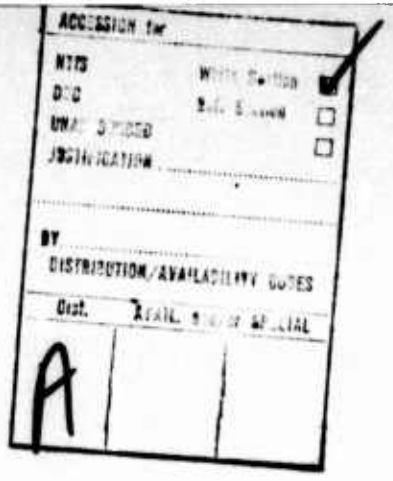
G. Hayase, et al

Rockwell International Corporation

Prepared for:

Aeronautical Systems Division

June 1974



NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Publication of this technical report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

JAMES H. HALL, Colonel, USAF
Deputy for Development Planning

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD/A002853

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ASD/XR 74-10	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A Structural Weight Estimation Program (SWEEP) for Aircraft Volume II: Program Integration and Data Management Module	5. TYPE OF REPORT & PERIOD COVERED
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): G. Hayase, R. Hiyama, C. Martindale, H. Rockwell	8. CONTRACT OR GRANT NUMBER(s): F33615-71-C-1922
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Rockwell International Corp, L.A. Aircraft Div, Los Angeles International Airport, Los Angeles, California 90009	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: FX2826-71-01876/C093
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Development Planning, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio	12. REPORT DATE: June 1974
	13. NUMBER OF PAGES: 829
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

weight estimation, structural weights, integrated computer programs, preliminary weight estimation, first-order weight estimations, aircraft structure weights, aircraft structural weight optimization, flutter optimization program, structural synthesis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Three computer programs were written with the objective of predicting the structural weight of aircraft through analytical methods. The first program, the structural weight estimation program (SWEEP), is a completely integrated program including routines for airloads, loads spectra, skin temperatures, material properties, flutter stiffness requirements, fatigue life, structural sizing, and for weight estimation of each of the major aircraft structural components. The program produces first-order weight estimates

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

and indicates trends when parameters are varied. Fighters, bombers, and cargo aircraft can be analyzed by the program. The program operates within 100,000 octal units on the Control Data Corporation 6600 computer. Two stand-alone programs operating within 100,000 octal units were also developed to provide optional data sources for SWEEP. These include (1) the flexible airloads program to assess the effects of flexibility on lifting surface airloads, and (2) the flutter optimization program to optimize the stiffness distribution required for lifting surface flutter prevention.

The final report is composed of 11 volumes. This volume (volume II) contains the methodology, program description, and user's information for the SWEEP control program, input data processing module, final output module, and the data management module.

TABLE OF CONTENTS

Section

Page

FORTRAN Listing (Cont)

LIST OF ILLUSTRATIONS

LIST OF TABLES

APPENDIX A

DATA MANAGEMENT MODULE FLOW CHARTS

AND FORTRAN LISTS

MODULE FLOW CHARTS AND FORTRAN LISTS

FLOW CHART USAGE

The automatically generated computer program flow charts (AUTOFLOW) presented in this document include a table of contents, flow charts, and FORTRAN lists of all routines in the module. The 80-column card lists are sequenced and grouped by routine.

Because the AUTOFLOW system used is IBM-oriented, the functions of the BUFFERIN and BUFFEROUT statements are not recognized, but these statements appear in proper order in note boxes. Also, the PROGRAM name does not appear on the main program, and library routines READMS and WRITMS are listed as undefined external references.

CROSS-REFERENCE LIST

The AUTOFLOW table of contents which precedes the flow charts and FORTRAN lists serves to cross reference the latter two. This table lists the following from left to right:

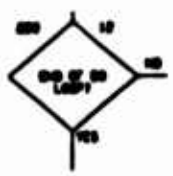
- The card identification from columns 73 through 80 of this card, or card sequence number. When sequence number is used in place of card identification, it is enclosed in parentheses.
- The page and box number where this card is displayed in a flow chart.
- The FORTRAN statement number from columns 1 through 5 of this card.
- The card identification(s) or sequence number(s) of the card(s) referring to this card (repeated as required).
- The pages and box numbers where the cards referring to this card are displayed in a flow chart (repeated as required).

FLOW CHARTS

The flow charts produced by AUTOFLOW use USASI conventional symbols. Since the flow charts are mechanically drawn from the program source deck, there are no omissions or vague generalizations about the processing within the boxes.

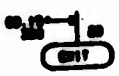
Every box on each page is uniquely numbered and may be referred to from elsewhere in the program. The source of a reference to a box will be indicated by showing the page and box number. If the number is followed by an asterisk, there are multiple references to this point, and the others may be found by using the cross-reference list.

The most-often-used symbol is the decision box. Like all boxes, its box number is above and to the right of the box. Its FORTRAN statement number is above and to the left of the box. The decision choices for the paths are printed.

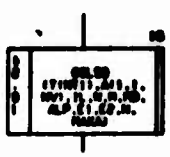


The unconditional transfer connecter has its page number destination printed above or to the left of the box number destination within the connecter. If there is a FORTRAN statement number at the destination, it is printed below the connecter.

The exit box example shows a connecter from page 9, box 15.



The subroutine call box includes the calling sequence. The page and box numbers of the flow chart of the called subroutine are shown on the left-hand side of the box. The page number is above the box number.

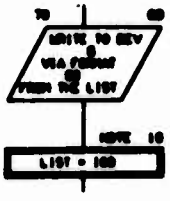


The note box encloses comments of a functional nature,

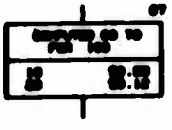
as differentiated from the 21 column comments, which are left justified without a box, that show the comment cards included in the FORTRAN deck.

The process box is used to enclose FORTRAN arithmetic statements.

Input and output are shown as communicating with a device. The list used follows, if appropriate:



The computed GØ TØ becomes a branch table showing the page and box number of each of the ordered branches.



The column connecters and initial connecters are the only boxes without external box numbers. The function of the initial connecter is always clear,

but the label given is the symbol in the next FORTRAN card, which is often blank.

The column connecter identifies the page and box number to which it connects.

TABLE OF CONTENTS

FOR

AUTOFLOW CHART SET

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

FORTRAN MODULE DATA MANAGEMENT MODULE

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - PROCEDURES

(000035) 2.02 700
(000038) 2.02 (000035) 2.03
(000054) 2.14 30 (000052) 2.13
(000057) 2.16 32 (000052) 2.13
(000070) 2.21 0000
(000070) 2.23 0000 (000070) 2.20
(000130) 3.04 0001
(000147) 3.08 (000151) 3.12
(000151) 3.12 04 (000147) 3.00
(000153) 3.14 (000157) 3.18
(000157) 3.18 08 (000154) 3.16
(000150) 3.70 0002 (000130) 3.03
(000160) 3.24 0003
(000173) 3.28 0004 (000160) 3.25
(000191) 3.29 70 (000160) 3.28
(000195) 3.30 00 (000160) 3.29

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE AMDATA

(000204) 6.01 AMDATA (000000) 2.20-X
(000240) 6.02 (000240) 6.03
(000240) 6.03 101
(000254) 6.07 (000253) 6.12
(000253) 6.12 150
(000207) 6.22 200 (000205) 6.21
(000208) 6.23 (000200) 6.24
(000200) 6.24 210
(000207) 6.20 (000200) 6.20
(000200) 6.20 220
(000300) 6.31 (000303) 7.01
(000303) 7.01 230
(000305) 7.03 240
(000308) 7.04 (000300) 7.05
(000308) 7.05 250
(000342) 7.08 (000315) 7.10
(000315) 7.10 260
(000317) 7.12 300 (000000) 6.21 (000304) 7.02
(000347) 7.23 (000330) 6.01
(000330) 8.0. 310
(000330) 8.13 0001
(000423) 8.16 0002 (000300) 8.12

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE AMDASC

(000414) 11.01 AMDASC (000352) 8.02-X
(000420) 11.05 10 (000173) 11.12 (000420) 11.15 (000502) 11.10
(000422) 11.10 00 (000400) 11.00
(000475) 11.13 20 (000400) 11.00
(000420) 11.16 40 (000400) 11.00
(000500) 11.10 00 (000400) 11.00
(000540) 11.01 000

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

CHART TITLE - SUBROUTINE ADDHR

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE ADDHR

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE CONST

CARD ID PAGE/COL NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/COL)

CARD ID	PAGE/SEQ	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)		
(0010767)	00.16		(0010777) 00.16		
(0010777)	00.16	105			
(0010790)	00.17	200	(0010837) 00.00	(0010950) 27.10	
(0010800)	00.18	202			
(0010833)	00.20	204			
(0010840)	00.21	205			
(0010880)	00.22	206	(0010893) 00.20		
(0010897)	00.23	207	(0010905) 00.21		
(0010900)	00.25		(0010901) 00.27		
(0010901)	00.27	208			
(0010905)	00.28	209	(0010902) 00.18		
(0010909)	00.31	0211			
(0011100)	00.32	0212			
(0011102)	00.34	0213	(0010900) 00.30	(0010909) 00.31	
(0011103)	00.38	0215	(0011101) 00.33		
(0011107)	00.39		(0011102) 00.40		
(0011109)	00.40	222			
(0011110)	00.41	300	(0010790) 00.17		
(0011111)	00.42	301			
(0011117)	00.01	302	(0011115) 00.43		
(0011118)	00.02	303	(0011115) 00.43		
(0011120)	00.04	304			
(0011121)	00.05	305			
(0011123)	00.07	306	(0011119) 00.03	(0011120) 00.04	
(0011127)	00.11	309	(0011122) 00.06		
(0011128)	00.12		(0011129) 00.13		
(0011129)	00.13	310			
(0011131)	00.14	311	(0011110) 00.41		
(0011132)	00.15	312			
(0011138)	00.16	3121			
(0011139)	00.19	3122			
(0011141)	00.01	3123	(0011137) 00.17	(0011138) 00.18	
(0011145)	00.05	313	(0011140) 00.20		
(0011146)	00.06		(0011147) 00.07		
(0011147)	00.07	314			
(0011148)	00.08	315	(0011131) 00.14		
(0011150)	00.10		(0011064) 00.33		
(0011161)	00.11	402			
(0011163)	00.15	404	(0011163) 00.12		
(0011168)	00.17		(0011171) 00.20		
(0011171)	00.20	405	(0011168) 00.17		
(0011171)	00.20		(0011170) 00.18		
(0011174)	00.23	406			
(0011175)	00.24	407			
(0011177)	00.26	408	(0011173) 00.22	(0011174) 00.23	
(0011181)	00.30	417	(0011165) 00.14	(0011175) 00.25	
(0011182)	00.31		(0011183) 00.32		
(0011183)	00.32	418			
(0011184)	00.33	419	(0011180) 00.10		
(0011188)	01.01	420	(0011188) 00.34		
(0011192)	01.04		(0011193) 01.05		
(0011193)	01.05	422			
(0011198)	01.08		(0011200) 01.10		
(0011200)	01.10	424			
(0011203)	01.12	425	(0011201) 01.11		
(0011207)	01.15		(0011200) 01.10		
(0011208)	01.16	426			
(0011210)	01.17	427	(0011201) 01.11		
(0011216)	01.20	4001			
(0011217)	01.21	4002			
(0011219)	02.01	4003	(0011215) 01.19	(0011216) 01.20	
(0011223)	02.05	420	(0011210) 01.22		
(0011224)	02.06		(0011225) 02.07		
(0011226)	02.07	430			
(0011228)	02.09	431			
(0011232)	02.12		(0011233) 02.13		
(0011233)	02.13	434			
(0011234)	02.14	435	(0011227) 02.08		
(0011238)	02.15		(0011241) 02.21		
(0011240)	02.18	440			
(0011240)	02.18		(0011238) 02.20		
(0011241)	02.20	444			

CARD ID PAGE/SEQ# NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE STATUS

CARD ID PAGE/BOX NAME REFERENCES (SOURCE EVIDENCE NO. AND PAGE/BOX)

CARD ID PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE DOSPR

CARD ID PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE DSTNDR

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE DSTINI

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE DSTIRP

CARD ID PAGE/BOX NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE SWEEP

CARD NO PAGE/SEQ NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE FUNCO

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE HACKED

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE NUSUB0

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE PRTQ04

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE QUIKIE

CARD ID	PAGE/BOX	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/BOX)
(003201)	110.00 01		
(003204)	110.00 05	(003220) 110.07	
(003230)	110.12 05		
(003231)	110.13 00	(003220) 110.11	
(003232)	110.14	(003233) 110.15	
(003233)	110.15 02		
(003235)	110.16	(003232) 110.16	
(003238)	110.19 04		
(003240)	110.21 05	(007705) 117.11	
(003244)	110.22 100		
(003245)	110.23 102		
(003247)	110.01 104		
(003248)	110.02 110	(003245) 110.24	
(003249)	110.03	(003250) 110.04	
(003250)	110.04 114		
(003252)	110.06	(003254) 110.07	
(003254)	110.07 116		
(003258)	110.10 120		
(003264)	110.12 122	(003252) 110.11	
(003268)	110.13 150	(003244) 110.22 (003250) 110.04 (003252) 110.11	
(003271)	110.14 155	(003243) 110.21	
(003272)	110.16 160		
(003275)	110.18 162		
(003278)	110.19 164	(003275) 110.17	
(003282)	120.01 165	(003272) 110.15	
(003288)	120.04 168	(003278) 110.18 (003281) 110.19	
(003290)	120.05 182		
(003301)	120.08 190	(003288) 120.04	
(003302)	120.07 192		
(003305)	120.09 200	(003301) 120.05	
(003307)	120.00 201		
(003308)	120.13 202	(003304) 120.12	
(003309)	120.14 204	(003304) 120.12	
(003310)	120.15 205	(003307) 120.13	
(003313)	120.17 204	(003311) 120.16	
(003314)	120.18 206	(003311) 120.16	
(003317)	121.01 208	(003315) 120.18	
(003318)	121.02 210	(003315) 120.19	
(003323)	121.03 300	(003308) 120.08	
(003324)	121.04 302		
(003331)	121.08 304	(003331) 121.07	
(003332)	121.09 306	(003331) 121.07	
(003335)	121.10 308	(003323) 121.02	
(003341)	121.12 400	(003323) 121.03	
(003342)	121.13 402		
(003344)	121.17 404	(003340) 121.16	
(003353)	122.01 406	(003340) 121.16	
(003354)	122.02 408	(003352) 121.17	
(003358)	122.04 450	(003341) 121.12	
(003360)	122.05 452		
(003361)	122.08 454	[illegible] 122.04	
(003367)	122.09 462	[illegible] 122.07	
(003370)	122.00 470	[illegible] 122.07	
(003372)	122.10 480	[illegible] 122.08	
(003374)	122.12 482		
(003377)	122.13 488	(003373) 122.11	
(003378)	122.14 492		
(003383)	122.15 500	(003360) 122.05 (003377) 122.13	
(003384)	122.16 502		
(003402)	122.21 510		
(003404)	123.01 520	(003381) 122.00	
(003407)	123.02 600	(003383) 122.15 (003403) 122.21	
(003408)	123.03	(00+000) 123.04	
(00+000)	123.04 610		
(00+002)	123.06	(00+004) 123.07	
(00+004)	123.07 612		
(00+006)	123.08	(00+007) 123.10	
(00+007)	123 10 614		
(00+013)	123.13 6140	(00+008) 123.11	
(00+016)	123.16	(00+019) 123.17	
(00+019)	123.17 615		
(00+02+)	123.22	(00+026) 123.23	

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE STATEMENT NO. AND PAGE/SEQ)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE SPRAT

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE TEMPRE

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE HEIDST

CARD ID PAGE/SEQ# NAME REFERENCES (SOURCE STATEMENT NO. AND PAGE/SEQ#)

CHART TITLE - NON-PROCEDURAL STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE HEADER

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)

CARD ID PAGE/SEQ NAME REFERENCES (SOURCE STATEMENT NO. AND PAGE/SEQ)

(0040000) 100.11 000
(0040040) 100.12 100 (0040040) 100.05
(0040080) 100.13 004 (0040080) 100.10
(0040120) 100.14 000 (0040070) 100.11
(0040160) 100.15 010
(0040200) 100.16 000 (0040200) 100.14
(0040240) 100.17 000
(0040280) 100.01 004 (0040280) 100.10
(0040320) 100.04 000 (0040280) 100.15 (0040200) 100.17
(0040360) 100.05 040 (0040070) 100.02
(0040400) 100.11 000 (0040070) 100.02

CHART TITLE - NON-PROGRAMMING STATEMENTS

CHART TITLE - INTRODUCTORY COMMENTS

CHART TITLE - SUBROUTINE HEADER

(0000140) 100.01 HEADER (0000750) 2.04-0
(0000700) 100.02 (0000750) 100.04
(0000750) 100.04 100
(0000700) 100.05 (0000000) 100.10
(0000022) 100.10 140
(0000101) 100.14 (0000100) 100.10
(0000100) 100.10 150
(0000100) 100.10 (0000110) 100.00
(0000110) 100.00 104
(0000121) 100.22 (0000122) 100.24
(0000122) 100.24 100
(0000131) 100.00 (0000130) 100.30
(0000130) 100.30 104
(0000140) 100.32 (0000141) 100.34
(0000141) 100.34 100
(0000147) 100.01 170 (0000140) 100.30
(0000140) 100.02 172 (0000100) 100.04
(0000140) 100.02 174
(0000154) 100.00 170
(0000102) 100.00 170 (0000140) 100.02
(0000104) 100.00 100
(0000107) 100.00 102 (0000103) 100.07
(0000120) 100.10 102 (0000100) 100.00
(0000022) 100.12 (0000100) 100.13
(0000022) 100.13 104
(0000170) 100.10 (0000170) 100.10
(0000172) 100.10 105
(0000170) 100.10 100 (0000171) 100.17
(0000177) 101.01 100 (0000140) 100.30
(0000102) 101.05 100 (0000300) 100.30 (0000204) 104.00
(0000100) 101.07 104
(0000100) 101.00 000 (0000104) 101.00
(0000000) 101.00 (0000002) 101.10
(0000002) 101.10 005
(0000004) 101.12 200 (0000004) 101.11 (0000240) 102.04 (0000247) 102.05
(0000002) 101.13 202 (0000200) 100.00
(0000204) 101.15 220
(0000000) 101.17 210
(0000204) 101.10 214 (0000700) 101.10
(0000000) 101.00 202 (0000007) 101.21
(0000000) 101.01 204
(0000000) 102.01 000 (0000004) 101.11 (0000240) 102.00
(0000210) 102.02 000
(0000214) 102.04 004 (0000000) 102.02
(0000210) 102.00 000 (0000217) 102.07
(0000210) 102.00 004
(0000210) 102.00 000
(0000210) 102.00 200 (0000210) 102.00
(0000011) 102.11 000
(0000004) 102.12 002 (0000000) 102.10
(0000000) 102.13 270 (0000003) 102.11
(0000004) 102.10 (0000202) 102.10
(0000202) 102.10 270

CARD ID	PAGE/SEQ	NAME	REFERENCES (SOURCE SEQUENCE NO. AND PAGE/SEQ)
(005237)	162.10	(005240) 162.23	
(005238)	162.21 274		
(005240)	162.23 276	(005238) 162.20	
(005247)	162.25 280		
(005248)	162.25 282		
(005260)	163.01 306	(005269) 161.20	
(005268)	163.03 330	(005265) 161.20	
(005269)	163.03	(005268) 163.01	
(005270)	163.04 332	(005270) 163.00	
(005271)	163.05 334		
(005273)	163.07 336	(005270) 163.04	
(005274)	163.08 338	(005270) 163.08	
(005275)	163.08 340	(005303) 161.10 (005270) 163.04 (005273) 163.07	
(005278)	163.11	(005302) 163.12	
(005284)	163.12 342		
(005288)	163.14	(005288) 163.15	
(005289)	163.15 344		
(005292)	163.16	(005294) 163.18	
(005294)	163.18 350		
(005298)	163.22	(005303) 163.20	
(005300)	163.24 352		
(005303)	163.25 354	(005298) 163.23	
(005304)	163.27 400	(005253) 161.14	
(005305)	163.28 402		
(005310)	163.29 450	(005304) 163.27	
(005313)	163.30 460	(005310) 163.29	
(005316)	163.32	(005320) 163.35	
(005318)	163.34 462		
(005320)	163.35 465	(005317) 163.33	
(005329)	164.01 470	(005310) 163.29	
(005330)	164.03	(005332) 164.06	
(005330)	164.06 472		
(005332)	164.06 475	(005329) 164.04	
(005339)	164.07 480	(005310) 163.29	
(005339)	164.08	(005343) 164.11	
(005341)	164.10 482		
(005343)	164.11 488	(005340) 164.08	
(005348)	164.13	(005351) 164.16	
(005350)	164.15 492		
(005351)	164.16 498	(005349) 164.14	
(005355)	164.18	(005417) 166.09	
(005356)	164.19 502	(005356) 164.18 (005351) 165.01	
(005361)	165.01 510	(005356) 164.18	
(005363)	165.02 512	(005361) 165.01	
(005417)	166.09 550	(005378) 164.23	
(005422)	166.11	(005440) 166.18	
(005428)	166.12 710	(005422) 166.11 (005438) 166.13	
(005438)	166.13 720	(005442) 166.11	
(005442)	166.14 730	(005438) 166.13	
(005446)	166.16 750	(005427) 166.12	

CHART TITLE - NON-PROCEDURAL STATEMENTS

[illegible] TABLE OF DIAGNOSTICS AUTOFLOW CHART SET - DEMO PAGE 1

LOCATION		DIAGNOSTIC
CARD ID	PAGE/SEQ	
[illegible]	2.04	UNDEFINED - 'READ10' EXTERNAL REFERENCE
[illegible]	3.02	UNDEFINED - 'WRITE' EXTERNAL REFERENCE
[illegible]	3.07	UNDEFINED - 'WRITE' EXTERNAL REFERENCE
[illegible]	10.10	UNDEFINED - 'WRITE' EXTERNAL REFERENCE
[illegible]	10.24	UNDEFINED - 'WRITE' EXTERNAL REFERENCE
[illegible]	20.01	UNDEFINED - 'READ10' EXTERNAL REFERENCE
[illegible]	20.07	UNDEFINED - 'WRITE' EXTERNAL REFERENCE
[illegible]	20.10	UNDEFINED - 'WRITE' EXTERNAL REFERENCE
[illegible]	20.01	UNDEFINED - 'READ10' EXTERNAL REFERENCE
[illegible]	20.12	UNDEFINED - 'WRITE' EXTERNAL REFERENCE

CHART TITLE - INTRODUCTORY COMMENTS

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
                              PROGRAM DATAIN
ddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
```

MAIN CONTROL DATA MANAGEMENT, READ MT DIST. AND INERTIA
WRITTEN 25 MAY 1972

CHART TITLE - PROCEDURES

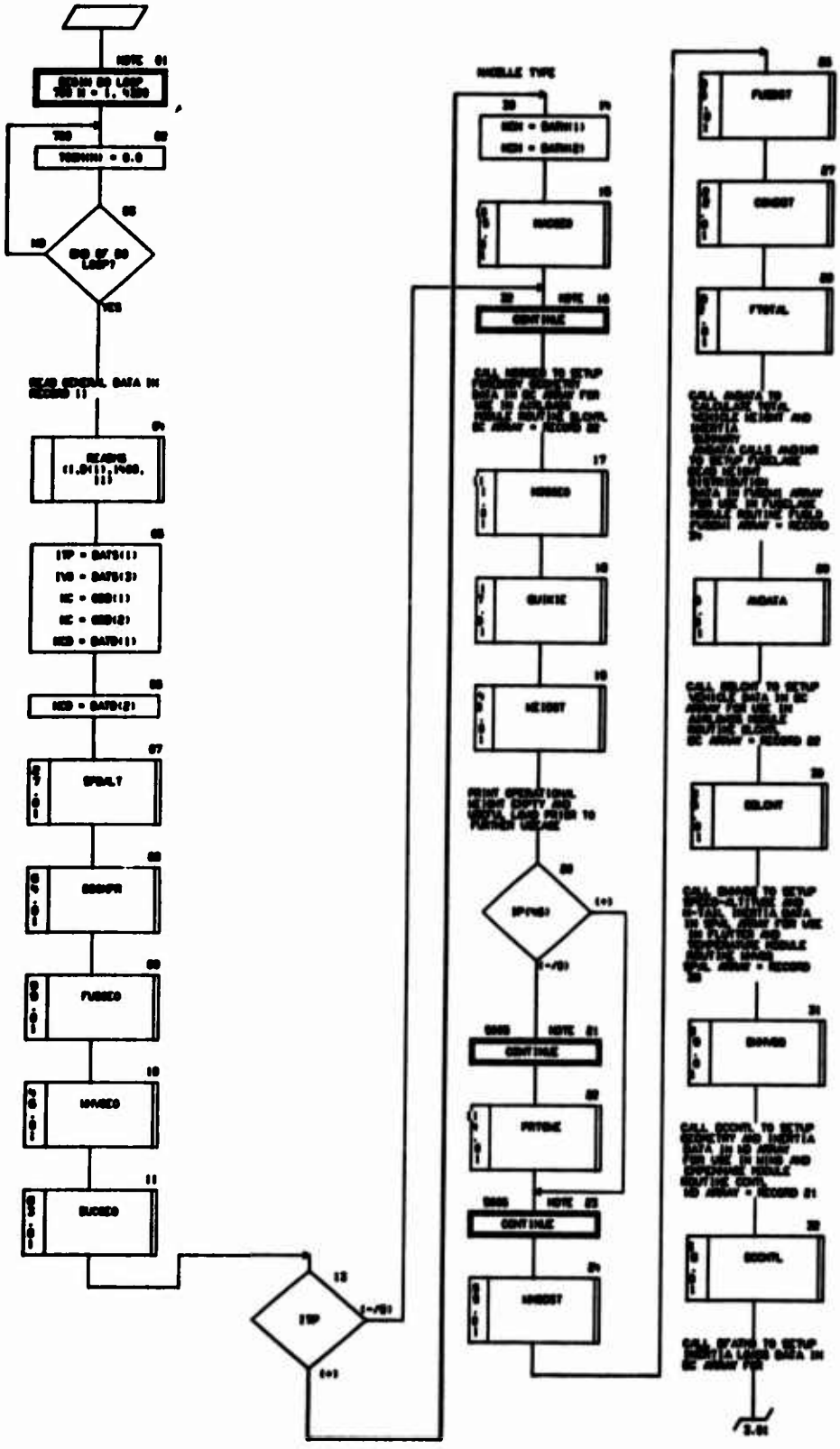


CHART TITLE - PROCEDURES

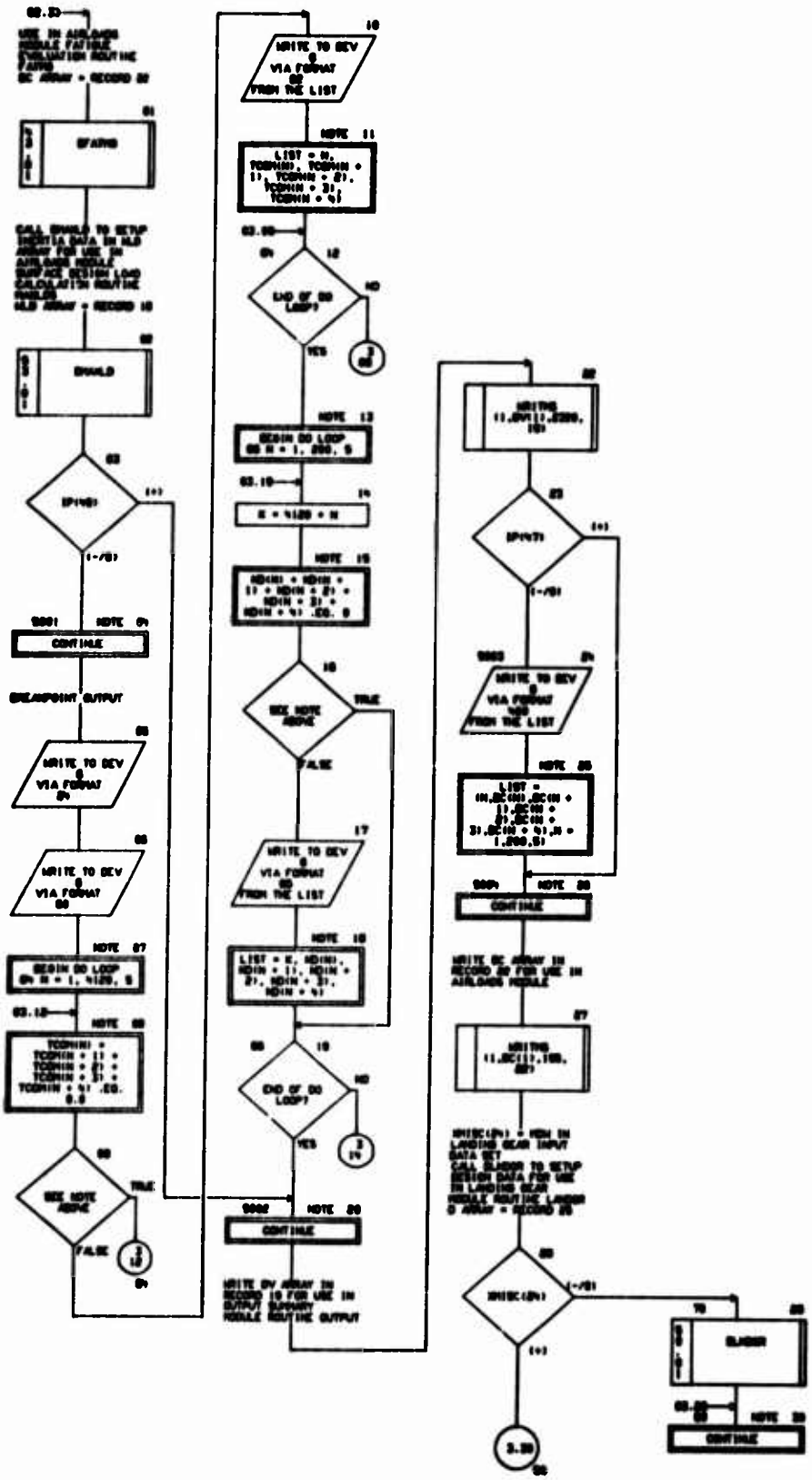


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
          PROGRAM DATAIN
           COMMON [illegible]
          COMMON /[illegible]/ [illegible]
          COMMON /[illegible]/ [illegible]
          DIMENSION [illegible]
          DIMENSION [illegible]
          DIMENSION [illegible]
          EQUIVALENCE [illegible]
           [illegible]
          EQUIVALENCE [illegible]
          EQUIVALENCE [illegible]
           [illegible]
          EQUIVALENCE [illegible]
          EQUIVALENCE [illegible]
           [illegible]
[illegible]       FORMAT(1H1,[illegible] TOSH AT END OF DATA MANAGEMENT LINK **,
            [illegible] DATAIN - [illegible] **)
[illegible]       FORMAT([illegible] VEHICLE GEOMETRY AND MISC DATA FILE ***)
[illegible]       FORMAT([illegible])
[illegible]       FORMAT([illegible] )
[illegible]       FORMAT(1H1,[illegible] SC ARRAY --- LOADS DATA - - RECORD [illegible] ***,
           [illegible] DATAIN - [illegible] **//
           (10X,I3,5F10.4) )
```

[illegible] AUTOFLOW CHART SET - DEBUG DATA MANAGEMENT MODULE PAGE [illegible]

CHART TITLE - [illegible]

****

[illegible]

****

CHART TITLE - SUBROUTINE ADATA

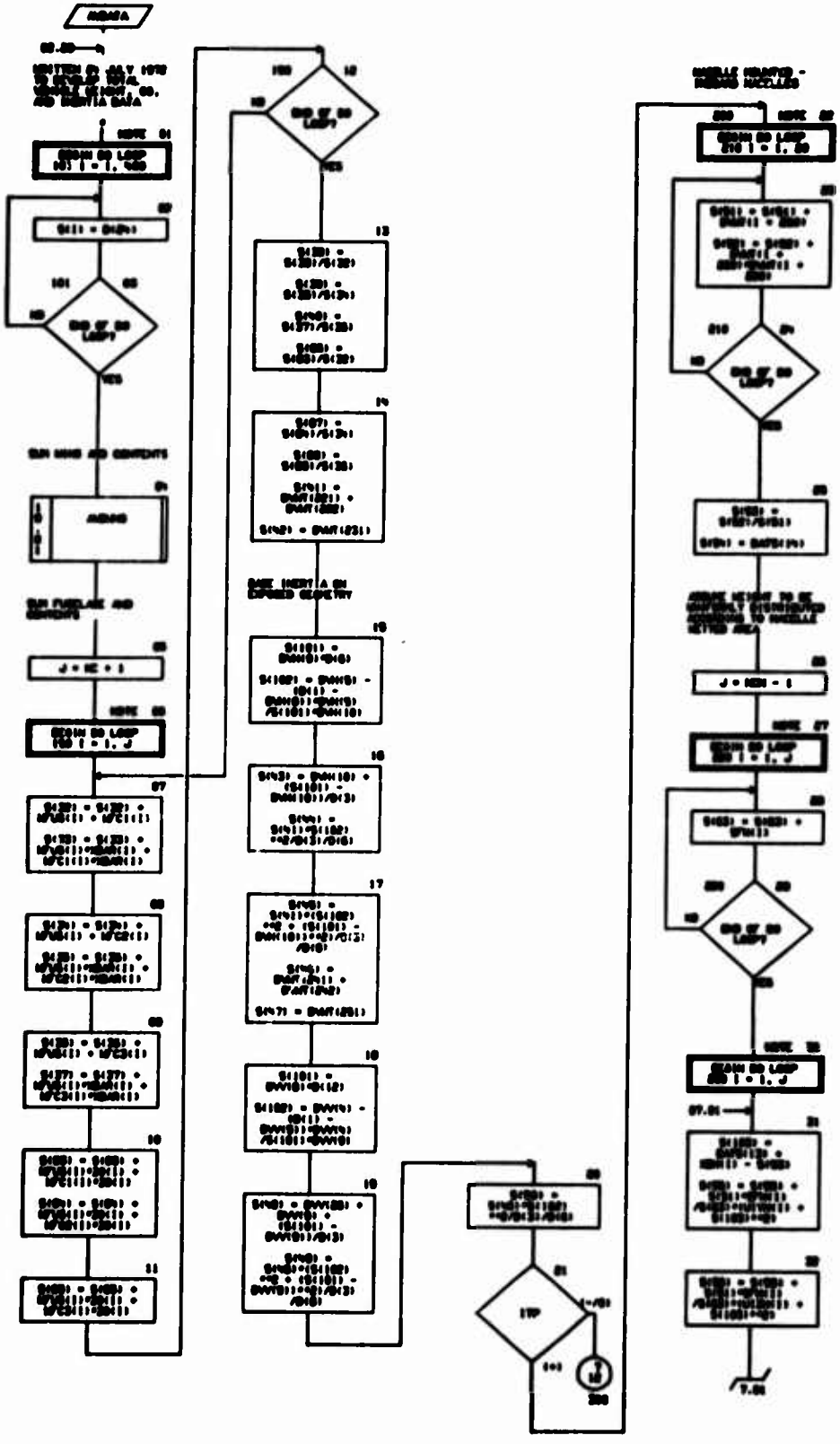


CHART TITLE - ELEVATING AGAIN

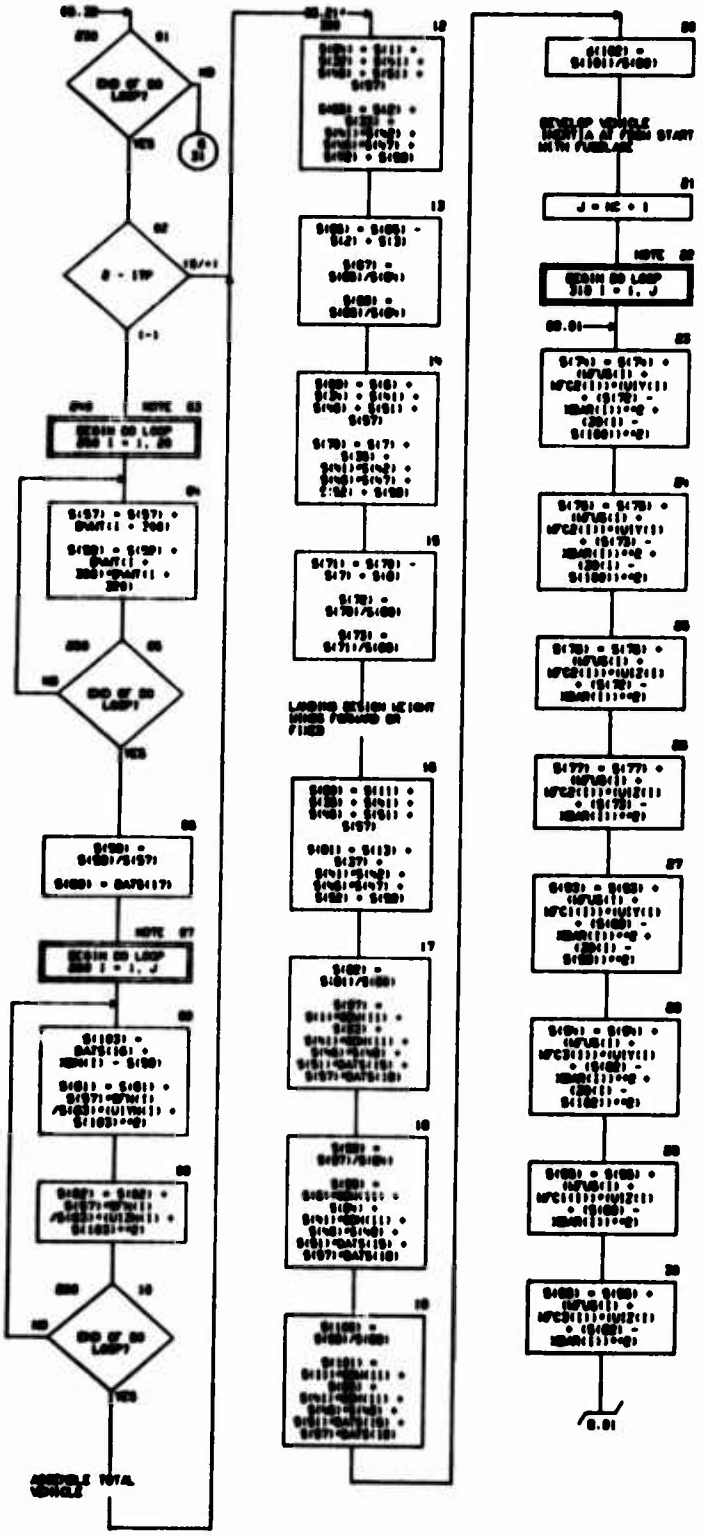


CHART TITLE - SUBROUTINE ANDATA

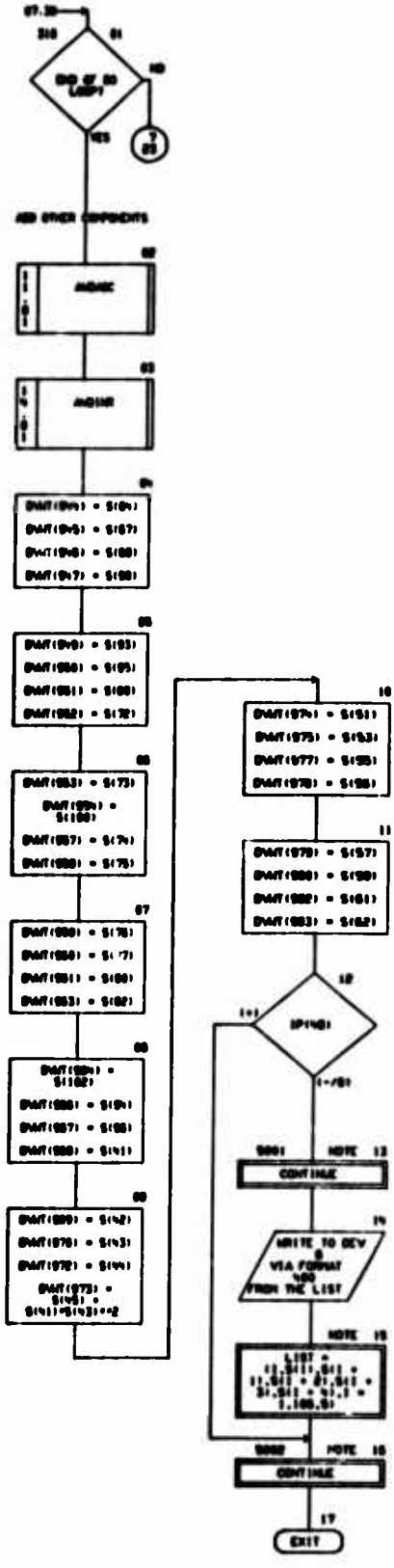


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /PRINT/ IP(80)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION GDH(50),GDH(40),GDV(40)
      DIMENSION DATS(40)
      DIMENSION DVH(30),DVV(30),DVB(440),DVH(150),DVAT(1000)
      DIMENSION I0(20),KBAR(20),ULY(20),UIZ(20)
      DIMENSION SYM(10),UIYM(10),UI2M(10),KBM(10)
      DIMENSION NFUS(20)
      DIMENSION NFC1(20),NFC2(20),NFC3(20)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GDH(1),GD(25)),(GDH(1),GD(30)),(GDV(1),GD(3+1))
      EQUIVALENCE (DATS(1),GD(46))
      EQUIVALENCE (DVH(1),DV(37)),(DVV(1),DV(40)),(DVB(1),DV(43)),
     (DVH(1),DV(97)),(DVAT(1),DV(112))
      EQUIVALENCE (I0(1),DVB(1)),(KBAR(1),DVB(19)),
     (ULY(1),DVB(40)),(UIZ(1),DVB(42))
      EQUIVALENCE (SYM(1),DVH(7)),(UIYM(1),DVH(13)),
     (UI2M(1),DVH(19)),(KBM(1),DVH(1))
      EQUIVALENCE (NFUS(1),DVAT(3+1))
      EQUIVALENCE (NFC1(1),DVAT(70)),(NFC2(1),DVAT(80)),
     (NFC3(1),DVAT(82))
      EQUIVALENCE (ITP,ND(111)),(NC,ND(115)),(KCM,ND(119))
400   FORMAT(1H1 5(1) FROM AVDATA,71X,2HP* AVDATA - IP(48) **/
        (110, 9F10.4))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*********************************************************************
C                      SUBROUTINE ANDASC
C*********************************************************************
```

CHART TITLE - SUBROUTINE ADASC

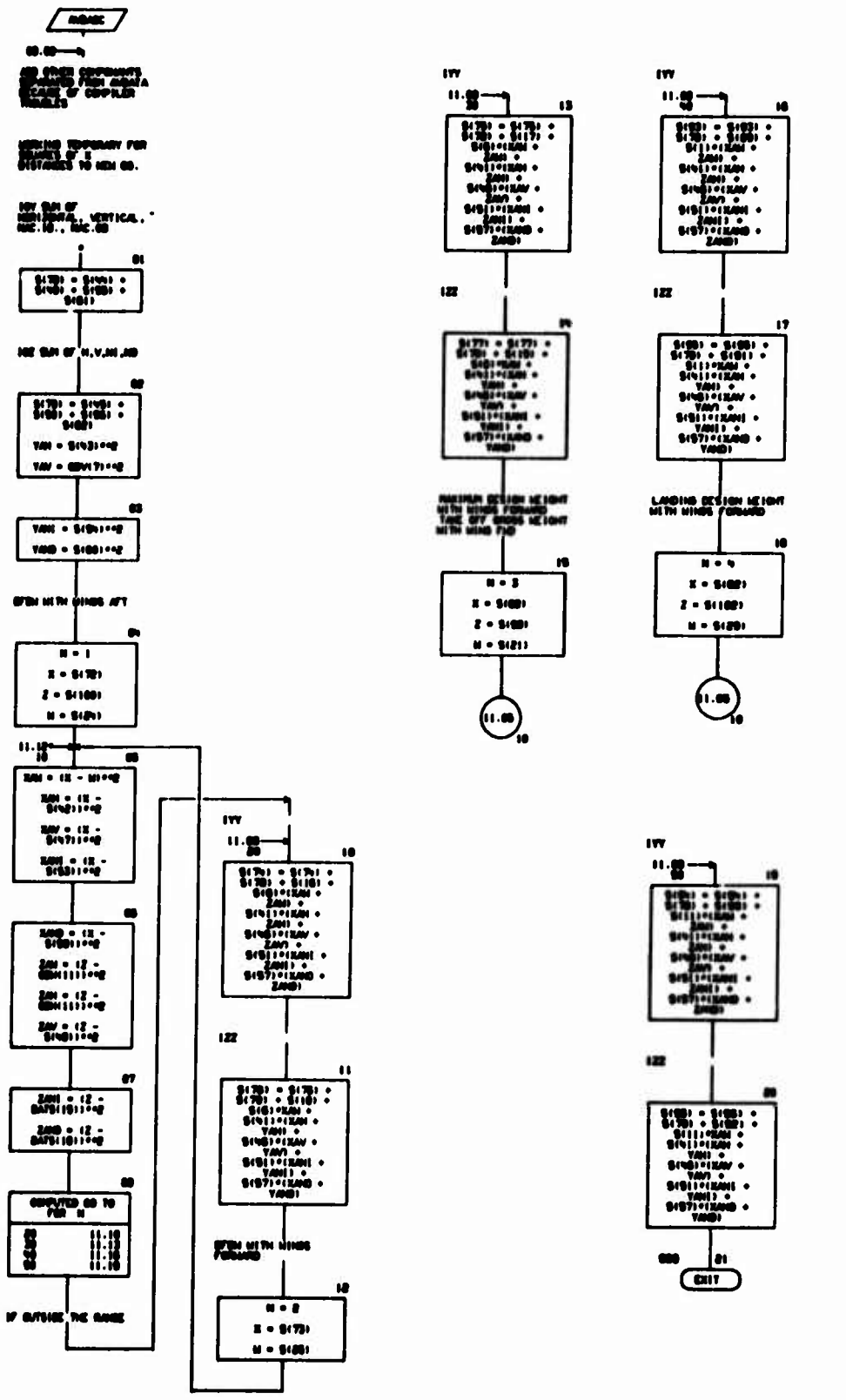


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMMON TCOM(9200)
DIMENSION D(1700),GD(1700),DV(2120),S(1400),AD(200)
DIMENSION  GDH(50),GDH(40),GDV(40)
DIMENSION  DATS(40)
EQUIVALENCE(D(1),TCOM(1)),(GD(1),TCOM(70)),(DV(1),TCOM(140)),
 (S(1),TCOM(372)),(AD(1),TCOM(912))
EQUIVALENCE (GDH(1),GD(25)),(GDH(1),GD(30)),(GDV(1),GD(39))
EQUIVALENCE (DATS(1),GD(48))
EQUIVALENCE (XAH,S(110)), (XAV,S(111)), (XAV,S(112)),
 (XAH,S(113)), (XAND,S(114))
EQUIVALENCE                (YAH,S(115)), (YAV,S(116)),
 (YAH,S(117)), (YAND,S(118))
EQUIVALENCE (ZAH,S(119)), (ZAH,S(120)), (ZAV,S(121)),
 (ZAH,S(122)), (ZAND,S(123))
EQUIVALENCE (X,S(124)), (Z,S(125)), (N,S(126))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*****************************************************************
C                      SUBROUTINE ANDINR
C*****************************************************************
```

CHART TITLE - SUBROUTINE ADDNR

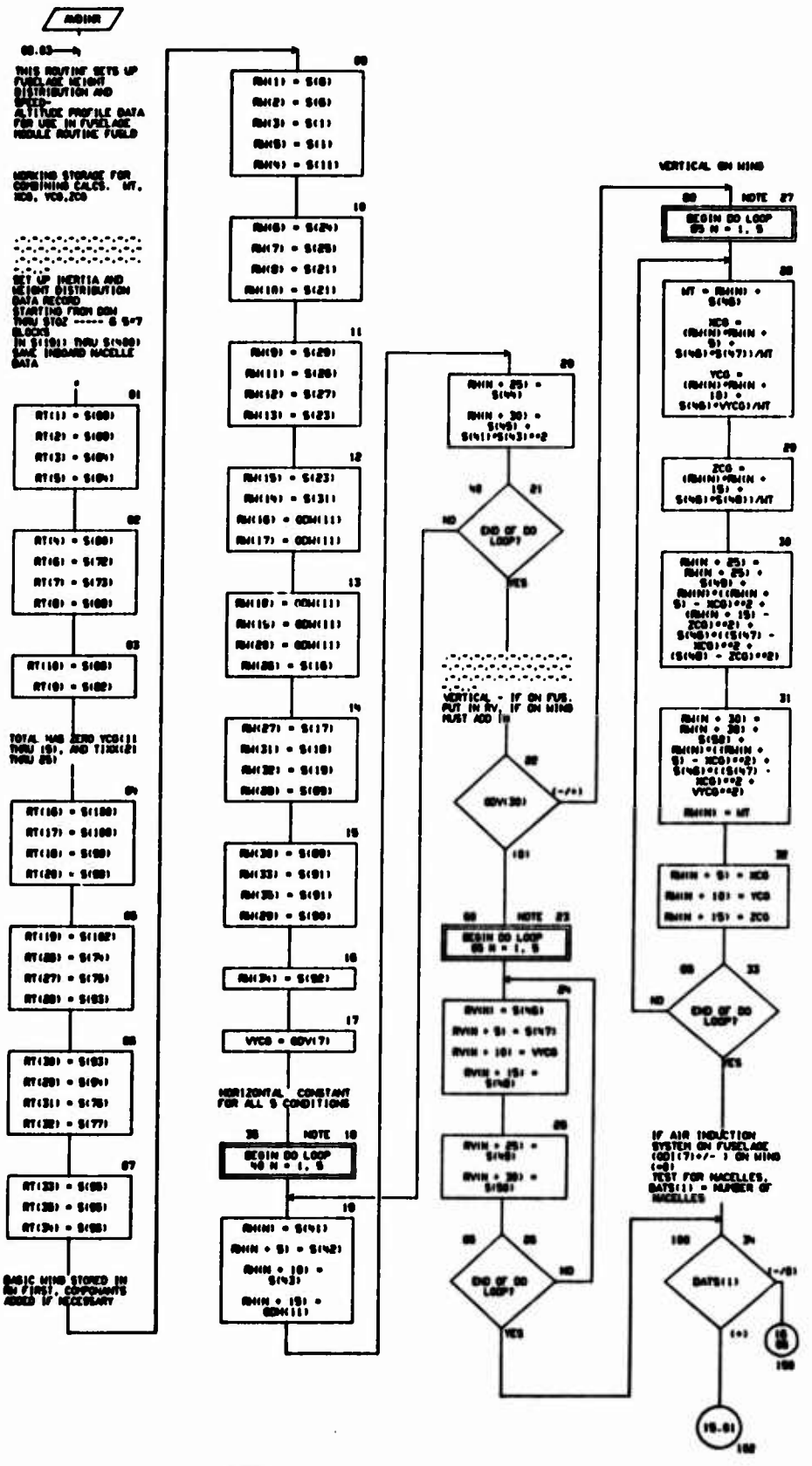


CHART TITLE - SUBROUTINE AGINR

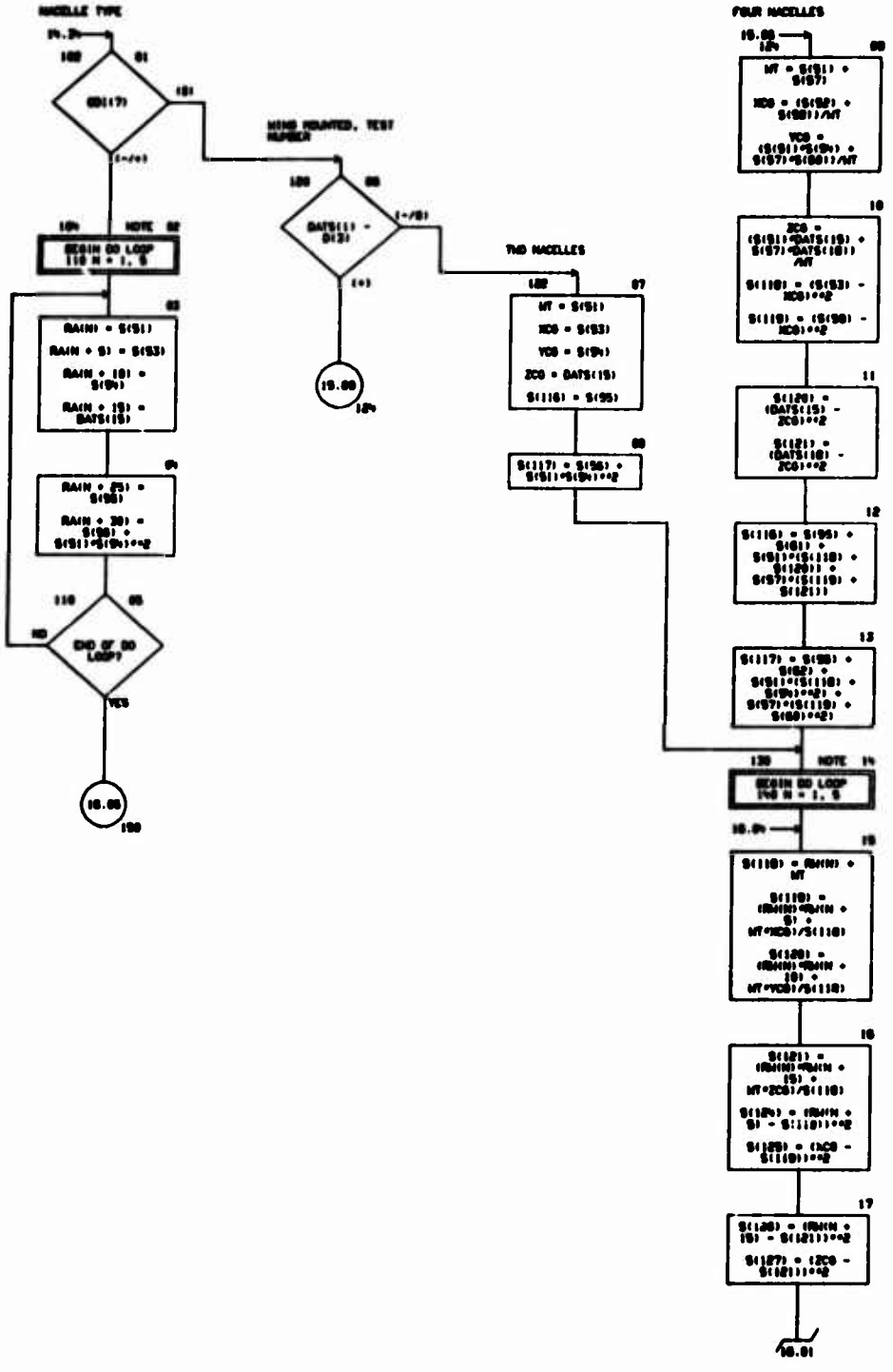


CHART TITLE - SUBROUTINE ANDINR

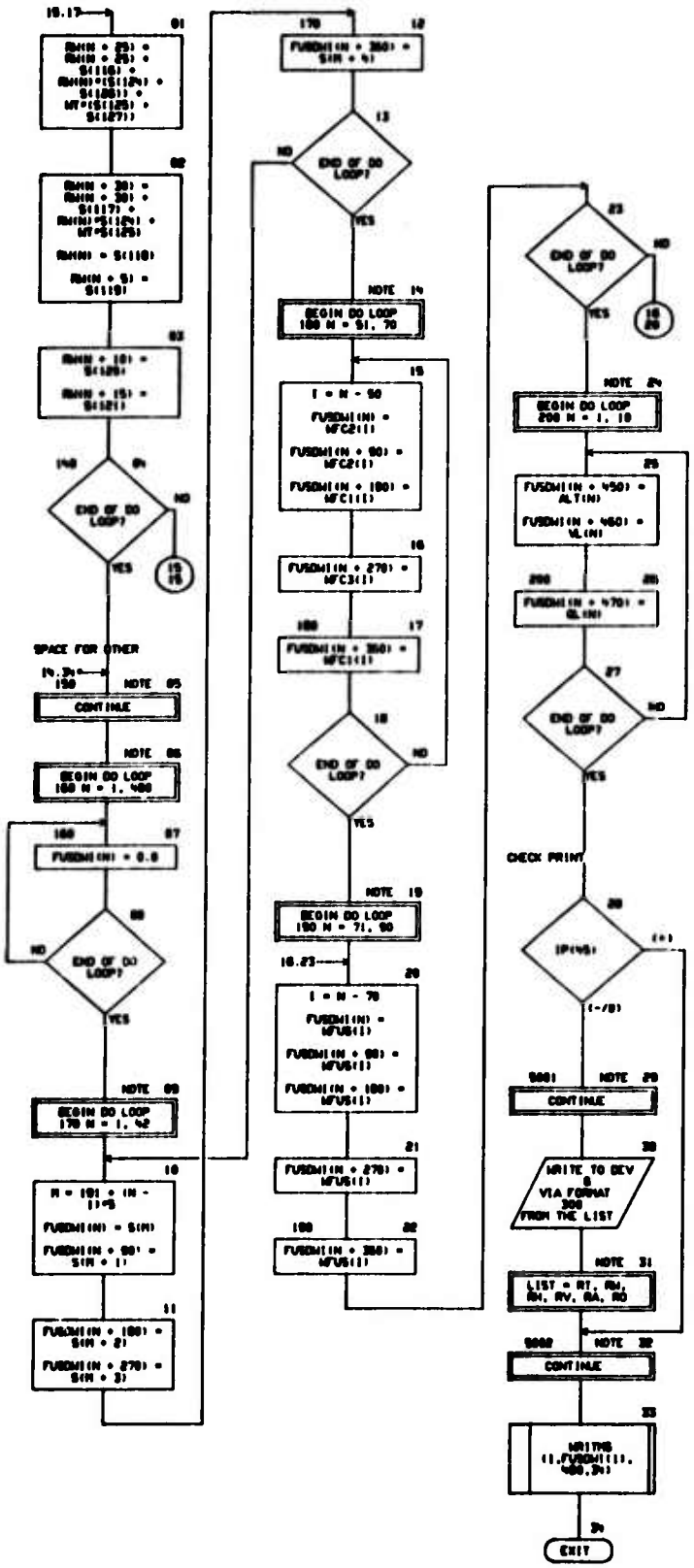


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4200)
      COMMON /IPRINT/ IP(60)
      DIMENSION B(700),GD(700),DV(2200),S(400),ND(200)
      DIMENSION FUCOM(400)
      DIMENSION GD(120)
      DIMENSION GDM(90),GDM(40),GDV(40)
      DIMENSION DATS(40)
      DIMENSION DVAT(1000)
      DIMENSION ALT(10),VL(10),GL(10)
      DIMENSION MFUS(20),MFC1(20),MFC2(20),MFC3(20)
      DIMENSION RT(30),RM(30),RH(30),RV(30),RA(30),RD(30)
      EQUIVALENCE (B(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3701)),(ND(1),TCOM(4101))
      EQUIVALENCE (GD(1),GD(1))
      EQUIVALENCE (GDM(1),GD(25)),(GDM(1),GD(30)),(GDV(1),GD(31))
      EQUIVALENCE (DATS(1),GD(40))
      EQUIVALENCE (DVAT(1),DV(112))
      EQUIVALENCE (ALT(1),DV(1)),(VL(1),DV(71)),(GL(1),DV(91))
      EQUIVALENCE (MFUS(1),DVAT(31)),(MFC1(1),DVAT(70))
     , (MFC2(1),DVAT(90)),(MFC3(1),DVAT(92))
      EQUIVALENCE (RT(1),S(10)), (RM(1),S(250)),(RH(1),S(25)),
     (RV(1),S(250)), (RA(1),S(33)), (RD(1),S(350))
      EQUIVALENCE ( MT,S(110)),( XC0,S(111)),( YC0,S(112)),(ZC0,S(113))
      EQUIVALENCE (VYC0,S(115)), (AZC0,S(114))
  200 FORMAT(1HI,2BHCHECK PRINT FOR ANOINR,6BX,21H** ANOINR - IP(45) **,
                                                  //3HRT, 9E10.6/6(3X,9E10.6/)
     /3HRM, 9E10.6 / 6( 3X,9E10.6/ ) /3HRH, 9E10.6 /6(3X,9E10.6 /)
     /3HRV, 9E10.6 / 6( 3X,9E10.6/ ) /3HRA, 9E10.6 /6(3X,9E10.6 /)
     /3HRD, 9E10.6 / 6( 3X,9E10.6/ ) )
```

```
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
                          SUBROUTINE AHEAD
))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
```

CHART TITLE - SUBROUTINE ADD00

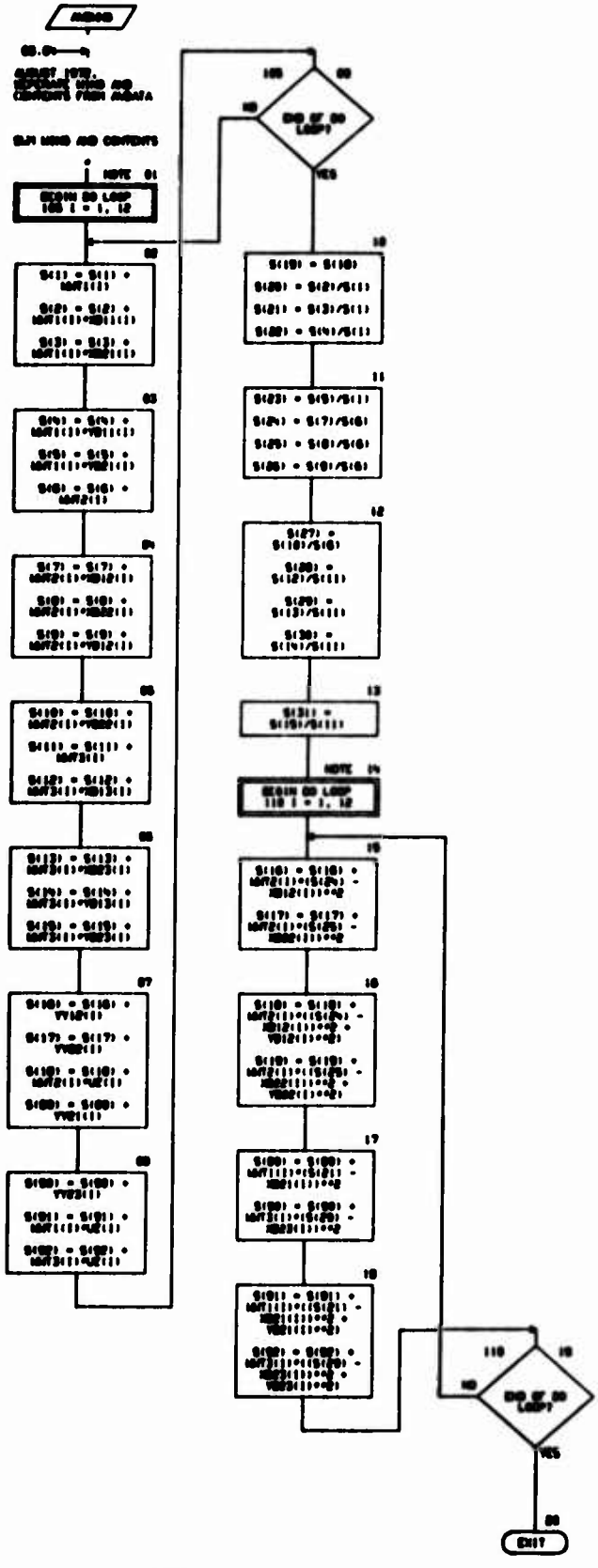


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(320)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION DVAT(1000)
      DIMENSION          UZ(12),UVT1(12),UVT2(12),UVT3(12),
     1 ND1(12),ND21(12),YD11(12),YD21(12),ND12(12),ND22(12),
     2 YD12(12),YD22(12),ND13(12),ND23(12),YD13(12),YD23(12),
     3 YY12(12),YY02(12),YY21(12),YY23(12)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (DVAT(1),DV(1121))
      EQUIVALENCE                    (UZ(1),DVAT(961)),
     1 (UVT1(1),DVAT(997)),(UVT2(1),DVAT(952)),(UVT3(1),DVAT(949)),
     2 (ND1(1),DVAT(588)),(ND2 (1),DVAT(581)),(YD11(1),DVAT(593)),
     3 (YD21(1),DVAT(605)),(ND12(1),DVAT(617)),(ND22(1),DVAT(629)),
     4 (YD12(1),DVAT(641)),(YD22(1),DVAT(653)),(ND13(1),DVAT(665)),
     5 (ND23(1),DVAT(677)),(YD13(1),DVAT(689)),(YD23(1),DVAT(701)),
     6 (YY12(1),DVAT(713)),(YY22(1),DVAT(725)),(YY21(1),DVAT(737)),
     7 (YY23(1),DVAT(749))
```

CHART TITLE - INTRODUCTORY COMMENTS

EXECUTIVE CONSOL

CHART TITLE - SUBROUTINE CONDS1

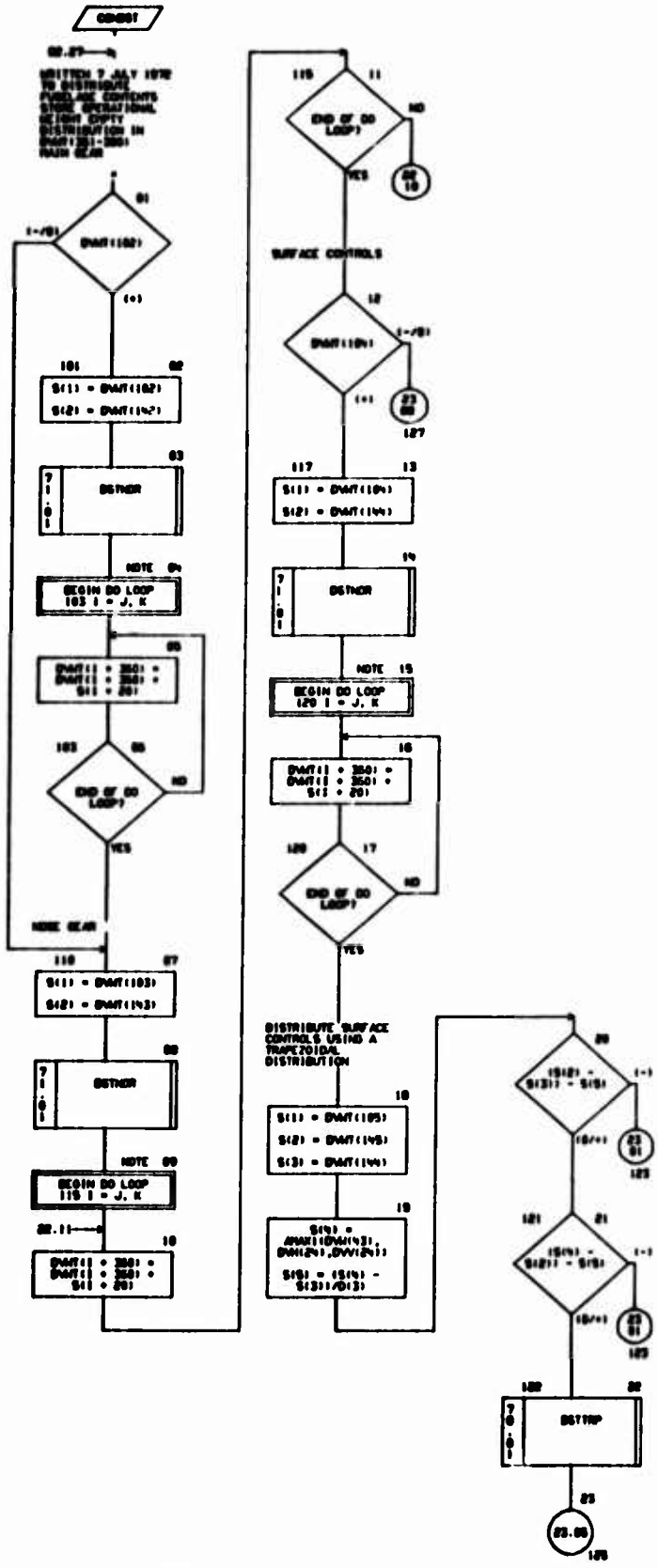


CHART TITLE - SUBROUTINE GOOEY

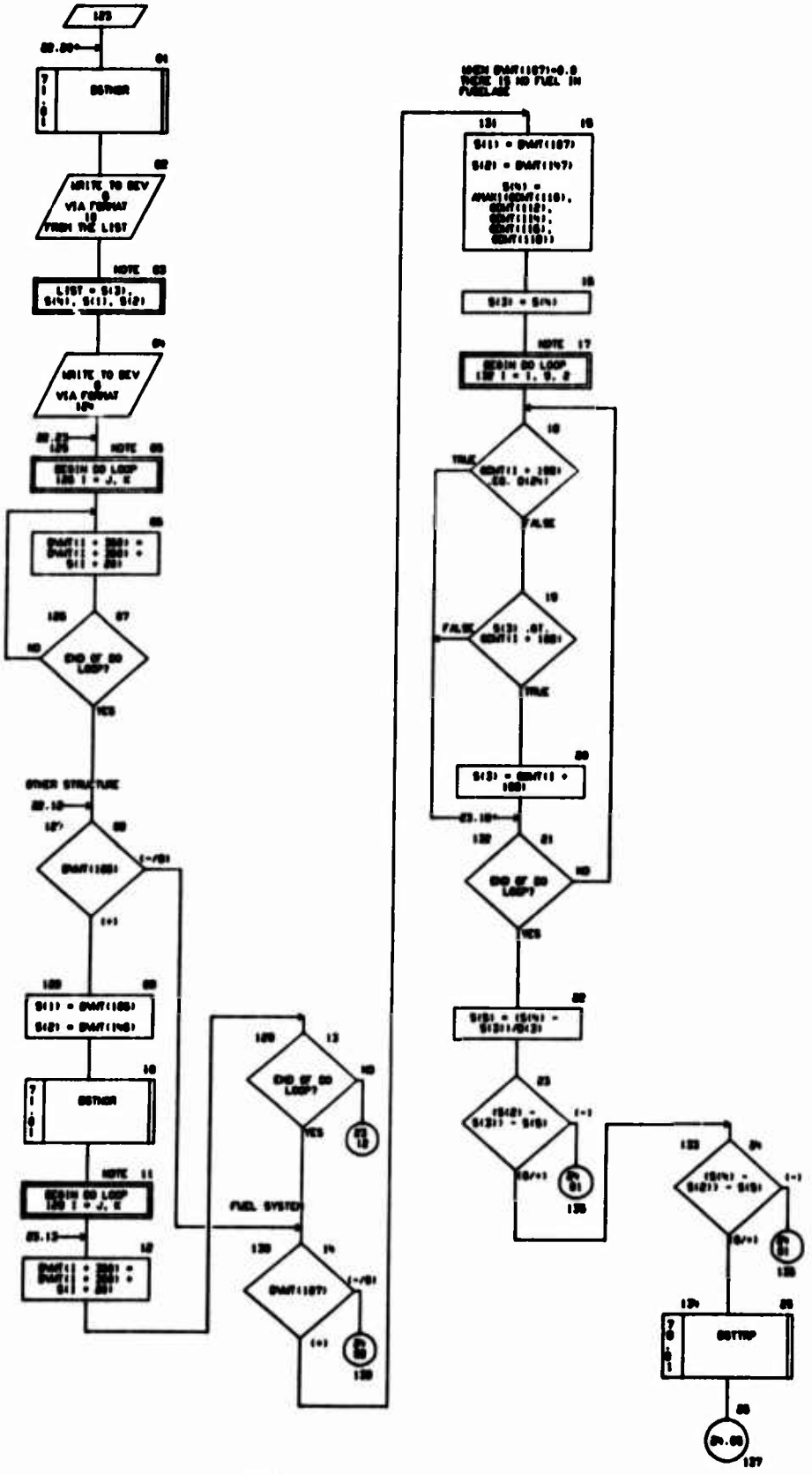


CHART TITLE - SUBROUTINE CONDST

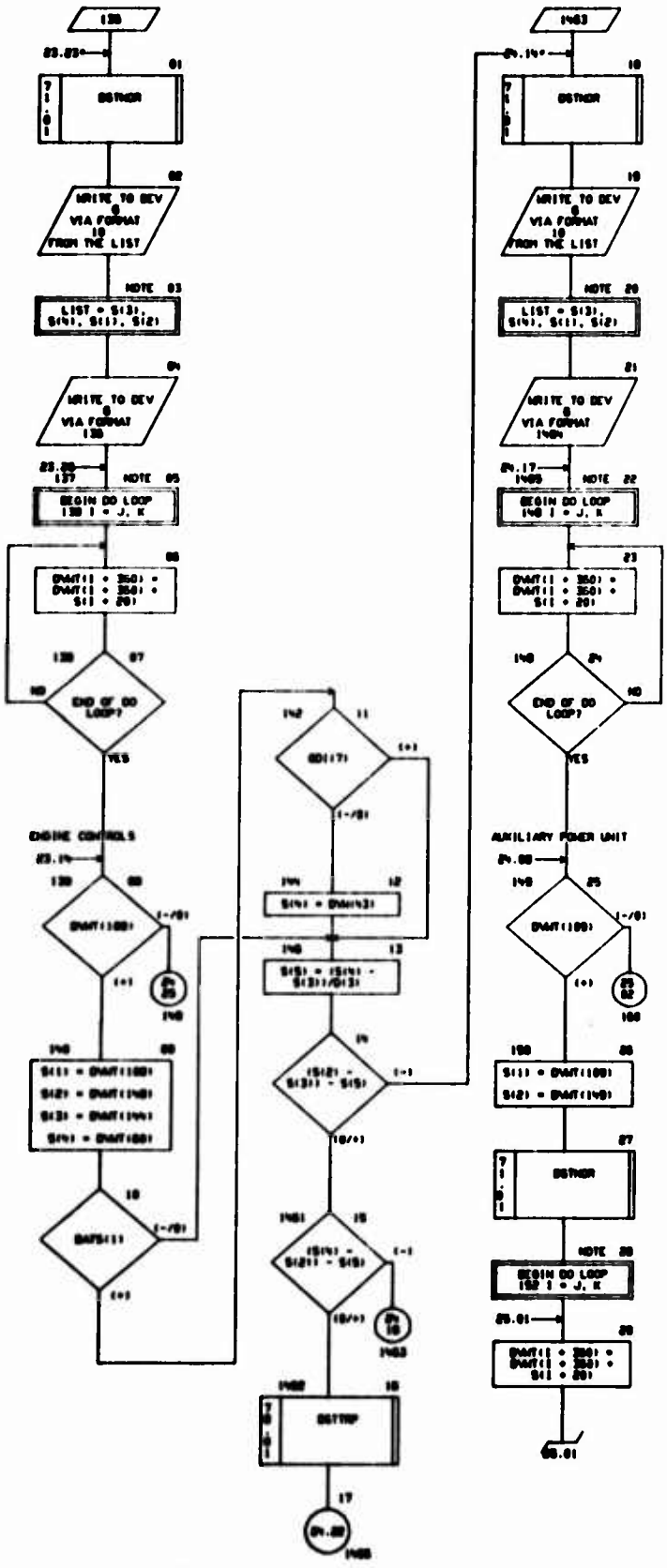


CHART TITLE - SUBROUTINE ORDER

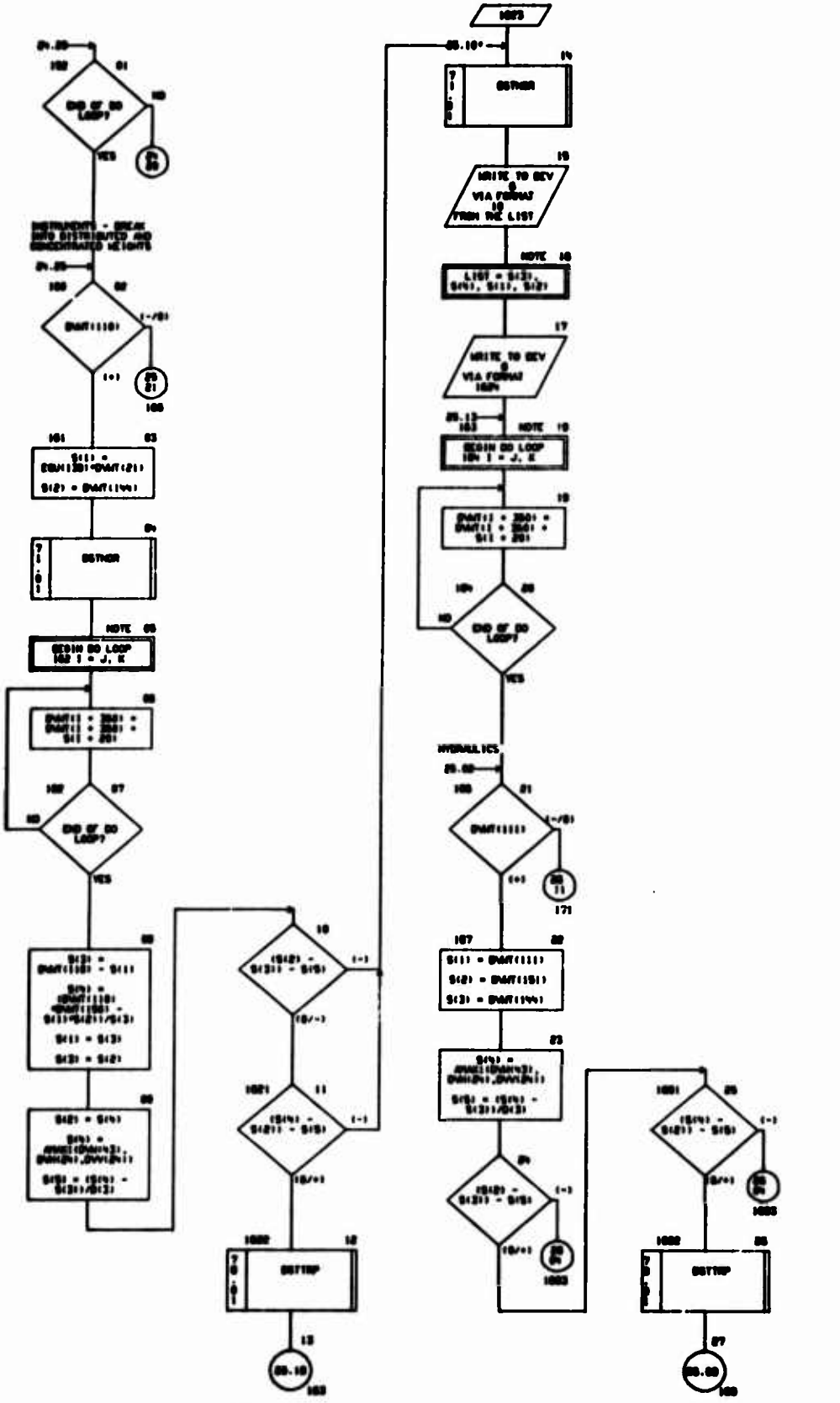


CHART TITLE - SUBROUTINE CONDST

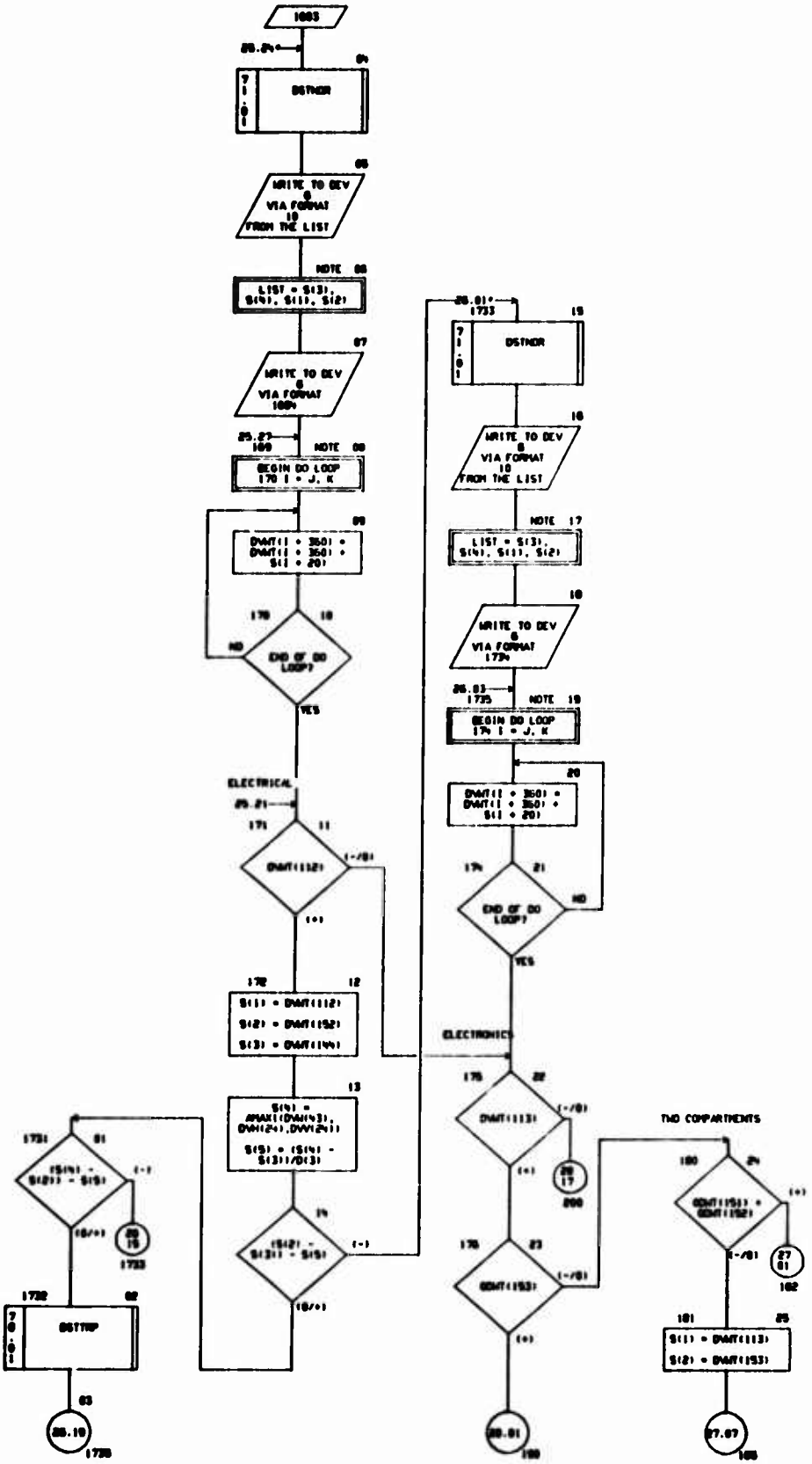


CHART TITLE - SUBROUTINE CONDST

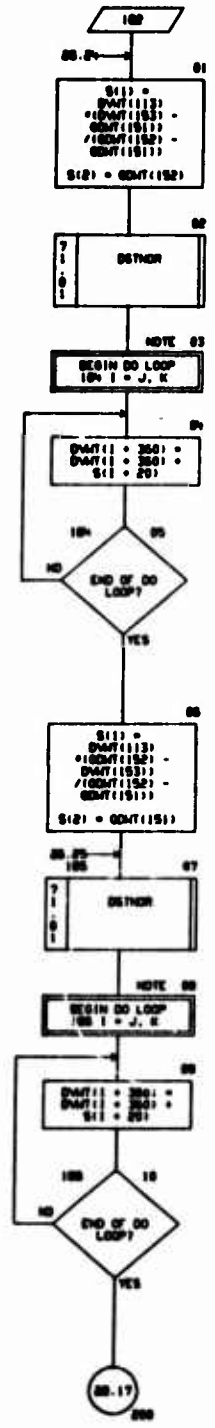


CHART TITLE - SUBROUTINE CONDST

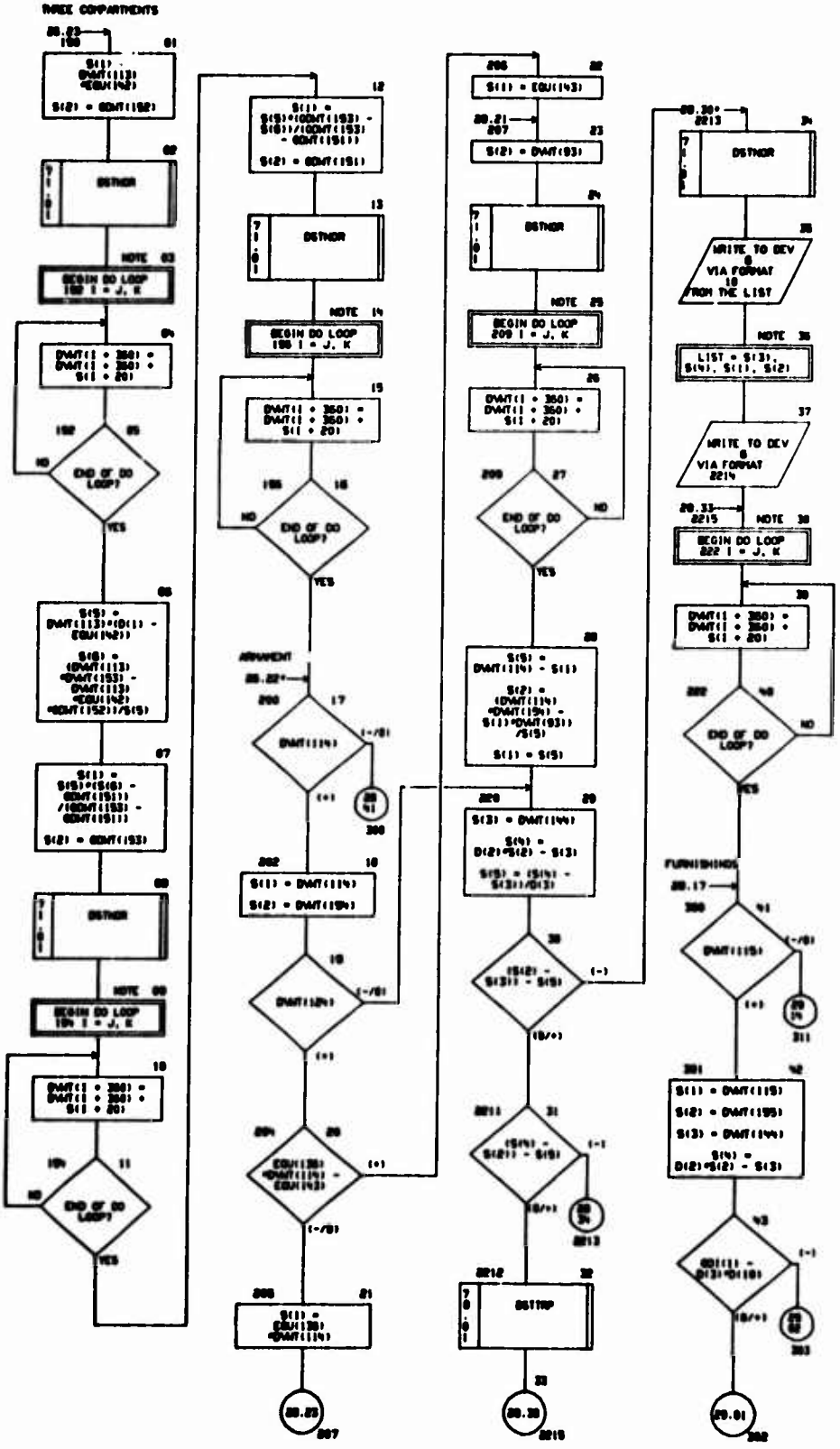


CHART TITLE - SUBROUTINE CONST

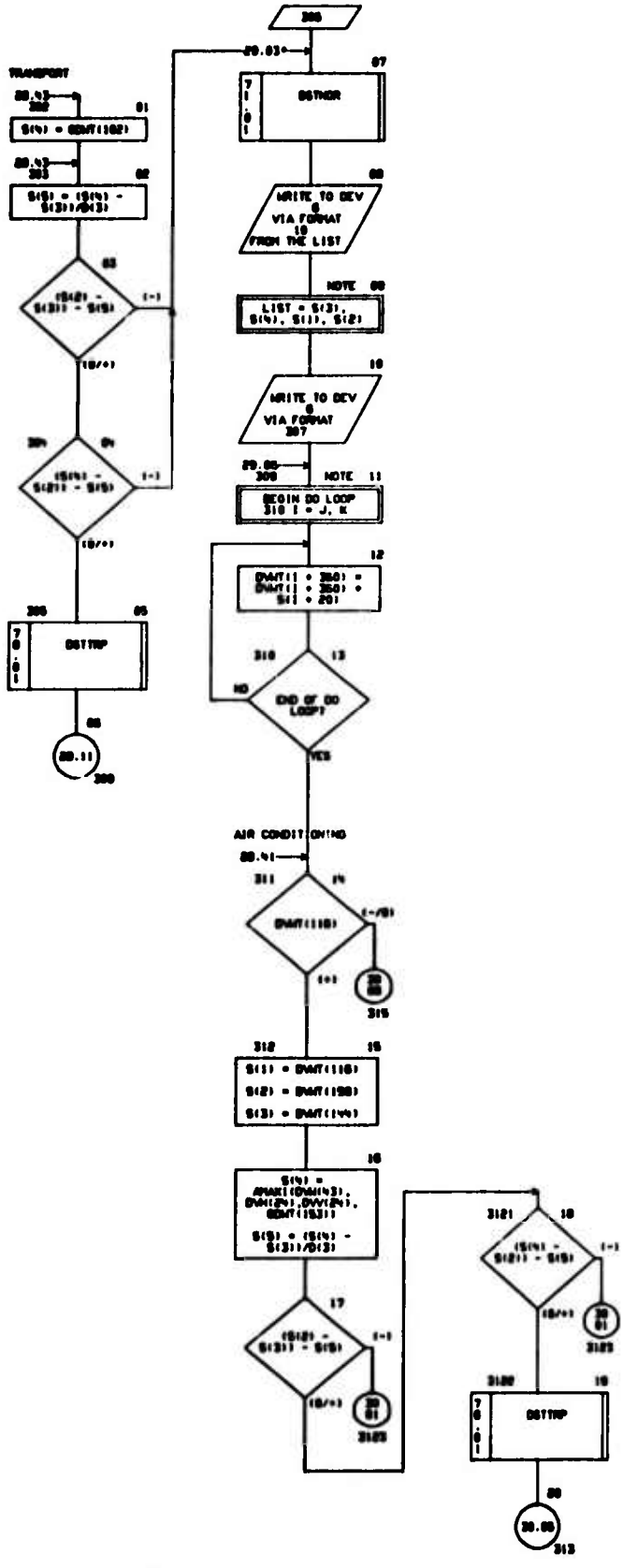


CHART TITLE - SUBROUTINE CONDST

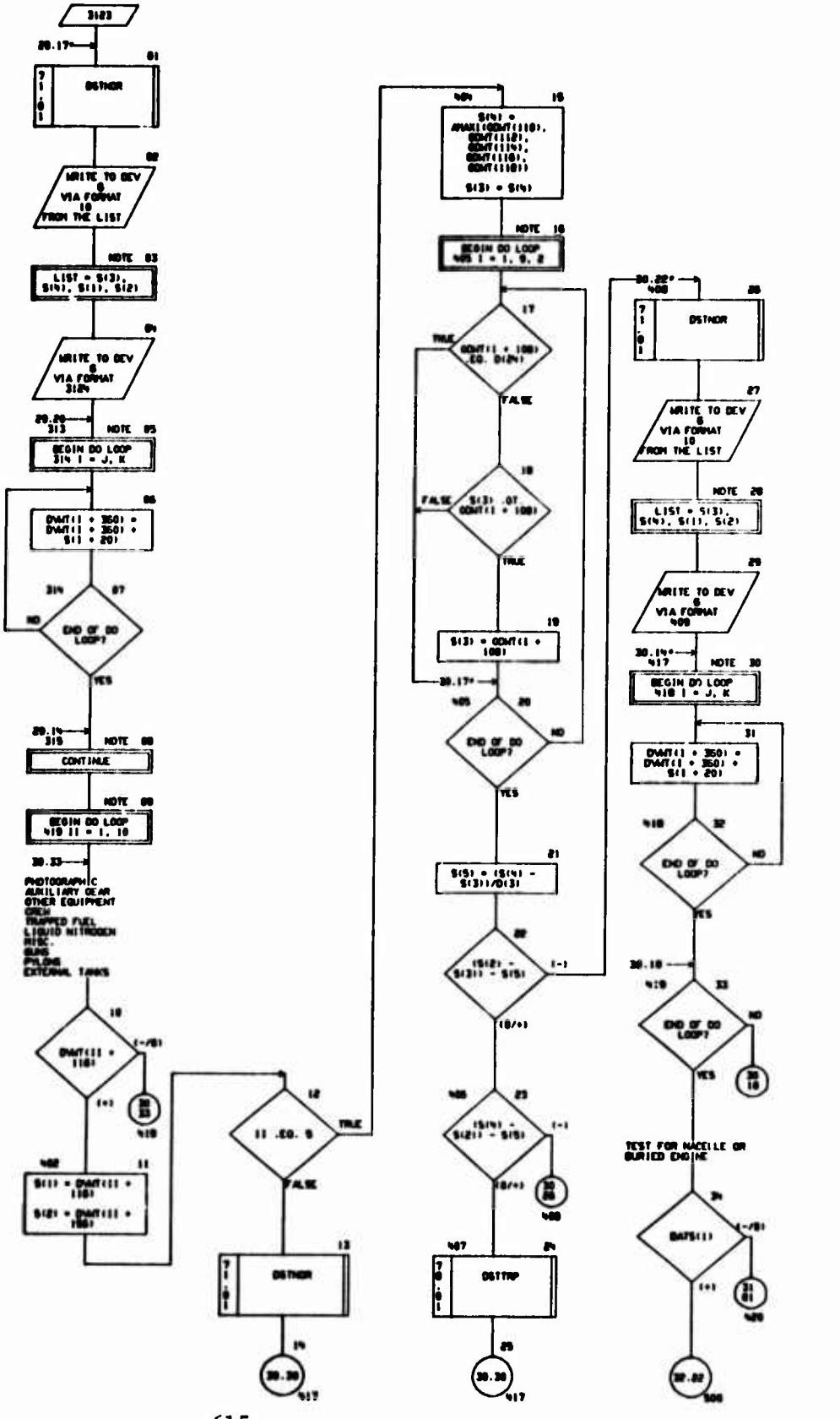


CHART TITLE - SUBROUTINE CONDST

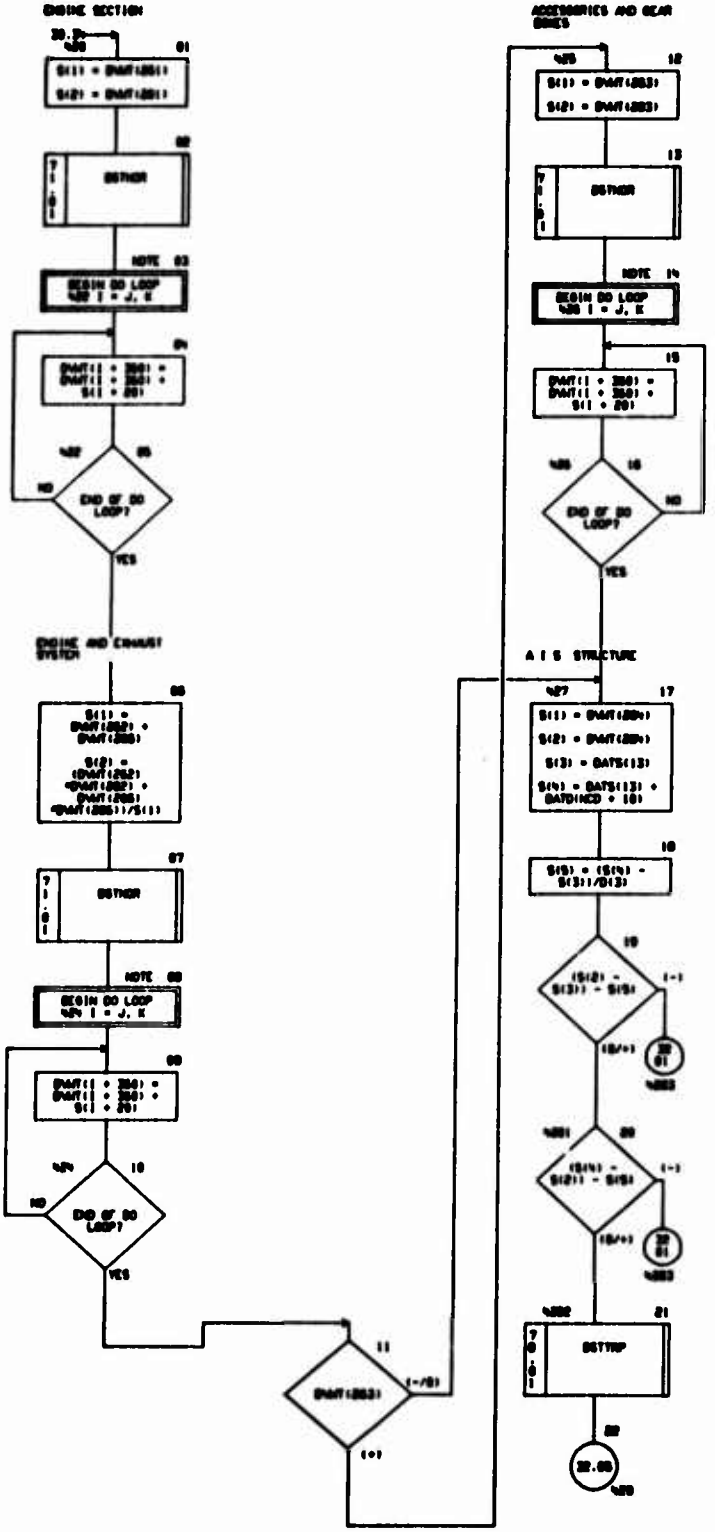


CHART TITLE - SUBROUTINE CONDST

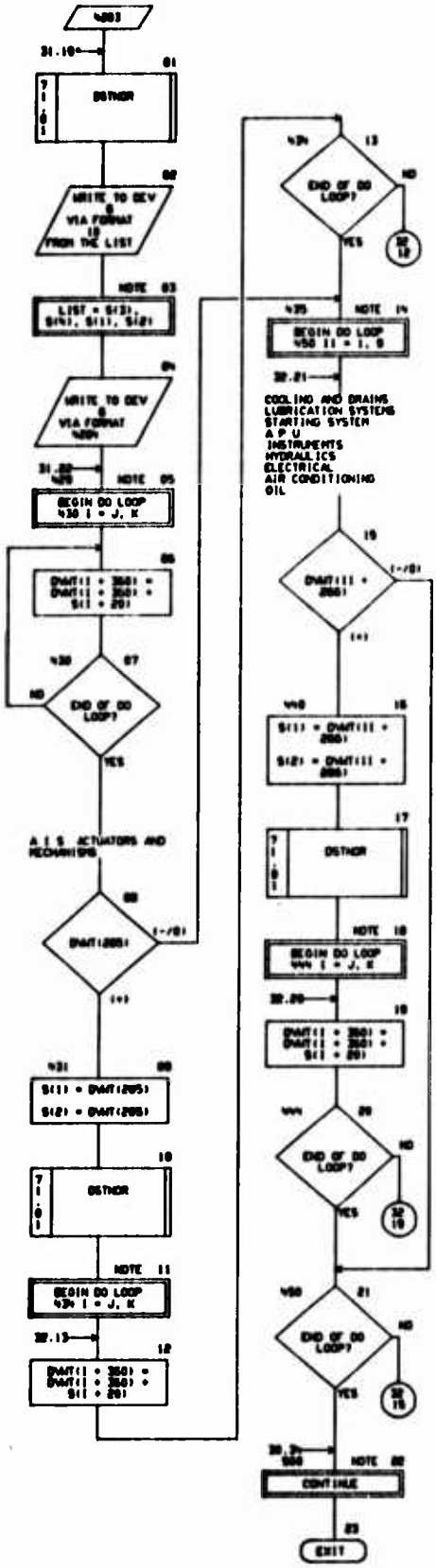


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION B(700),GD(700),DV(2200),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION GD1(20),GDAT(150),GDD(60),DATS(40),DATD(70)
      DIMENSION DWH(50),DVH(30),DVV(30),DMAT(500)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3701)),(ND(1),TCOM(4121))
      EQUIVALENCE (EQU(1),D(1))
      EQUIVALENCE (GD(1),GD(1)),(GDAT(1),GD(61)),(GDD(1),GD(121)),
     (DATS(1),GD(141)),(DATD(1),GD(150))
      EQUIVALENCE (DWH(1),DV(1)),(DWH(1),DV(31)),(DVV(1),DV(61)),
                 (DMAT(1),DV(121))
      EQUIVALENCE                (J,ND(102)),(K,ND(103)),(L,ND(104))
      EQUIVALENCE (NGD,ND(117))
10    FORMAT(20H*** WARNING FROM CONDST *** /
     20X,33HPREVIOUS FORE AND AFT LIMITS WERE, 2F10.2/
     20X,F12.2,17H LBS AT FUS. STA.,F8.2,10X,21HDISTRIBUTED BY OSTNER)
12    FORMAT(1H+,10X,18HSURFACE CONTROLS)
13    FORMAT(1H+,2X,11HFUEL SYSTEM)
14    FORMAT(1H+,20X,19HENGINE CONTROLS)
15    FORMAT(1H+,2X,11HINSTRUMENTS)
16    FORMAT(1H+,25X,10HHYDRAULICS)
17    FORMAT(1H+,25X,10HELECTRICAL)
21    FORMAT(1H+,27X,8HARMAMENT)
22    FORMAT(1H+,2X,11HFURNISHINGS)
31    FORMAT(1H+,10X,18HAIR CONDITIONING)
40    FORMAT(1H+,23X,12HTRAPPED FUEL)
45    FORMAT(1H+,20X,19HA I S STRUCTURE)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*************************************************************
C                    SUBROUTINE DBLCNT
C*************************************************************
```

CHART TITLE - SUBROUTINE SELOUT

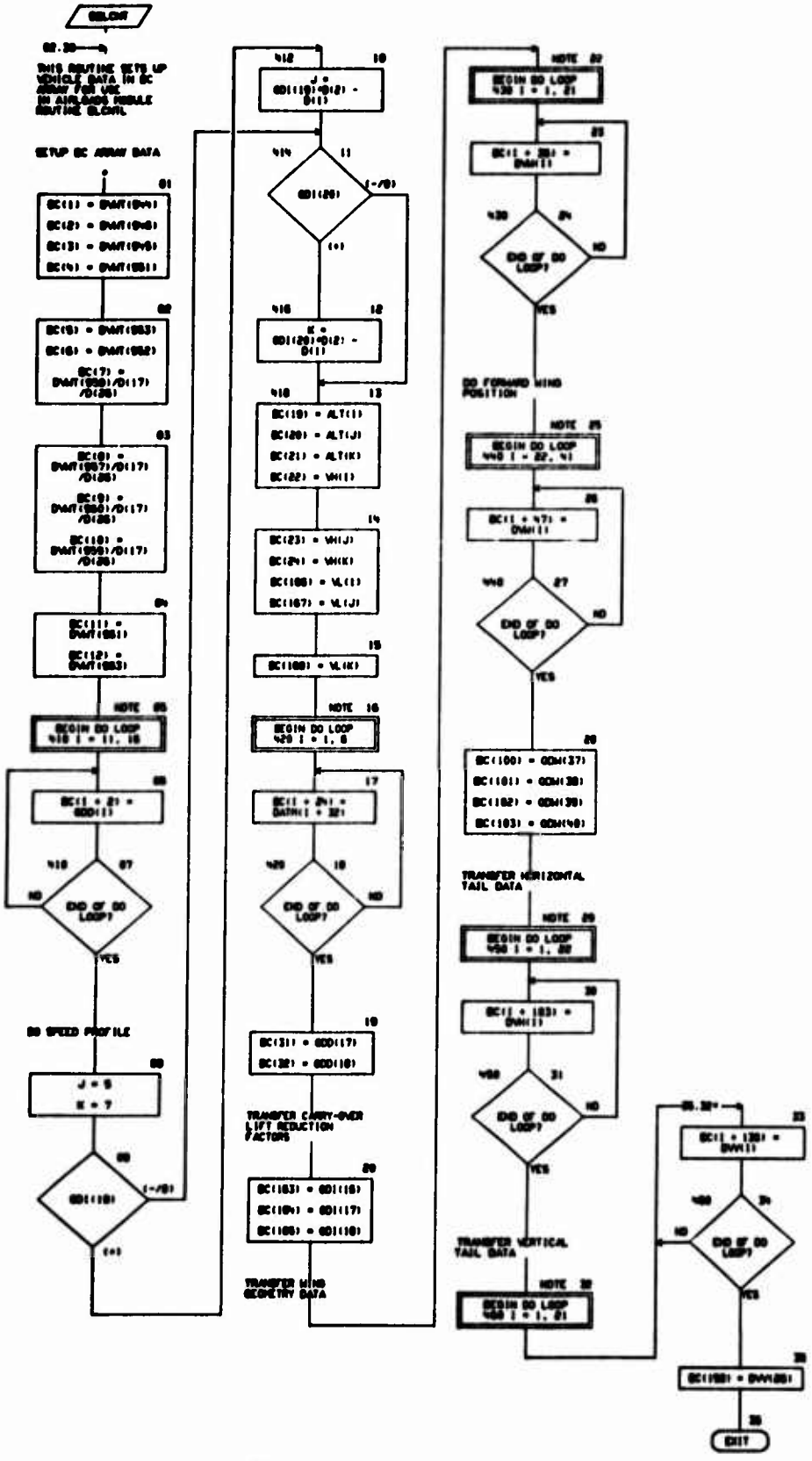


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
COMMON TCOM(4320)
DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
DIMENSION BC(200)
DIMENSION GD(20),GDD(30),DATH(40),GDM(50)
DIMENSION DVM(50),DVH(30),DVV(30),DVAT(1000),ALT(10),VH(10),
 VL(10)
EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401))
 , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
EQUIVALENCE (GD(1),GD(1)),(GDD(1),GD(21)),(DATH(1),GD(51))
 , (GDM(1),GD(129))
EQUIVALENCE (DVM(1),DV(32)),(DVH(1),DV(37)),(DVV(1),DV(49))
 , (DVAT(1),DV(1121)),(ALT(1),DV(1)),(VH(1),DV(6))
 , (VL(1),DV(7))
EQUIVALENCE (BC(1),DV(2121))
```

```
CHART TITLE - INTRODUCTORY COMMENTS


                    ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
                                              SUBROUTINE BCCNTL
                    ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
```

CHART TITLE - SUBROUTINE SCONTL

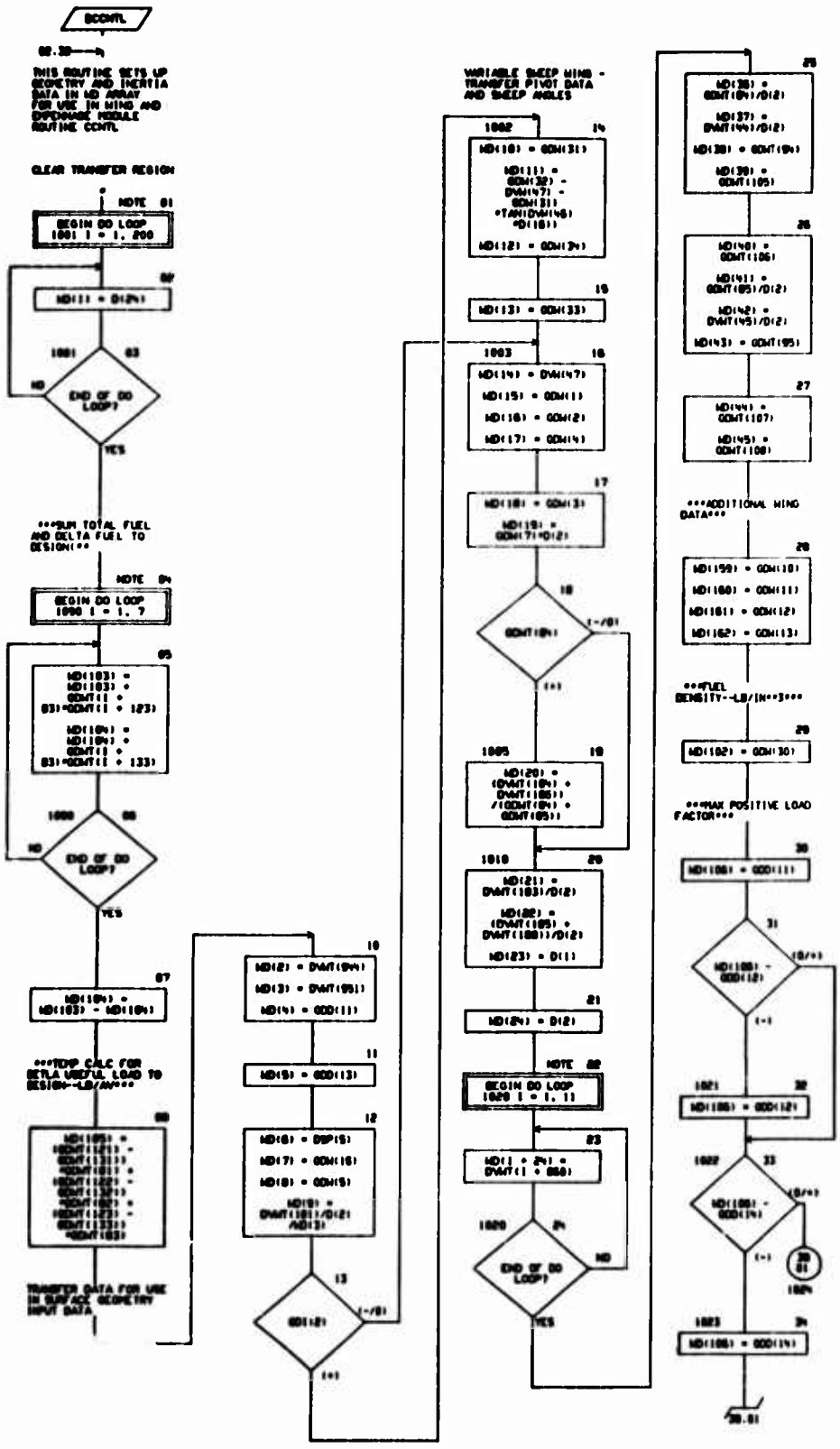


CHART TITLE - SUBROUTINE SCONTL

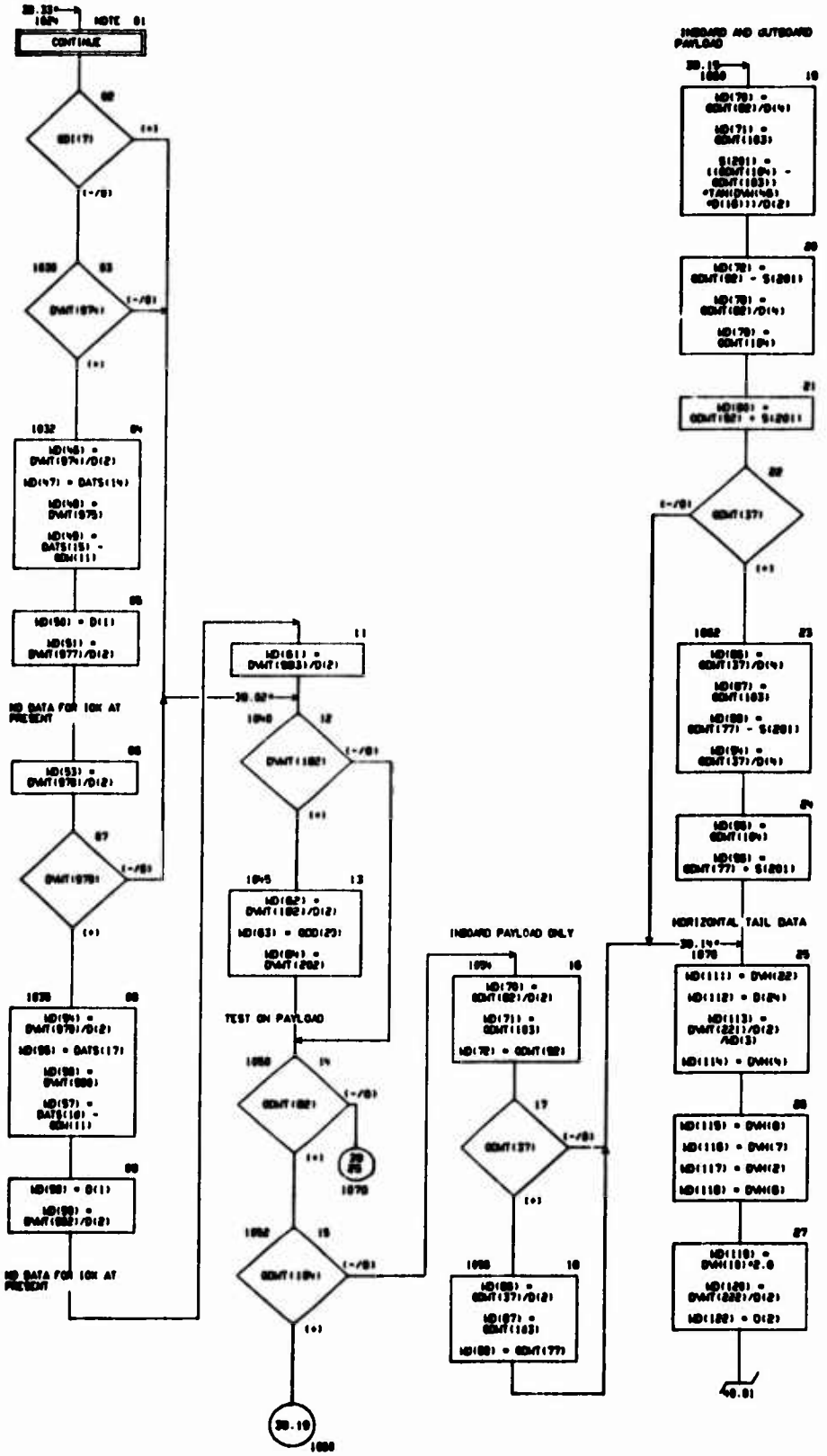


CHART TITLE - SUBROUTINE SCONTL

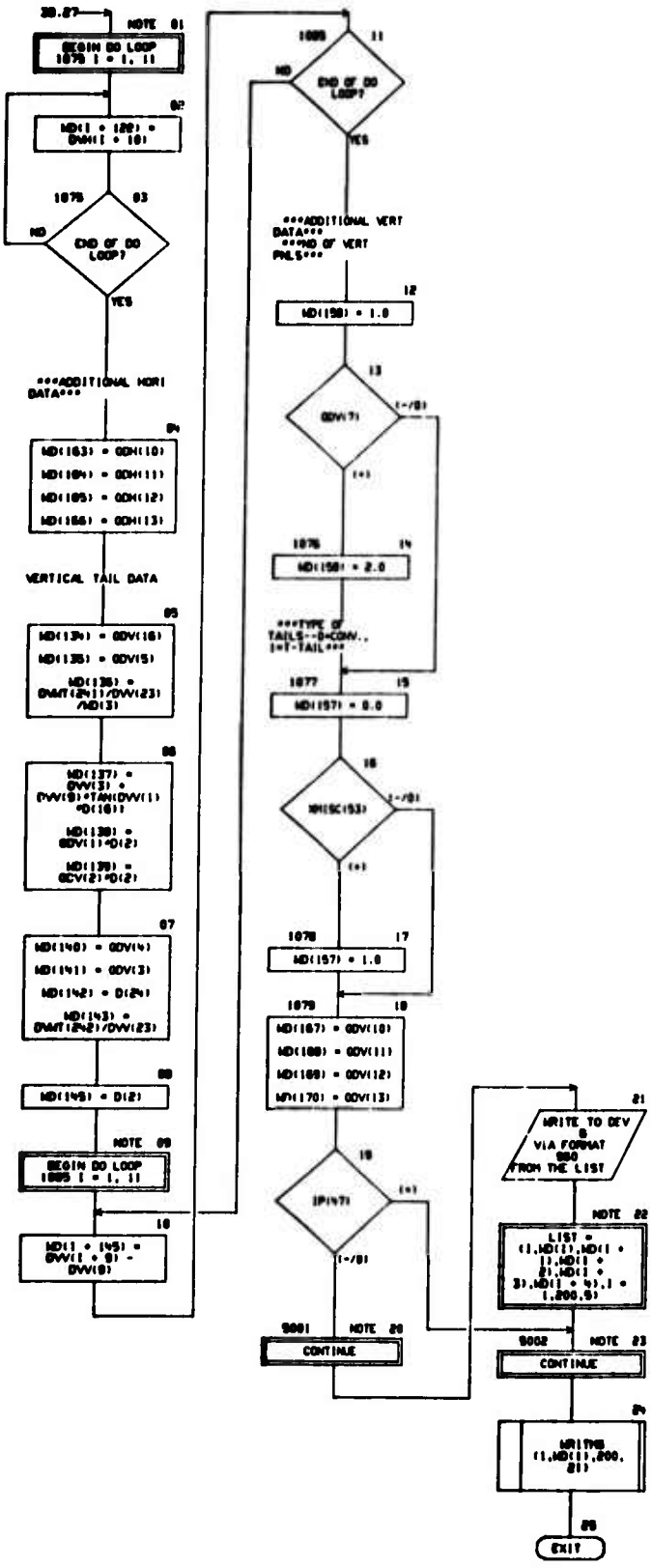


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /MISC/ MMISC(100)
      COMMON /PRINT/ IP(80)
      DIMENSION D(700),GD(700),DV(2320),S(4000),ND(200)
      DIMENSION GD1(20)
      DIMENSION GDD(30),GDMT(160),GDM(50),GDM(40),GDV(40),DATS(40)
      DIMENSION DVM(50),DVM(30),DVV(30),DVMT(1000)
      DIMENSION DSP(10)
      DIMENSION ND(200)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401))
     , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GD1(1),GD(1))
      EQUIVALENCE (GDD(1),GD(21)),(GDMT(1),GD(91)),(GDM(1),GD(25))
     , (GDM(1),GD(36)),(GDV(1),GD(34)),(DATS(1),GD(40))
      EQUIVALENCE (DVM(1),DV(32)),(DVM(1),DV(37)),(DVV(1),DV(40))
     , (DVMT(1),DV(112))
      EQUIVALENCE (DSP(1),DV(3))
      EQUIVALENCE (ND(1),S(1))
  888 FORMAT(1SH1 ND(1) FROM DCCNTL,70X,21H** DCCNTL - IP(47) **/
     (110, 9F10.4))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*************************************************************
C                  SUBROUTINE DFATMG
C*************************************************************
```

CHART TITLE - SUBROUTINE DFATHB

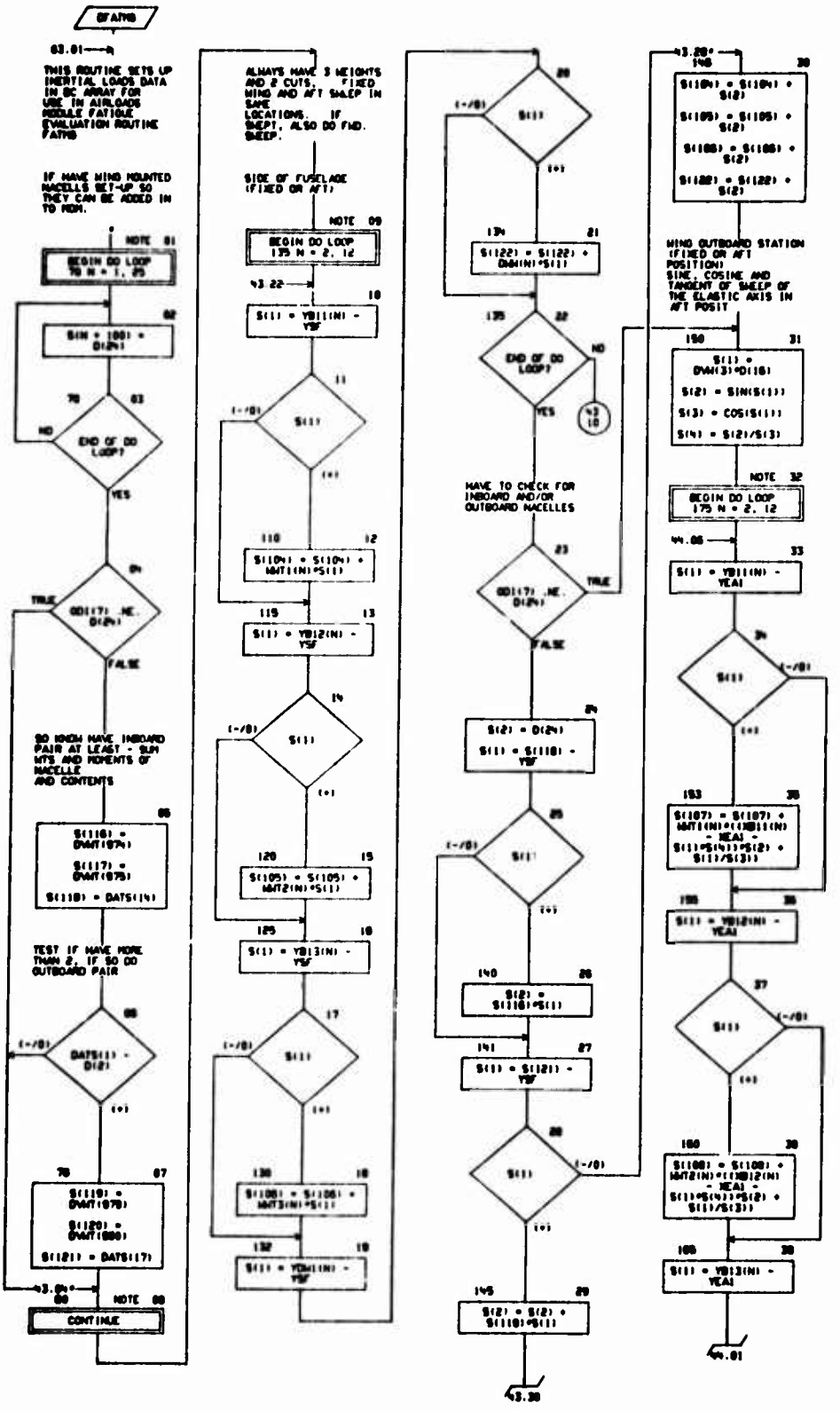


CHART TITLE - SUBROUTINE SWEEP

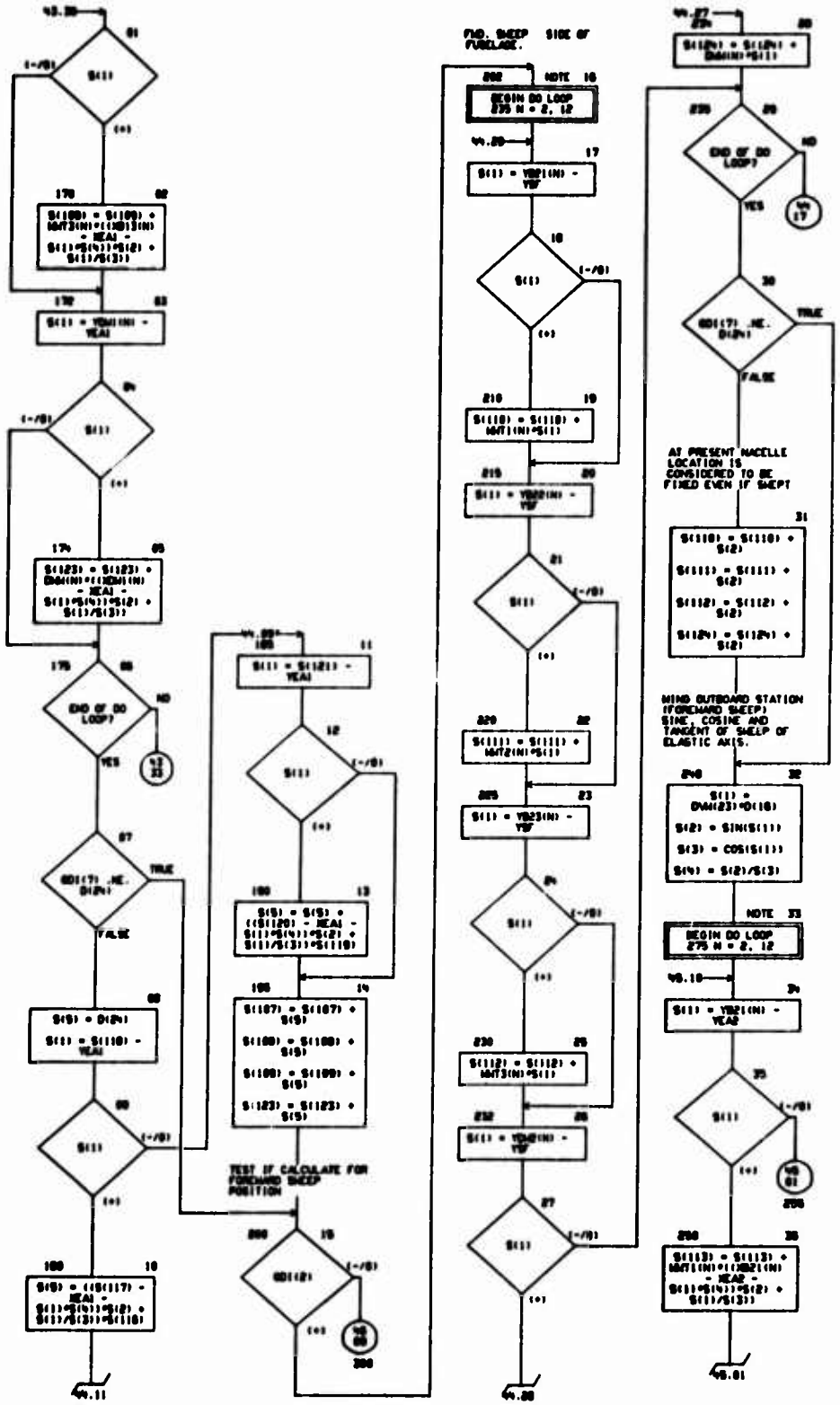


CHART TITLE - SUBROUTINE STATUS

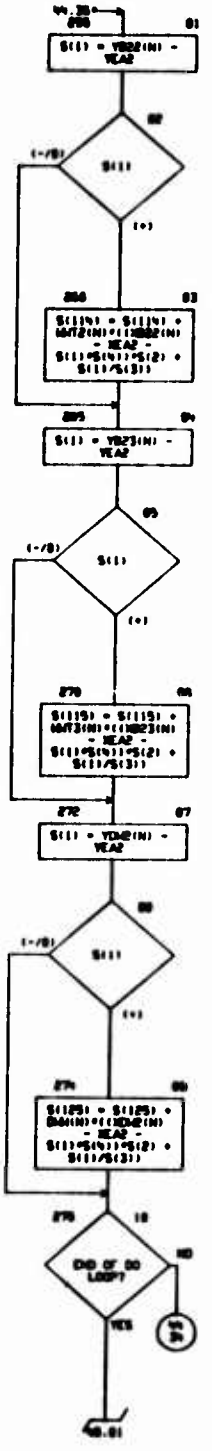


CHART TITLE - SUBROUTINE STATUS

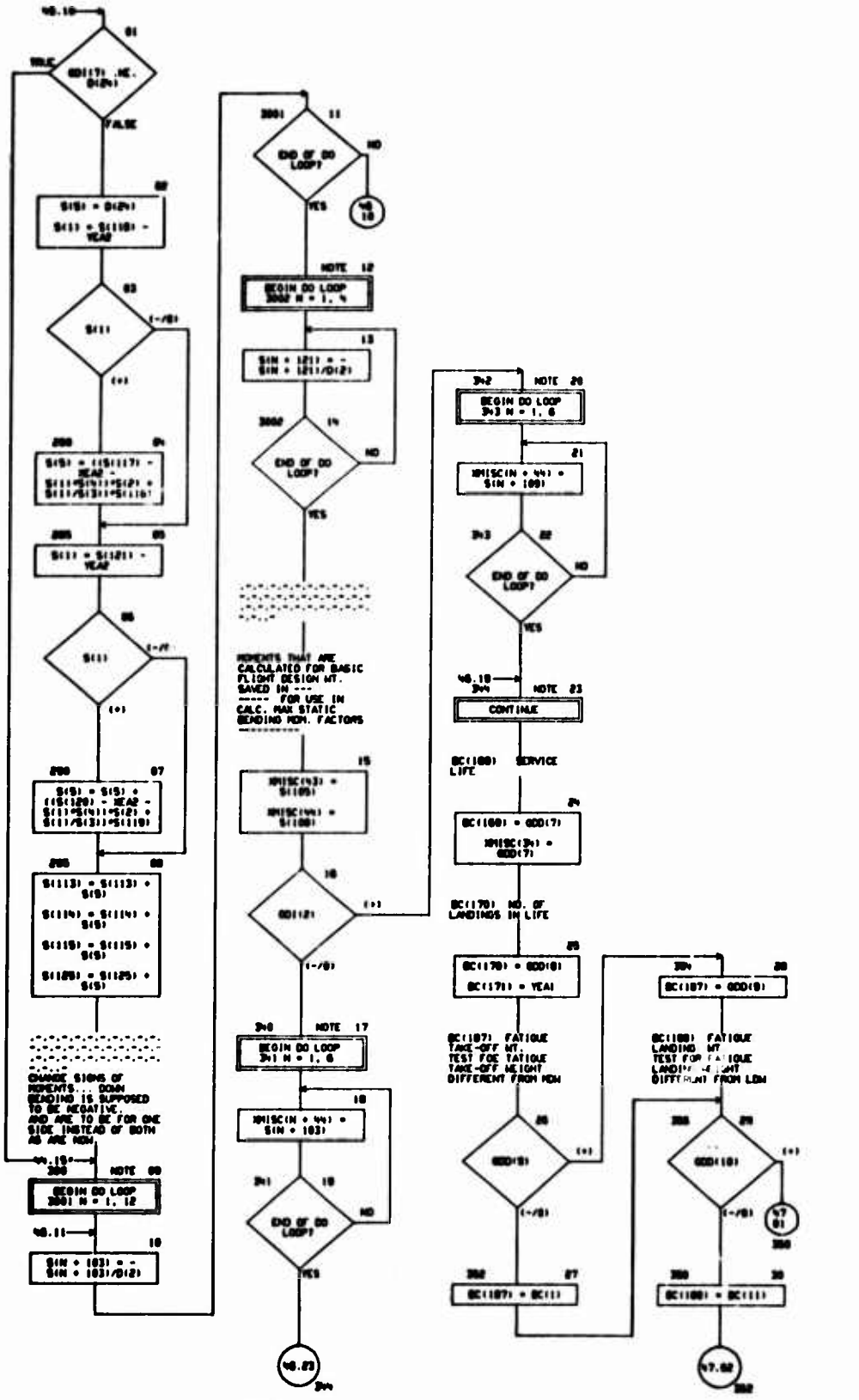


CHART TITLE - SUBROUTINE STATUS

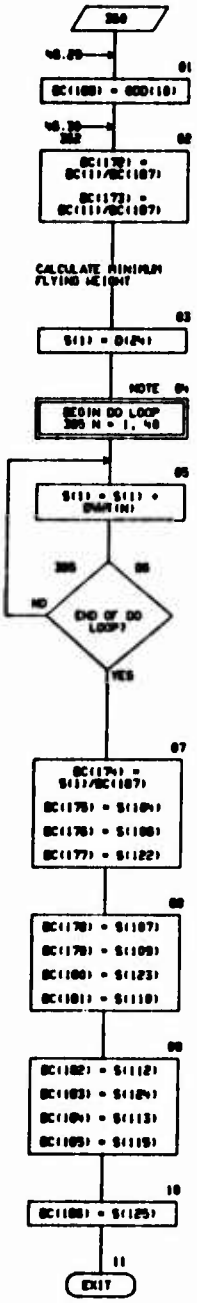


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /MISC/ XMISC(100)
      DIMENSION B(700),GD(700),DV(2320),S(400),ND(200)
     , DVAT(1000)
     , GD1(20), DATS(40), DVM(50), BC(200), MAT(12), MAT2(12)
     , MAT3(12), XB11(12), XB21(12), YB11(12), YB21(12), XB12(12)
     , XB22(12), YB12(12), YB22(12), XB13(12), XB23(12), YB13(12)
     , YB23(12), GDD(30)
      DIMENSION DAM(12),XDM1(12),XDM2(12),YDM1(12),YDM2(12)
      EQUIVALENCE (TCOM(1),B(1)), (TCOM(701),GD(1)), (TCOM(1401),DV(1))
     , (TCOM(3721),S(1)), (TCOM(4121),ND(1))
     , (DV(321),DVM(1)), (DV(1121),DVAT(1)), (DV(2121),BC(1))
     , (GD(1),GD1(1)), (GD(21),GDD(1)), (GD(461),DATS(1))
     , (YEAR,DVAT(309)), (XEAR,DVAT(395))
     , (YEAR,DVAT(308)), (XEAR,DVAT(307))
     , (YBF ,DVM(1))
      EQUIVALENCE           (MAT(1),DVAT(487)), (MAT2(1),DVAT(52))
     , (MAT3(1),DVAT(945)), (XB11(1),DVAT(568)), (XB21(1),DVAT(58))
     , (YB11(1),DVAT(583)), (YB21(1),DVAT(605)), (XB12(1),DVAT(617))
     , (XB22(1),DVAT(629)), (YB12(1),DVAT(641)), (YB22(1),DVAT(653))
     , (XB13(1),DVAT(665)), (XB23(1),DVAT(677)), (YB13(1),DVAT(689))
     , (YB23(1),DVAT(701))
      EQUIVALENCE (DAM(1),DVAT(872))
     , (XDM1(1),DVAT(896)),(XDM2(1),DVAT(908))
     , (YDM1(1),DVAT(920)),(YDM2(1),DVAT(932))
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE SLADOR

CHART TITLE - SUBROUTINE BLADER

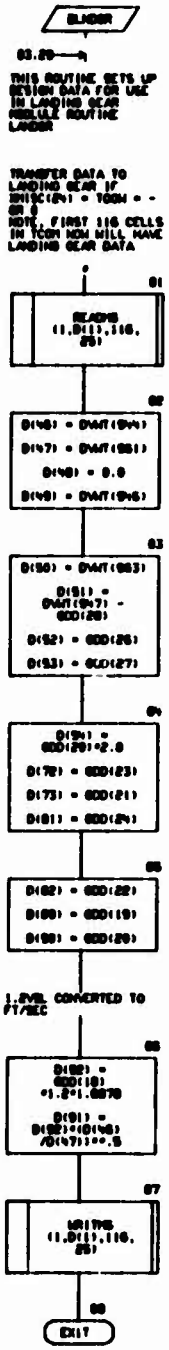


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),GD(700),DV(2320),S1(400),ND(200)
      DIMENSION GDD(30),DVAT(1000)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401))
     . (S1(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GDD(1),GD(21)),(DVAT(1),DV(1121))
```

```
CHART TITLE - INTRODUCTORY COMMENTS



                  ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
                                        SUBROUTINE DMHLD
                  ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
```

CHART TITLE - SUBROUTINE DWNLD

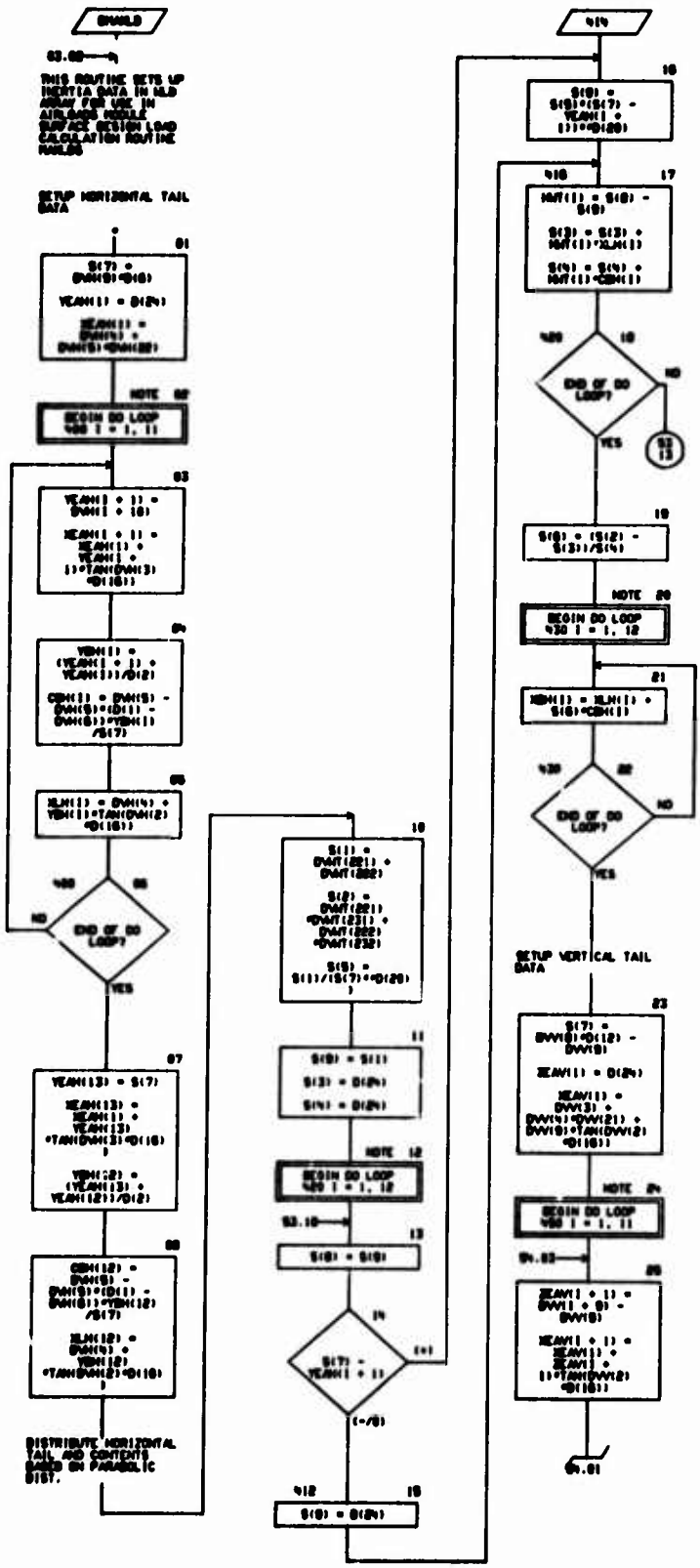


CHART TITLE - SUBROUTINE SWAILS

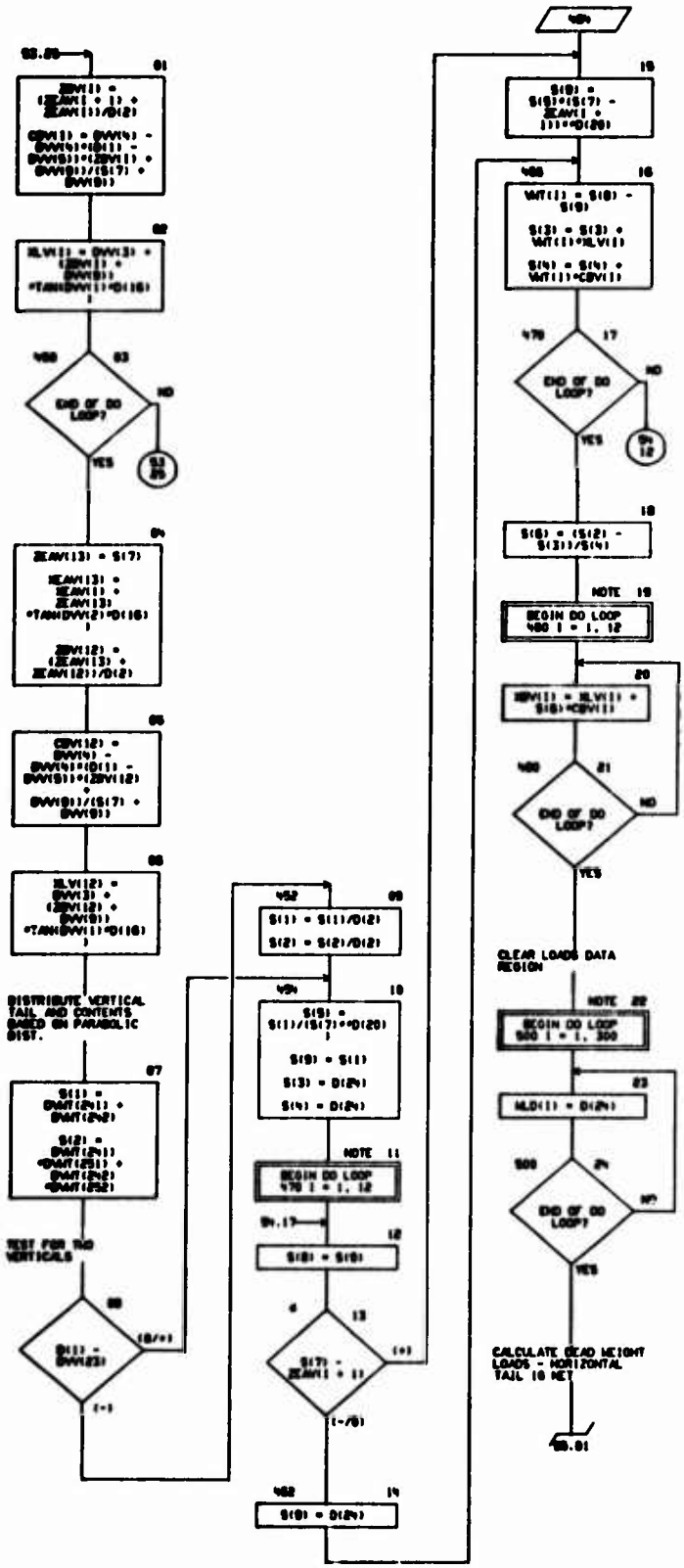


CHART TITLE - SUBROUTINE SWAILS

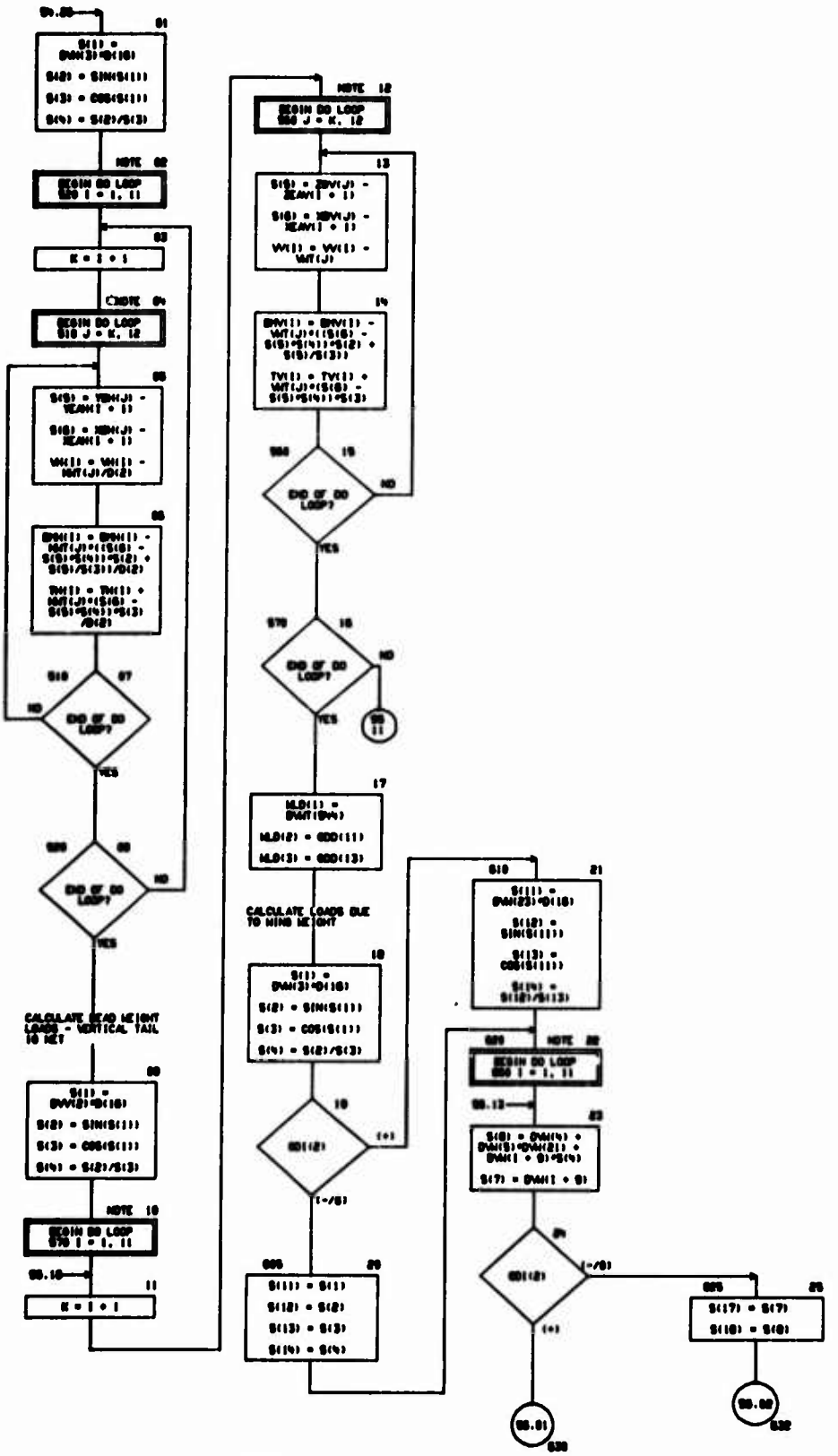


CHART TITLE - SUBROUTINE DWALD

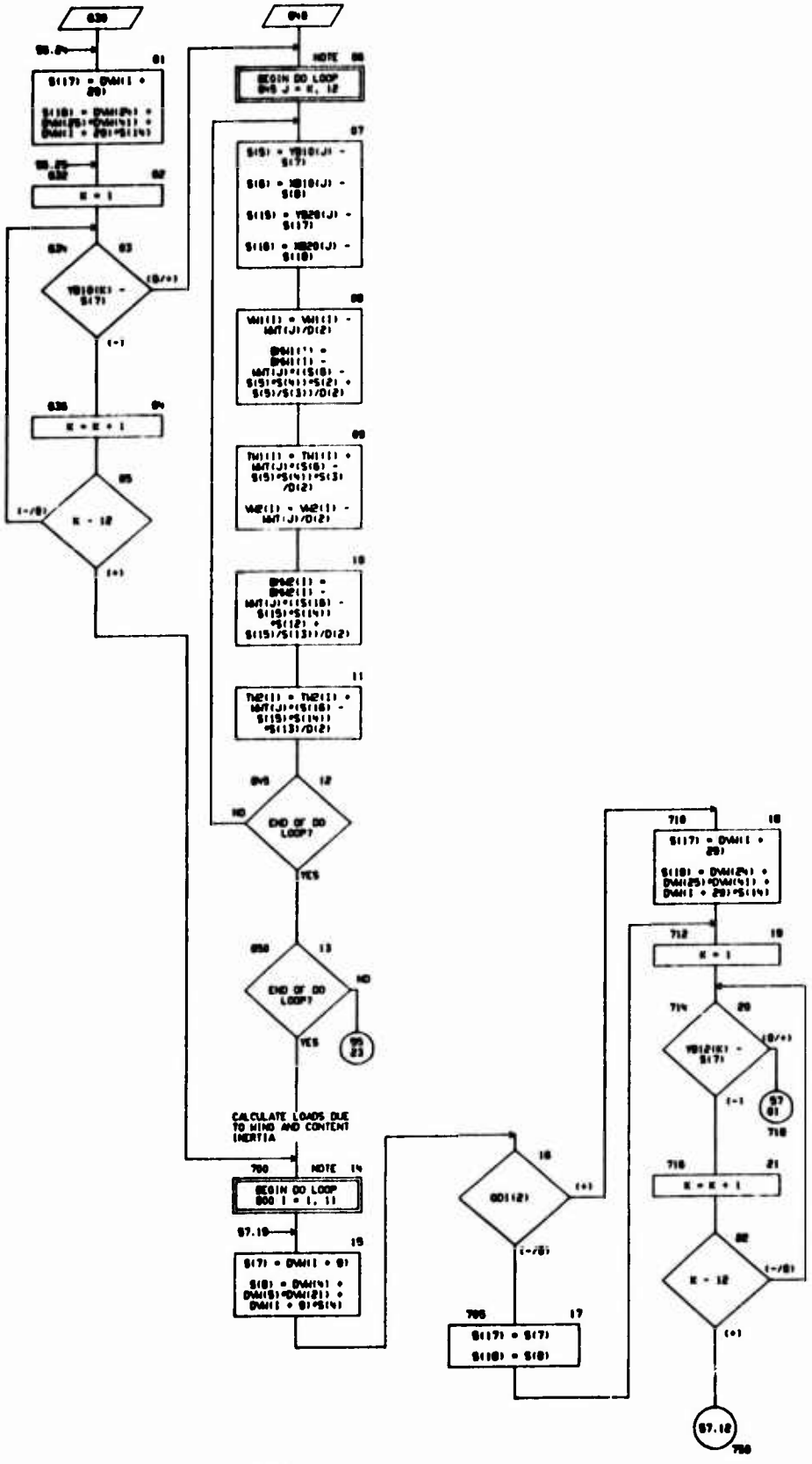


CHART TITLE - SUBROUTINE SHAKLD

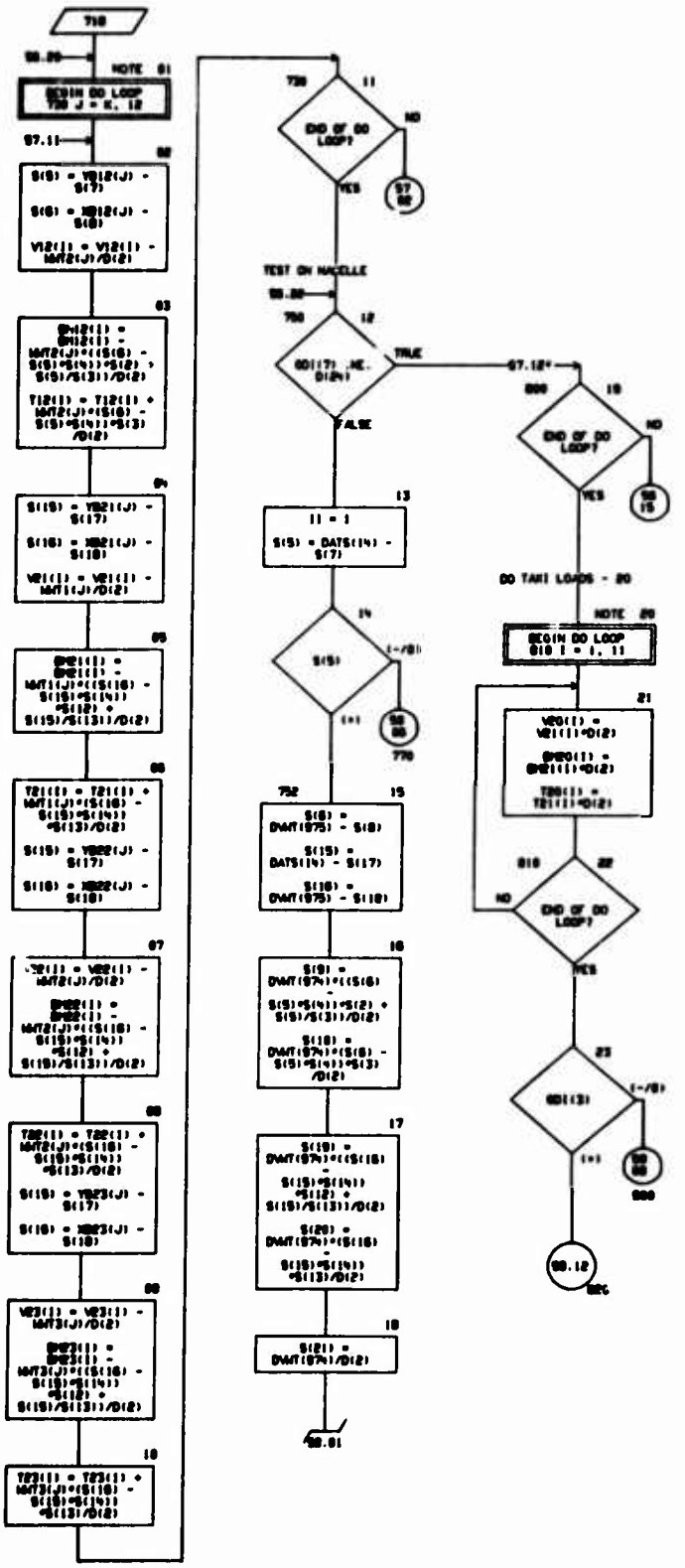


CHART TITLE - SUBROUTINE CHAOLD

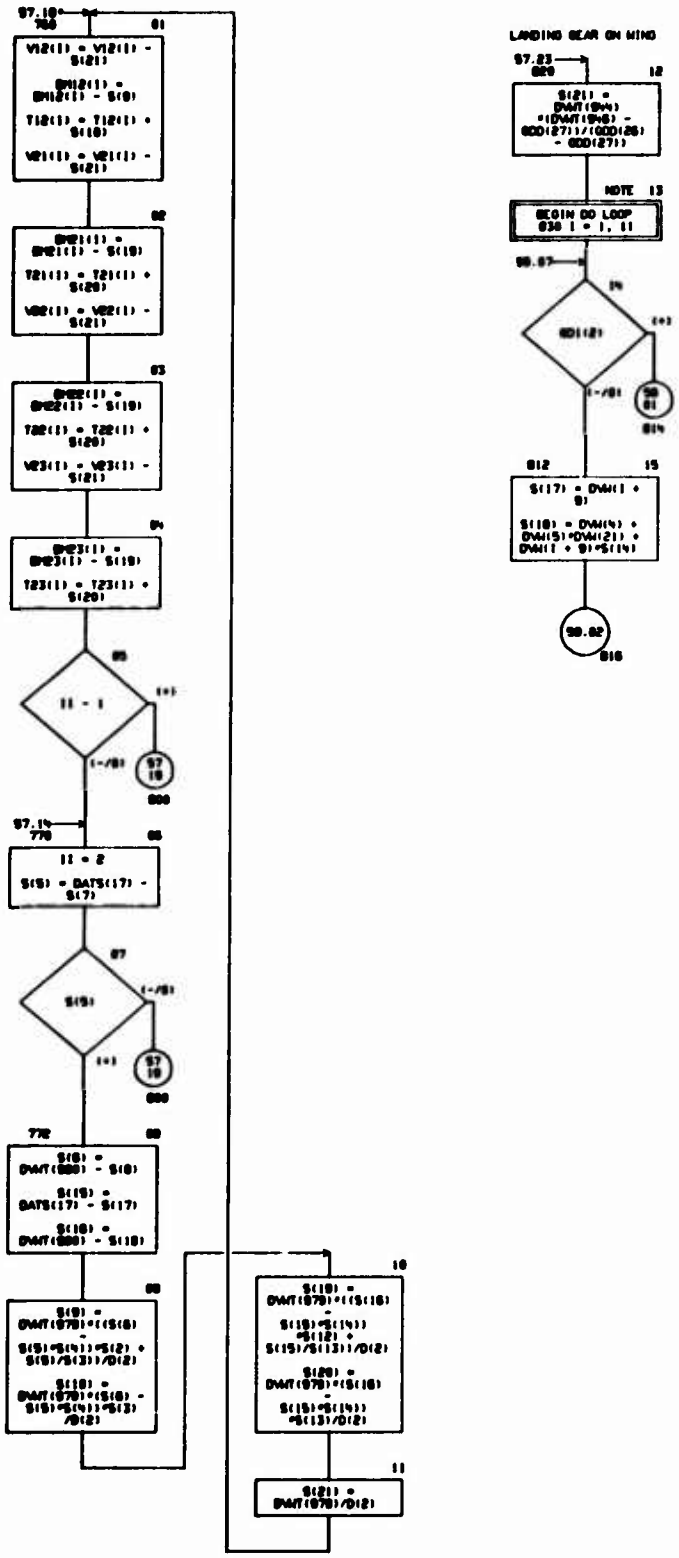


CHART TITLE - SUBROUTINE SWAMLD

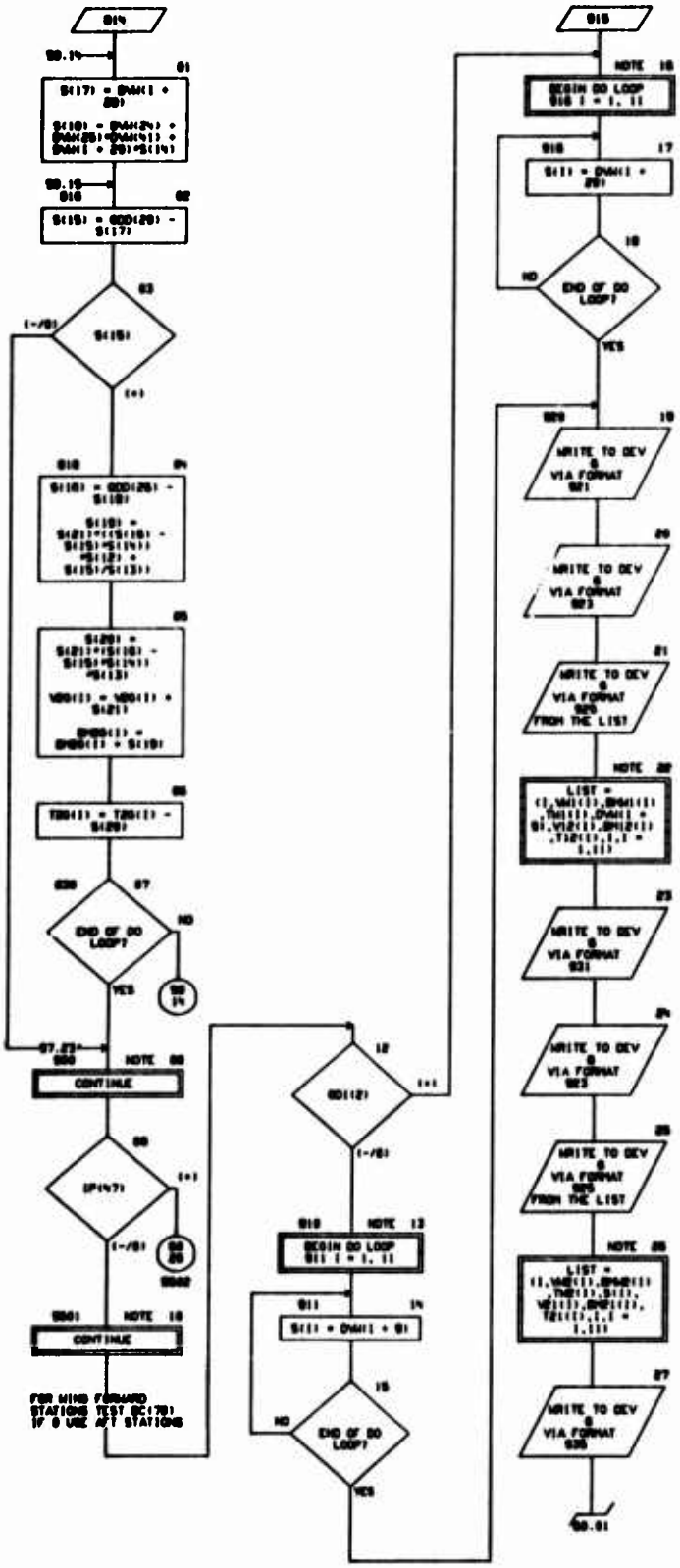


CHART TITLE - SUBROUTINE SHAILO

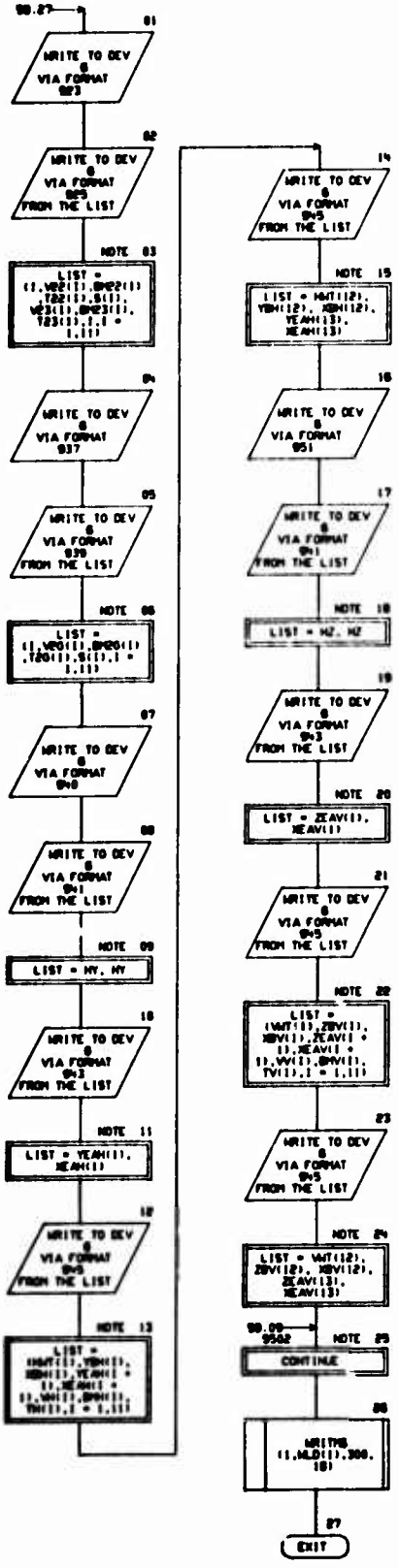


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION GD1(20),GDD(30),DATS(40)
      DIMENSION DVM(50),DVM(30),DVV(30),DVMT(1000)
      DIMENSION YB10(12),XB10(12),YB20(12),XB20(12),WMT(12)
      DIMENSION WMT1(12),WMT2(12),WMT3(12),XB11(12),XB21(12),YB11(12)
     , YB21(12),XB12(12),XB22(12),YB12(12),YB22(12),XB13(12)
     , XB23(12),YB13(12),YB23(12)
      DIMENSION WMT(12),YBH(12),XBH(12),WMT(12),ZBV(12),XBV(12)
      DIMENSION YEAH(13),XEAH(13),ZEAV(13),XEAV(13)
     , CBH(12),XLH(12),CBV(12),XLV(12)
      DIMENSION WLD(300)
      DIMENSION V00(11),BM00(11),T20(11),VW1(11),BW1(11),TW1(11)
     , V02(11),BW2(11),TW2(11),V21(11),BW21(11),T21(11)
     , V12(11),BW12(11),T12(11),V22(11),BW22(11),T22(11)
     , V23(11),BW23(11),T23(11),VW(11),BPH(11),TH(11)
     , VV(11),BWV(11),TV(11)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401))
     , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GD1(1),GD(1)),(GDD(1),GD(21)),(DATS(1),GD(49))
      EQUIVALENCE (DVM(1),DV(1321)),(DVM(1),DV(1371)),(DVV(1),DV(1401))
     , (DVMT(1),DV(1121))
      EQUIVALENCE (YB10(1),DVMT(401)),(XB20(1),DVMT(413))
     , (YB20(1),DVMT(425)),(WMT(1),DVMT(473)),(XB10(1),DVMT(485))
      EQUIVALENCE            (WMT1(1),DVMT(497)), (WMT2(1),DVMT(521))
     , (WMT3(1),DVMT(545)), (XB11(1),DVMT(569)), (XB21(1),DVMT(581))
     , (YB11(1),DVMT(593)), (YB21(1),DVMT(605)), (XB12(1),DVMT(617))
     , (XB22(1),DVMT(629)), (YB12(1),DVMT(641)), (YB22(1),DVMT(653))
     , (XB13(1),DVMT(665)), (XB23(1),DVMT(677)), (YB13(1),DVMT(689))
     , (YB23(1),DVMT(701))
      EQUIVALENCE (WMT(1),DVMT(872)),(YBH(1),DVMT(884))
     , (XBH(1),DVMT(896)),(WMT(1),DVMT(908)),(ZBV(1),DVMT(920))
     , (XBV(1),DVMT(932))
      EQUIVALENCE (YEAH(1),S(49)),(XEAH(1),S(62)),(ZEAV(1),S(75))
     , (XEAV(1),S(88))
      EQUIVALENCE (WLD(1),S(101))
      EQUIVALENCE (V00(1),WLD(1)),(BM00(1),WLD(5)),(T20(1),WLD(26))
     , (VW1(1),WLD(37)),(BW1(1),WLD(48)),(TW1(1),WLD(59))
     , (V02(1),WLD(70)),(BW2(1),WLD(81)),(TW2(1),WLD(92))
     , (V21(1),WLD(103)),(BP2(1),WLD(114)),(T21(1),WLD(125))
     , (V12(1),WLD(136)),(BW12(1),WLD(147)),(T12(1),WLD(158))
     , (V22(1),WLD(169)),(BW22(1),WLD(180)),(T22(1),WLD(191))
     , (V23(1),WLD(202)),(BW23(1),WLD(213)),(T23(1),WLD(224))
     , (VW(1),WLD(235)),(BPW(1),WLD(246)),(TW(1),WLD(257))
     , (VV(1),WLD(268)),(BWV(1),WLD(279)),(TV(1),WLD(290))
      EQUIVALENCE (CBH(1),WLD(1)),(XLH(1),WLD(13))
     , (CBV(1),WLD(25)),(XLV(1),WLD(37))
      DATA HY/1HY /, HZ/1HZ/
  821 FORMAT(1HI,49X,24HBENDING MOMENT AND TORQUE,2HX,
     1  21H** SHAILD - IP(47) **,              // 19X,19HHING ONLY AT
     1 10, 33X, 23HHING AND CONTENTS AT 10// 20X, 18HWT POSITION, 33X,
     2 27HWT POSITION GROSS WEIGHT 2 )
  823 FORMAT(1H0  I, 9X,3HBNEAR,6X,6HMOMENT,6X,6HTORQUE,6X,18HBUTT PLANE
     111X,3HBNEAR,6X,6HMOMENT,6X,6HTORQUE, 9X,1HI /)
  825 FORMAT(1H,3F12.0,6X,1F6.1,6X,3F12.0,6X,I3 )
  831 FORMAT(//1H0, 19X,19HHING ONLY AT 10,33X,23HHING AND CONTENTS AT 10
     //19X,18HFORWARD POSITION, 20X, 3HFORWARD POSITION AT GROSS WEIGH
     1 T 1 )
  835 FORMAT(1HI,69X,21H** SHAILD - IP(47) **/
                 19X, 23HHING AND CONTENTS AT 10, 35X, 23HHING AND CONTE
     NTS AT 10 // 9X, 3HFORWARD POSITION AT GROSS WEIGHT 2, 19X,
     3HFORWARD POSITION AT GROSS WEIGHT 3 )
```

CHART TITLE - NON-PROCEDURAL STATEMENTS

```
937   FORMAT(//1HO,40X, 25HNET WING LOADS AT 20 TAXI// 35X, 24HFORWARD PO
      SITION AT GROSS WEIGHT I // 28X, 1HI,5X,9HSHEAR,6X,8HMOMENT, 6X,
       8HTORQUE, 6X,18HBUTT PLANE / )
938   FORMAT(26X, 1I4, 3F12.0, 6X, 1F8.1 )
940   FORMAT(1HI, 38X, 20HHORIZONTAL TAIL AND CONTENTS ,
        82X,21H** DMAXLD - IP(47) **////)
941   FORMAT(7X,19HCOORD. OF E. A., 9X,7HSECTION, 5X, 11HCOORDINATES /
       11X, 1A1, 8X, 1HX, 8X, 5HSHEAR, 8X, 8HMOMENT, 8X, 8HTORQUE, 14X,
       8HWEIGHT, 5X, 1A1, 4H BAR, 5X, 5HX BAR /)
943   FORMAT( 4X,2F10.2)
945   FORMAT(70X,1F14.2,2F10.2/ 4X, 2F10.2, 3F14.0 )
951   FORMAT(1HI,38X, 20HVERTICAL TAIL AND CONTENTS ,
        23X,21H** DMAXLD - IP(47) **////)
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE SWEEP

CHART TITLE - SUBROUTINE DSGNPR

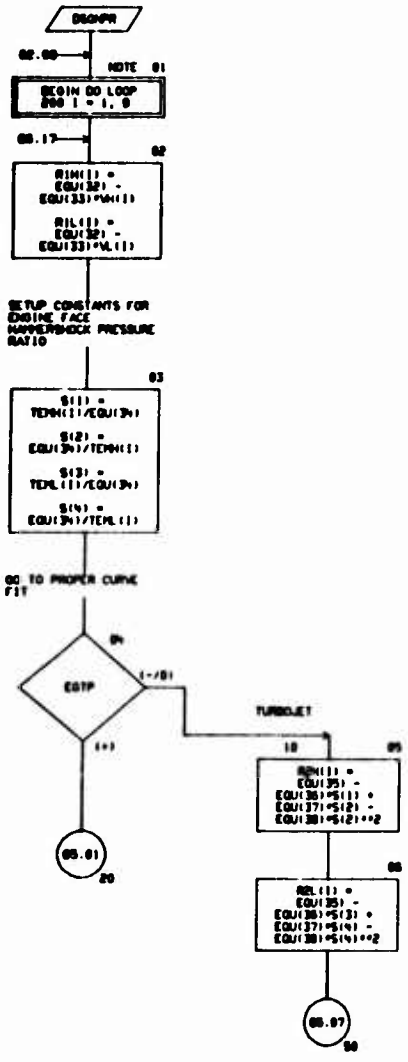


CHART TITLE - SUBROUTINE SWEEPR

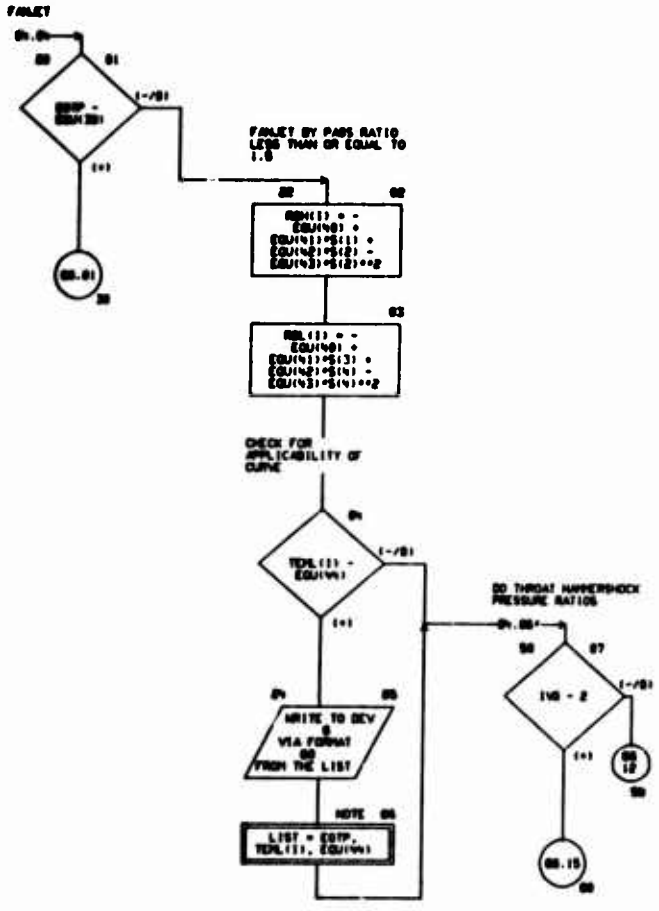


CHART TITLE - SUBROUTINE SKEDPR

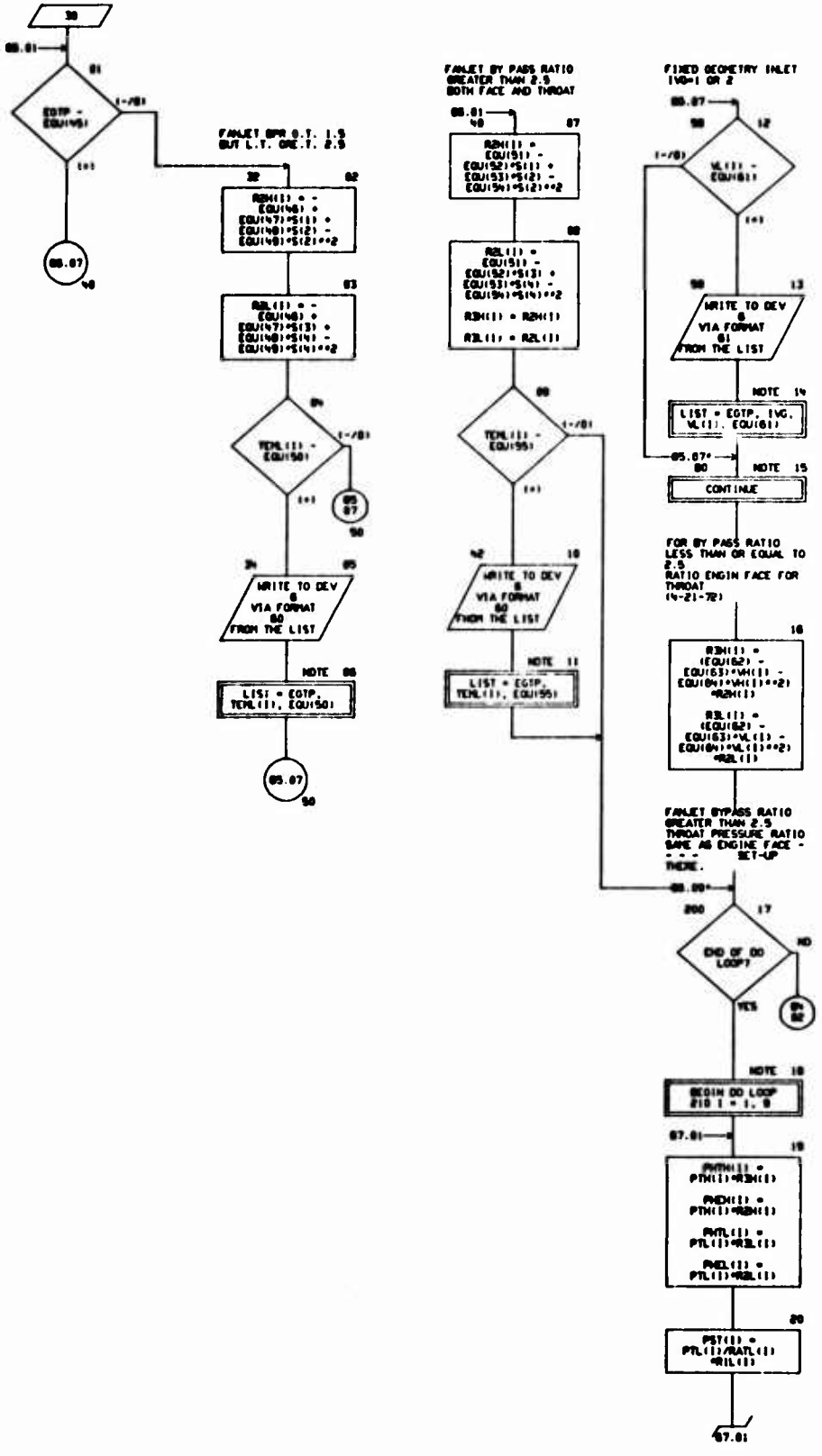


CHART TITLE - SUBROUTINE DSDPR

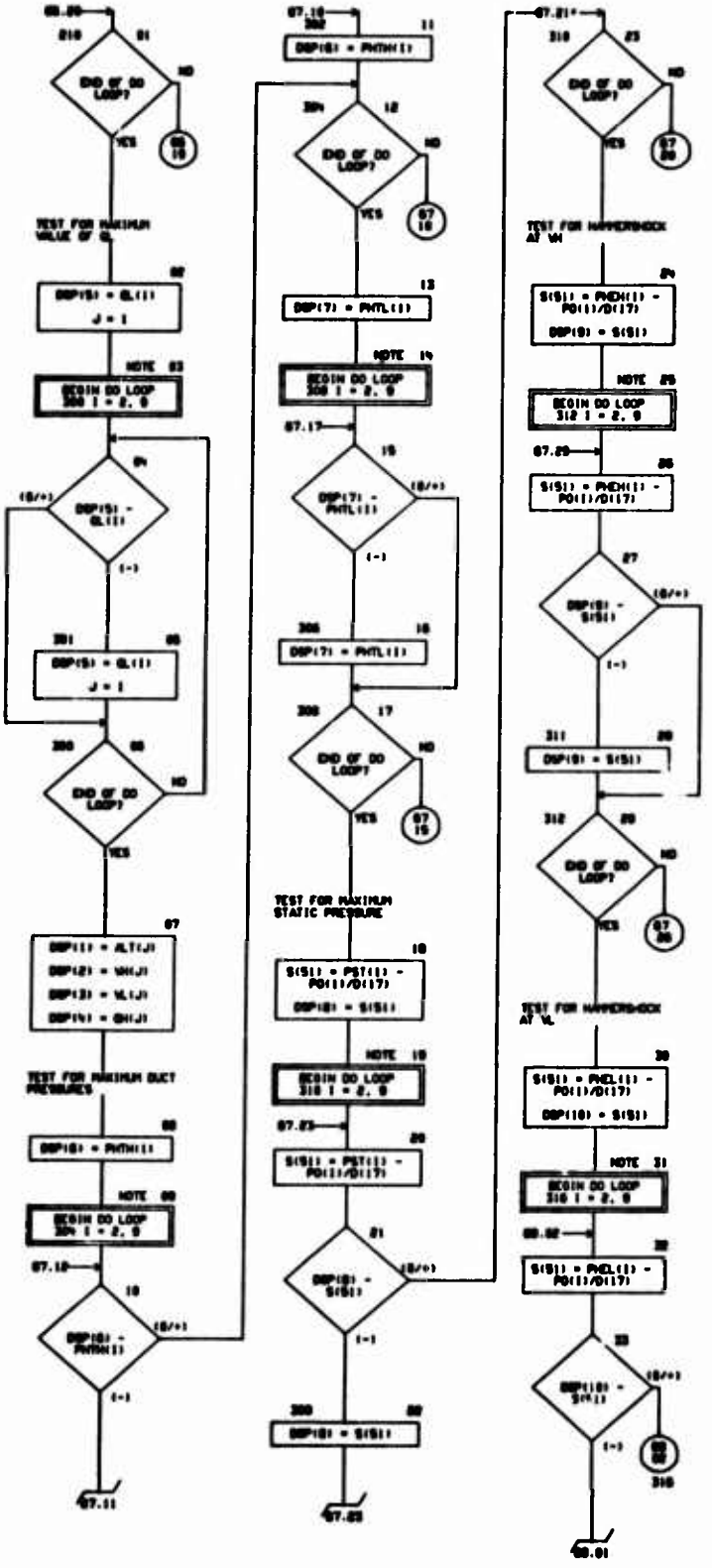


CHART TITLE - SUBROUTINE DSORPR

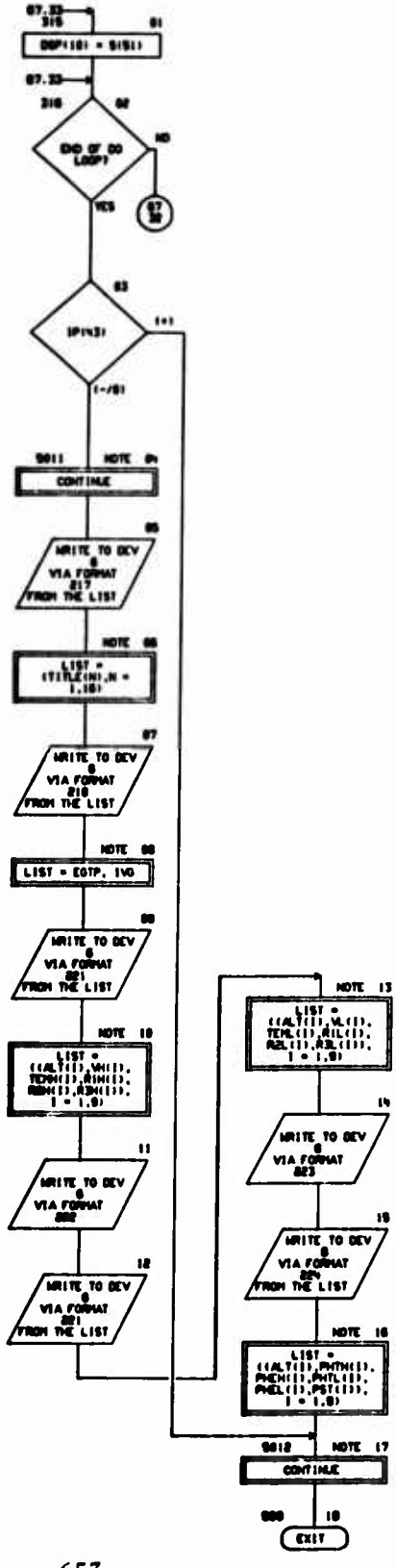


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCON(4320)
      COMMON /MISC/ XMISC(100)
      COMMON /IPRINT/ IP(80)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION DATS(40)
      DIMENSION TITLE(10)
      DIMENSION ALT(10),VM(10),VL(10),TEMH(10),TEML(10),PTM(10),
     1 PTL(10),PSL(10),RIM(10),RIL(10),R2M(10),R2L(10),R3M(10),
     2 R3L(10),PMTM(10),PMEM(10),PMTL(10),PMEL(10),PST(10),RATL(10)
      DIMENSION GM(10),GL(10),PO(10)
      DIMENSION DSP(10)
      EQUIVALENCE (S(1),TCON(1)),(GD(1),TCON(701)),(DV(1),TCON(1401)),
     1 (S(1),TCON(3721)),(ND(1),TCON(4121))
      EQUIVALENCE (EQU(1),D(81))
      EQUIVALENCE (DATS(1),GD(481))
      EQUIVALENCE (EQTP,DATS(2))
      EQUIVALENCE (TITLE(1),XMISC(85))
      EQUIVALENCE (GM(1),DV(81)),(GL(1),DV(91)),
     1 (PO(1),DV(21))
      EQUIVALENCE (ALT(1),DV(1)),(VM(1),DV(61)),(VL(1),DV(7)),
     1 (TEMH(1),DV(14)),(TEML(1),DV(15)),(PTM(1),DV(16)),
     2 (PTL(1),DV(17)),(PSL(1),DV(19)),(RIM(1),DV(20)),
     3 (RIL(1),DV(21)),(R2M(1),DV(22)),(R2L(1),DV(23)),
     4 (R3M(1),DV(24)),(R3L(1),DV(25)),(PMTM(1),DV(26)),
     5 (PMEM(1),DV(27)),(PMTL(1),DV(28)),(PMEL(1),DV(29)),
     6 (PST(1),DV(30)),(RATL(1),DV(3)))
      EQUIVALENCE (DSP(1),DV(3)))
      EQUIVALENCE (ND(101),I),(ND(102),J)
      EQUIVALENCE (ND(112),IVO)
   60 FORMAT(1H1,20X,23H*** WARNING MESSAGE ***/10X,
     1 41HRAM TEMPERATURE EXCEEDED FOR FANJET OPR =,F5.1/10X,
     2 8HRAM TEMP =,F8.2,7HLIMIT =,F8.2)
   61 FORMAT(1H1,20X,23H*** WARNING MESSAGE ***/10X,
     1 43HSPEED EXCEEDED FOR ENGINE INLET COMBINATION/20X,
     2 6HOPR =,F5.1,3X,12HINLET TYPE =,I3,3X,7HSPEED =,F5.2,3X,
     3 12HLIMIT SPEED =,F5.2)
  217 FORMAT(1H1,30X,21H** DSGNPR - IP(43) **//10X,8A10/10X,8A10)
  218 FORMAT(1H0, 35X, 38HSPEED PROFILE DESIGN CONSTANTS
          / 1H0, 30X, 14HBYPASS RATIO =, F5.2, 20X, 9HIVO =, I12 /
     1 1H0, 44X, 7HTEMP(1H), 9X, 9HSTATIC(1H), 12X, 15HHAMMERSHOCK (1H) /
     2 10X, 3HALT, 13X, 3HMN, 9X, 11HDEG RANKINE, 6X, 11HPRES. RATIO,
     3 6X, 4HFACE, 11X, 6HTHROAT )
  221 FORMAT( 8X, IF13.1, IF14.2, IF17.3, 3F16.4 )
  222 FORMAT(1H0,44X,7HTEMP(L),9X,9HSTATIC(L),12X,15HHAMMERSHOCK (L) /
     1 10X, 3HALT, 13X, 3HML, 9X, 11HDEG RANKINE, 6X, 11HPRES. RATIO,
     2 6X, 4HFACE, 11X, 6HTHROAT )
  223 FORMAT(1H0, 35X, 7HPRES(1H), 9X, 7HPRES(1H), 9X, 7HPRES(L), 9X,
     1 7HPRES(L), 10X, 9HSTATIC / 27X, 11HTHROAT-PSIA, 9X,
     2 11HENGINE-PSIA, 9X, 11HTHROAT-PSIA, 9X, 11HENGINE-PSIA, 6X,
     3 11HPRES THROAT )
  224 FORMAT( 8X, IF13.1, IF15.3, 4F16.3)
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE DSTNOR

CHART TITLE - SUBROUTINE DSTNOR

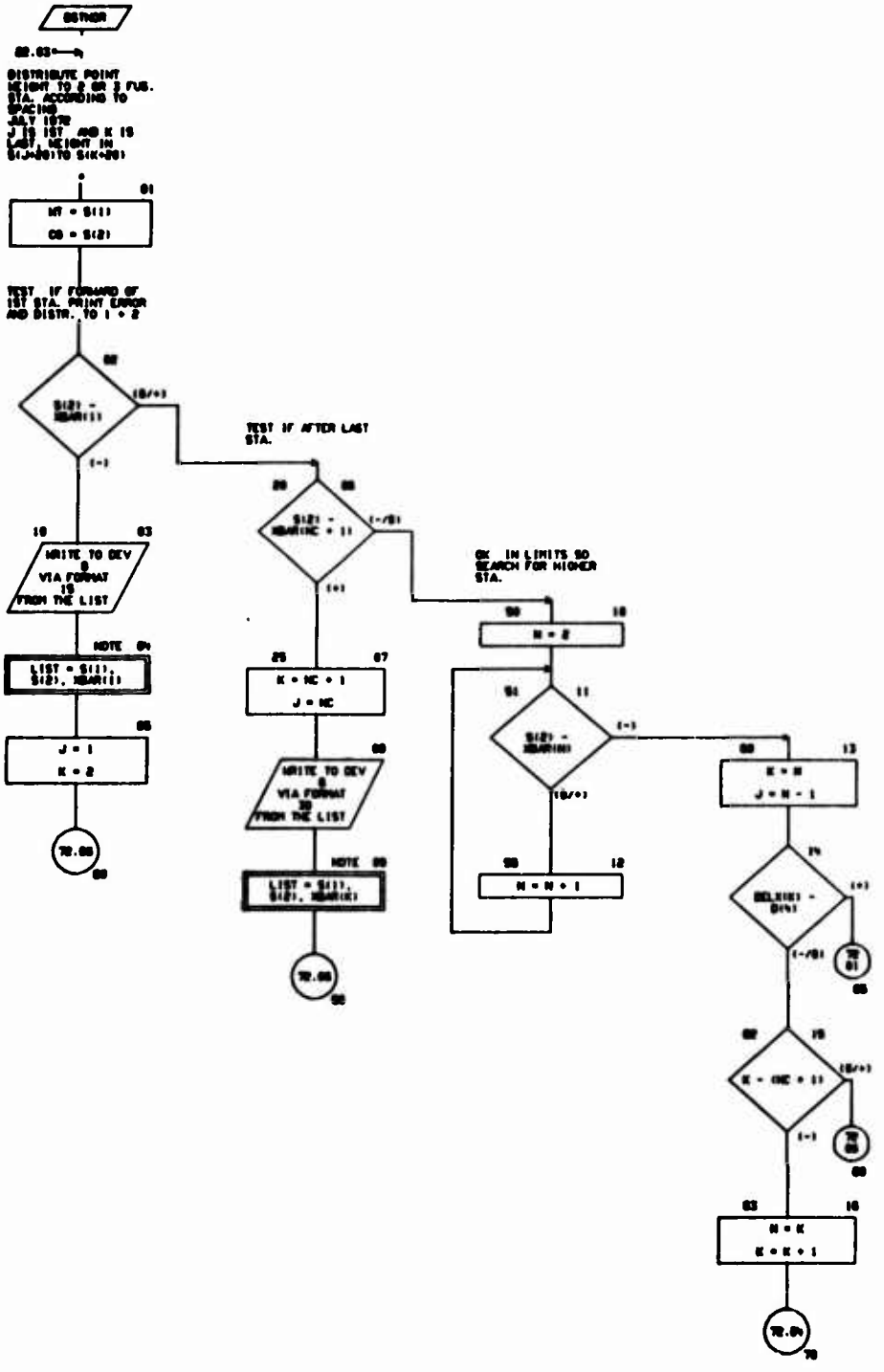


CHART TITLE - SUBROUTINE SETPOR

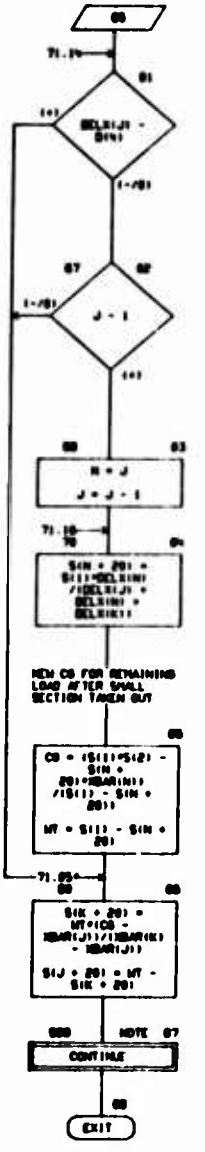


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),      DV(2320),S(400),ND(200)
     . ,NBAR(20), DELX(20), DNB(440)
      EQUIVALENCE (TCOM(1)),D(1)), (TCOM(1401),DV(1))
     . ,(TCOM(3721),S(1)), (TCOM(4121),ND(1))
     .,(DV(2),DN(1))
     .,(DNB(1401),NBAR(1)), (DNB(1601),DELX(1))
     .,(ND(102),J), (ND(103),K), (ND(115),NC)
  10  FORMAT(1H1,23H***** IN SUBROR, HEIGHT,IE13.6,19H LB. LOCATED AT,
     . IE13.6,19H IS END OF 1ST STA.,IE13.6, 2H *****)
  20  FORMAT(1H1,23H***** IN SUBROR, HEIGHT,IE13.6,19H LB. LOCATED AT,
     . IE13.6,20H IS AFT OF LAST STA.,IE13.6, 2H *****)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*************************************************************
C                    SUBROUTINE DSTTRI
C*************************************************************
```

CHART TITLE - SUBROUTINE DSTRI

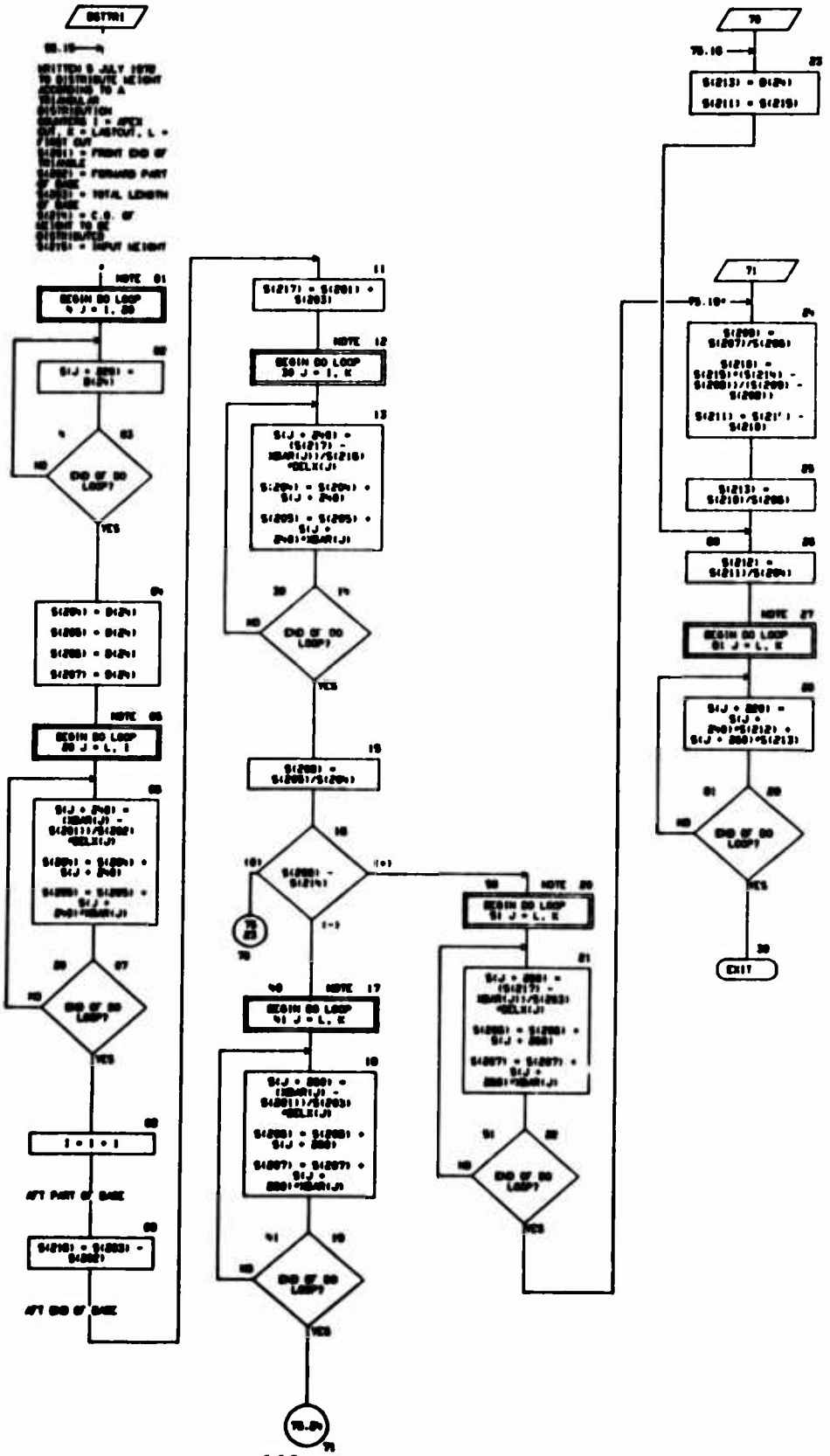


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
 COMMON TCOM(4320)
DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
DIMENSION ND(20),NBAR(20),DELX(20)
DIMENSION GDD(80),DVB(440)
EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401),
 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
EQUIVALENCE (GDD(1),GD(381)),(DVB(1),DV(431))
EQUIVALENCE (ND(1),GDD(56)),(NBAR(1),DVB(141)),(DELX(1),DVB(161))
EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104))
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE [illegible]

CHART TITLE - SUBROUTINE DSTRIP

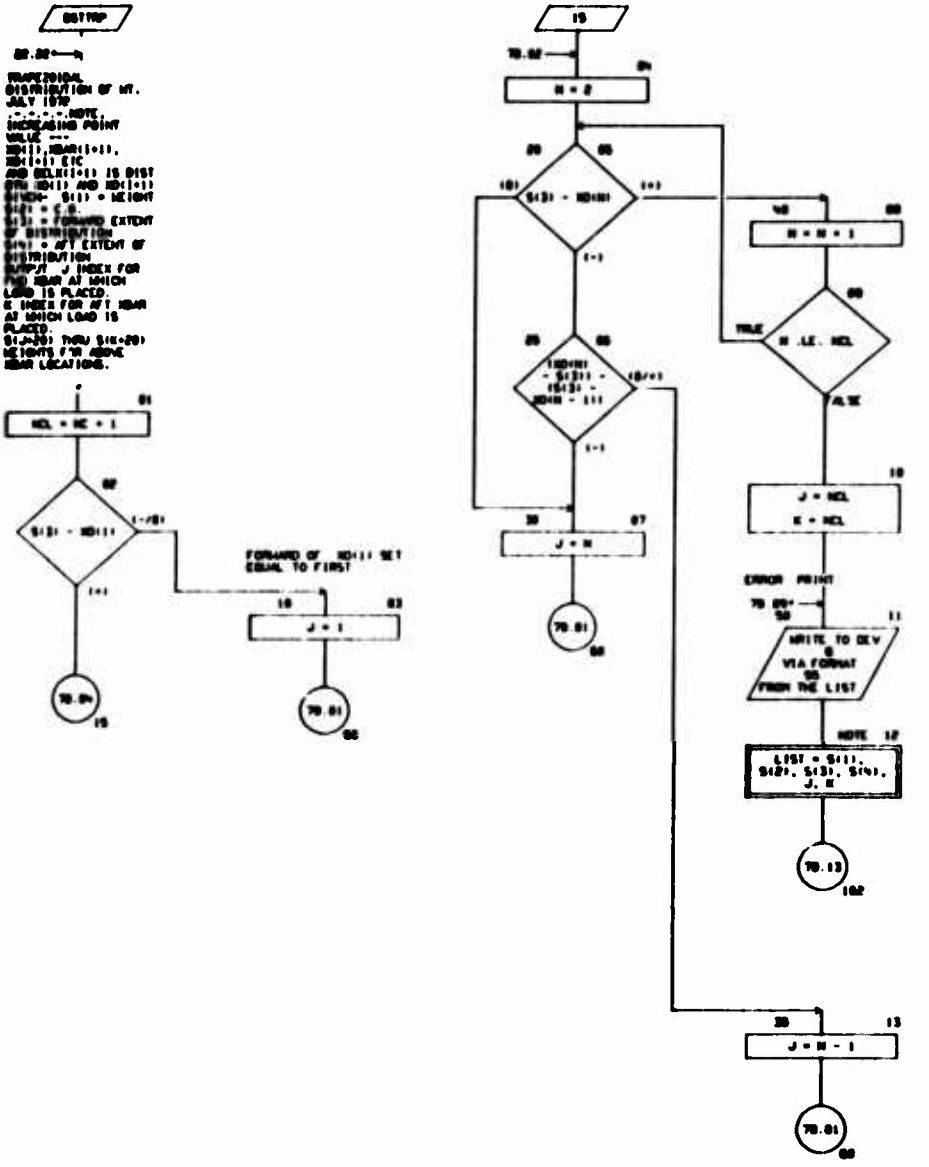


CHART TITLE - [illegible]

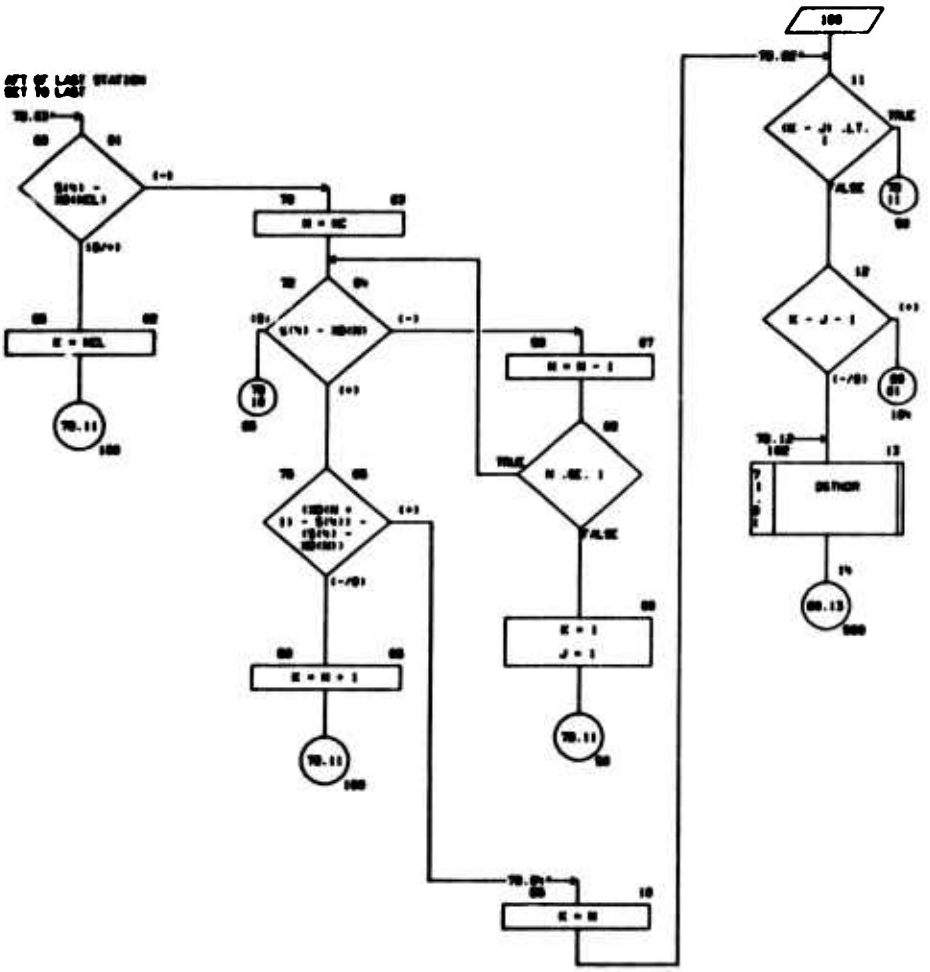

ACT OF LAST STATION
SET TO LAST
N = NC
E = NCL
N = N - 1
N .NE. 1
E = 1
J = 1
E = N + 1
E = N
NC - J .LT. 1
K - J - 1
DETHOR

CHART TITLE - SUBROUTINE DSTTRP

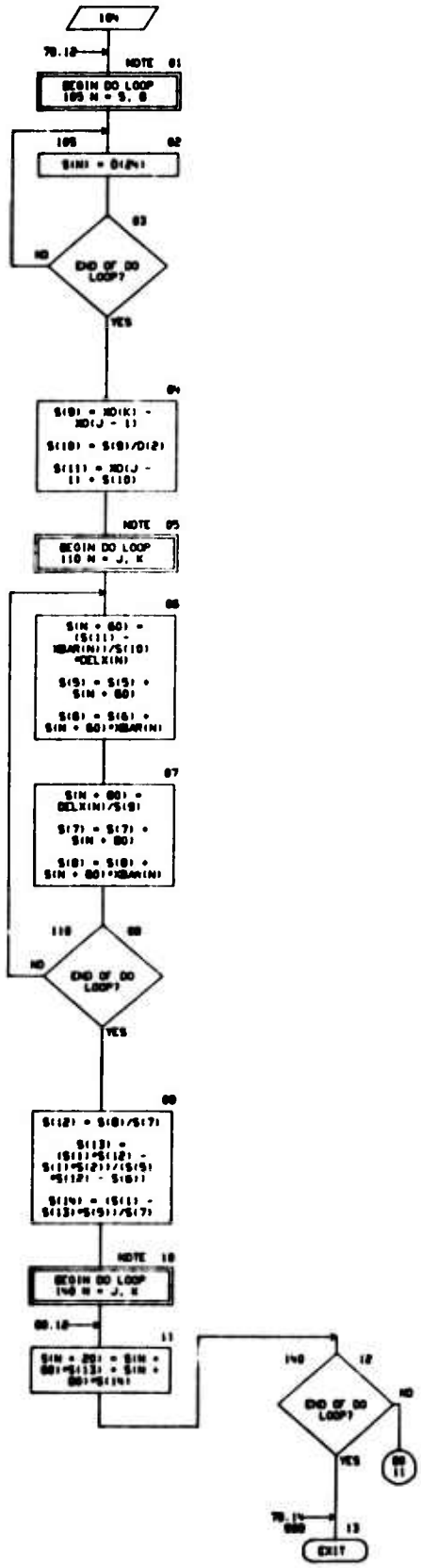


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION B(700),GD(700),DV(2320),S(400),ND(200)
     . ,GDB(80), DVB(440), XD(20), XBAR(20), DELX(20)
      EQUIVALENCE (TCOM(1),B(1)), (TCOM(701),GD(1)), (TCOM(1401),DV(1))
     . ,(TCOM(3721),S(1)), (TCOM(4121),ND(1))
     . ,(GD(281),GDB(1)), (GDB(80),XD(1))
     . ,(DV(441),DVB(1)), (DVB(141),XBAR(1)), (DVB(161),DELX(1))
     . ,(ND(182),J), (ND(183),X), (ND(119),NC)
  99  FORMAT(///1H0,'DV(',I5,') IN OSTTRP, WEIGHT CANNOT BE DISTRIBUTED /
     . 1X, 'DWT=',1F14.2,2X, 'DCG=',1F9.2, 2X, 'WFWD=', 1F9.2, 'WAFT=',
     . 1F9.2,1H/RETURN J,X =, 113, 1H,,113 )
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE SUCOCO

CHART TITLE - SUBROUTINE SWEEP

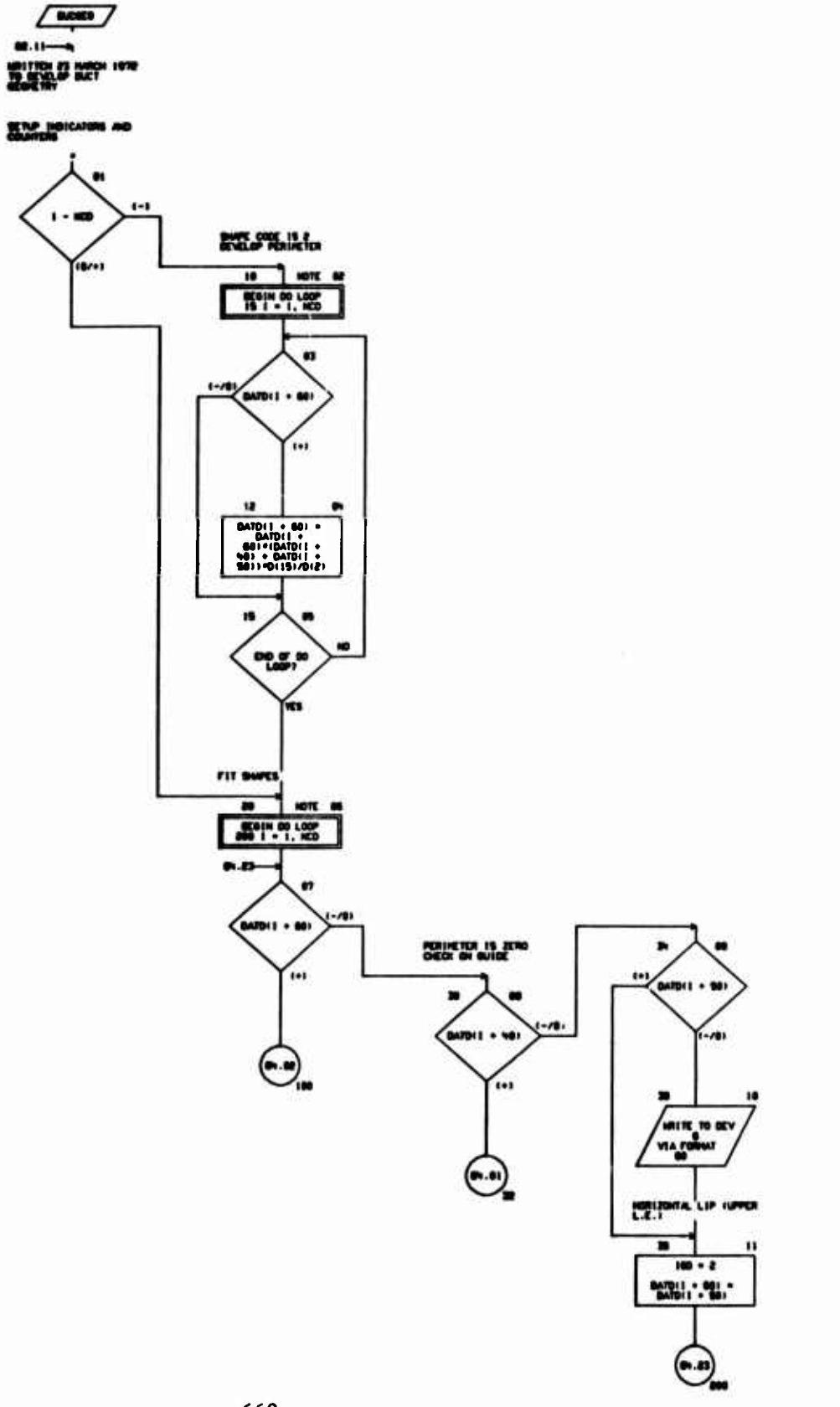


CHART TITLE - SUBROUTINE SUCSEQ

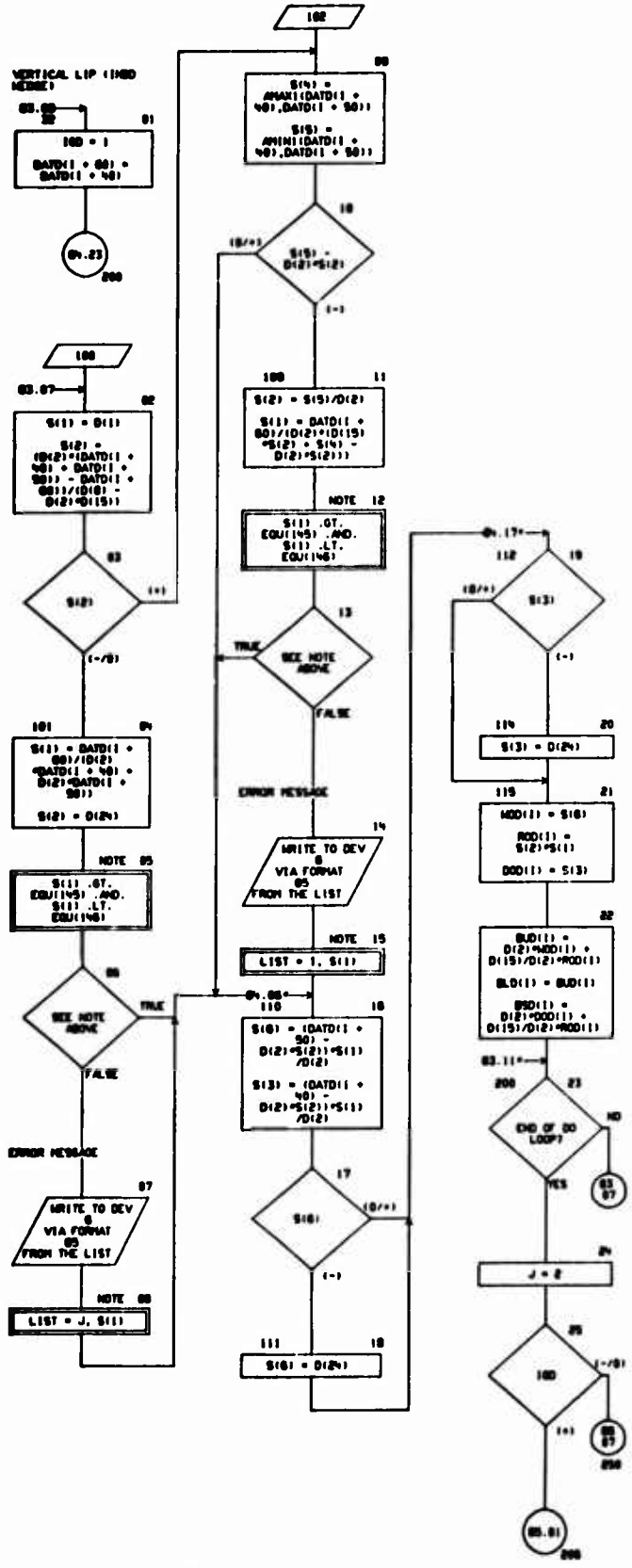
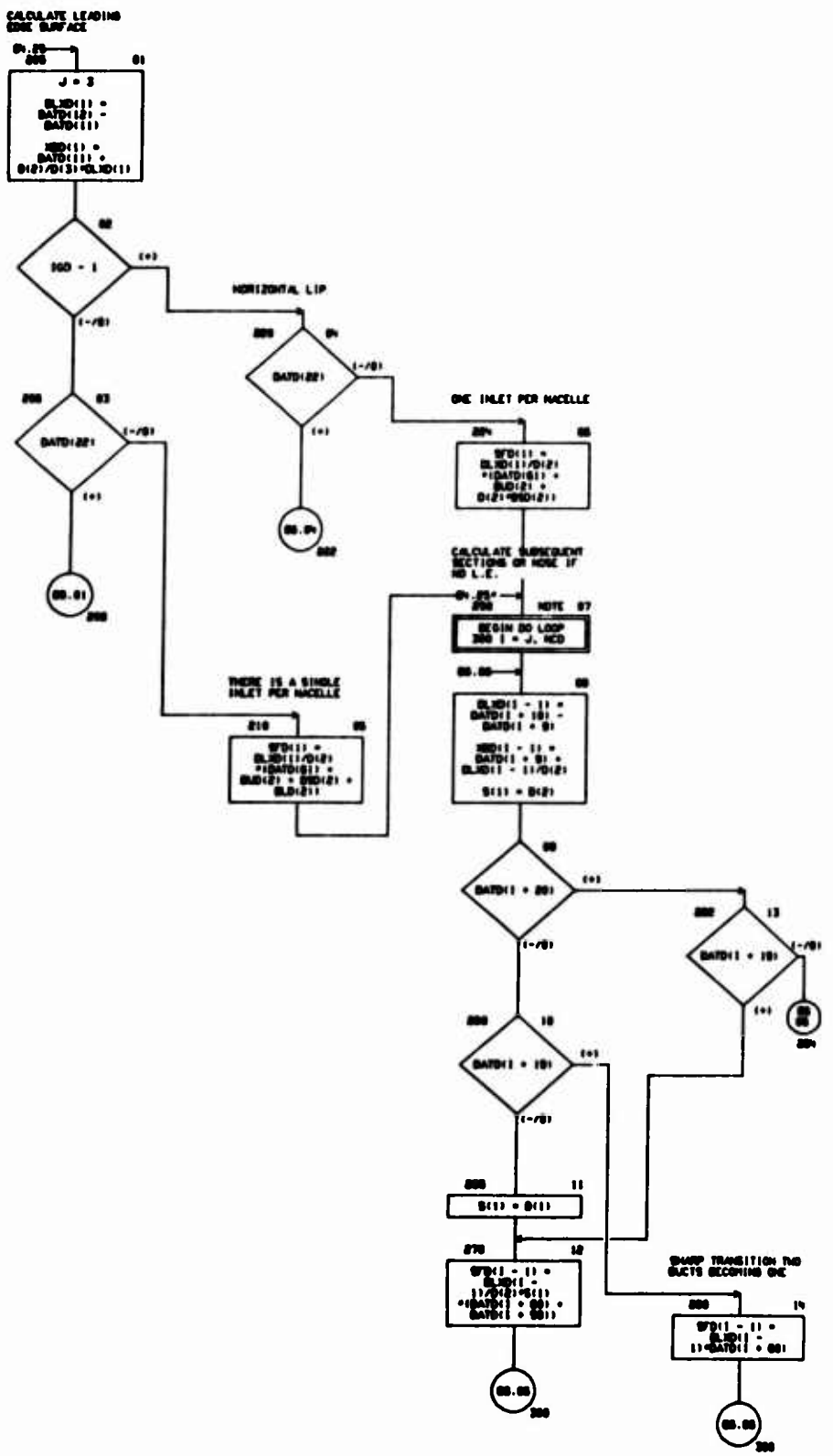


CHART TITLE - SUBROUTINE SUCSEQ



CHAR / TITLE - SUBROUTINE SUCKED

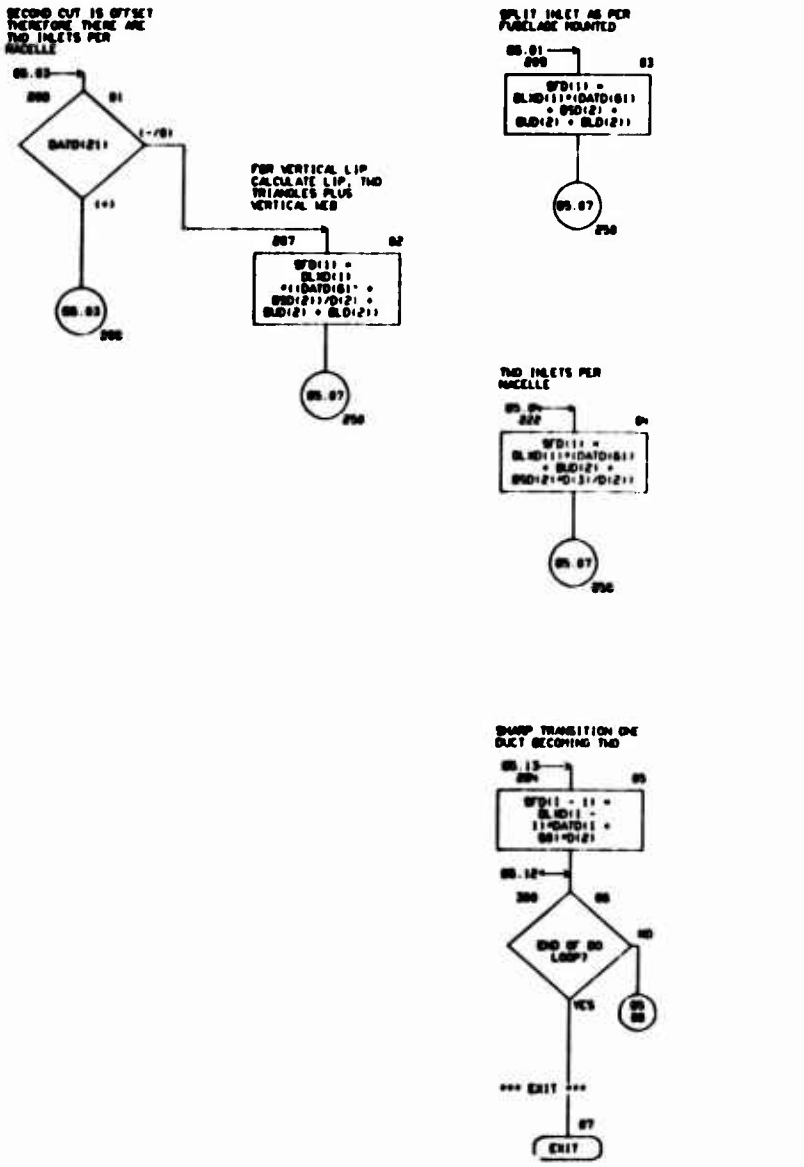


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION IX(700),ID(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION DATA(70)
      DIMENSION DVD(100)
      DIMENSION MSD(10),RSD(10),SSD(10),SUB(10),SLD(10),SSB(10),
     1 SLB(10),STB(10),MSB(10)
      EQUIVALENCE (D(1),TCOM(1)),(ID(1),TCOM(70)),(DV(1),TCOM(140)),
     1 (S(1),TCOM(372)),(ND(1),TCOM(412))
      EQUIVALENCE (EQU(1),D(0))
      EQUIVALENCE (DATD(1),ID(50))
      EQUIVALENCE (DVD(1),DV(87))
      EQUIVALENCE (MSD(1),DVD(1)),(RSD(1),DVD(11)),(SSD(1),DVD(21)),
     1 (SUB(1),DVD(31)),(SLD(1),DVD(41)),(SSB(1),DVD(51)),
     1 (SLB(1),DVD(61)),(STB(1),DVD(71)),(MSB(1),DVD(81))
      EQUIVALENCE (ND(101),I),(ND(102),J)
      EQUIVALENCE (ND(119),IGO),(ND(117),NCD),(ND(118),KCD)
   80 FORMAT(30HREMARKING FROM SUCCEO IN DATA MANAGEMENT /30X,23HOBJECT LIP
     1 GEOMETRY ERROR )
   85 FORMAT(30HREMARKING FROM SUCCEO IN DATA MANAGEMENT / 8H SECTION,
     1 I3, 9IN IS RECT. OR ROUNDED RECT., CORRECTION IS, F7.4 )
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE BDMOD

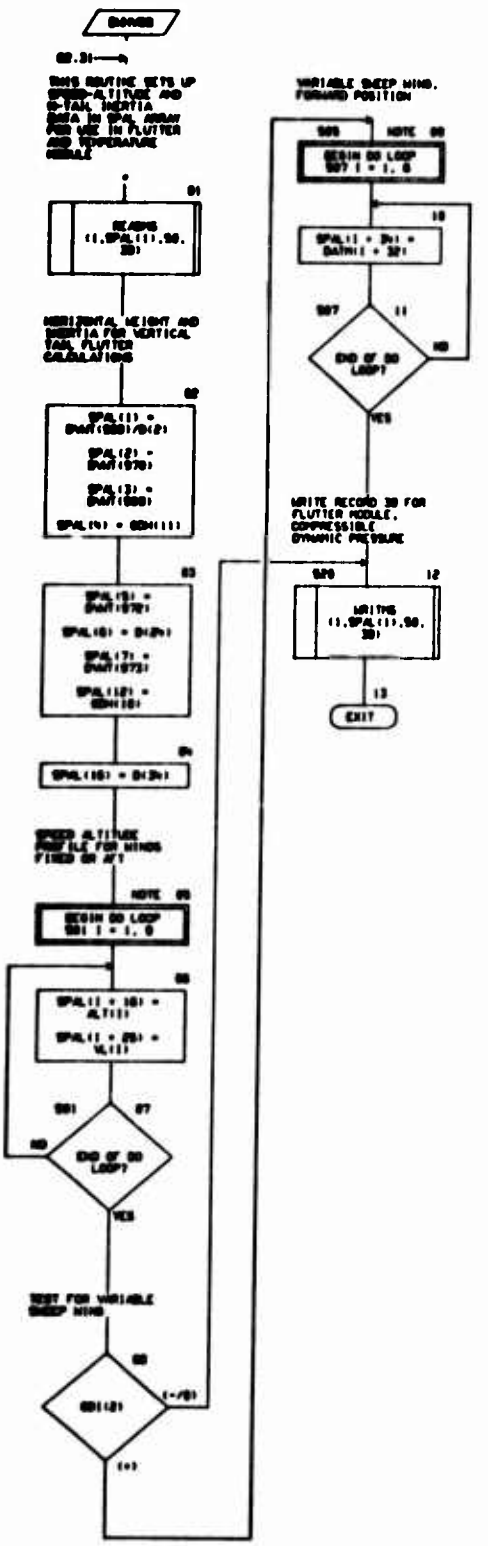


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),CD(700),DV(2320),S(400),ND(200)
      DIMENSION SPAL(50)
      DIMENSION CD1(20),CDM(40),DATM(40),ALT(10),VL(10)
      DIMENSION DART(1000)
      EQUIVALENCE (D(1),TCOM(1)),(CD(1),TCOM(70(1)),(DV(1),TCOM(140(1))
     . ,(S(1),TCOM(372(1)),(ND(1),TCOM(412(1))
      EQUIVALENCE (SPAL(1),S(1))
      EQUIVALENCE (CD1(1),CD(1)),(CDM(1),CD(30(1)),(DATM(1),CD(51))
     . ,(ALT(1),DV(1)),(VL(1),DV(7))
      EQUIVALENCE (DART(1),DV(112(1))
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE FTOTAL

CHART TITLE - SUBROUTINE FTOTAL

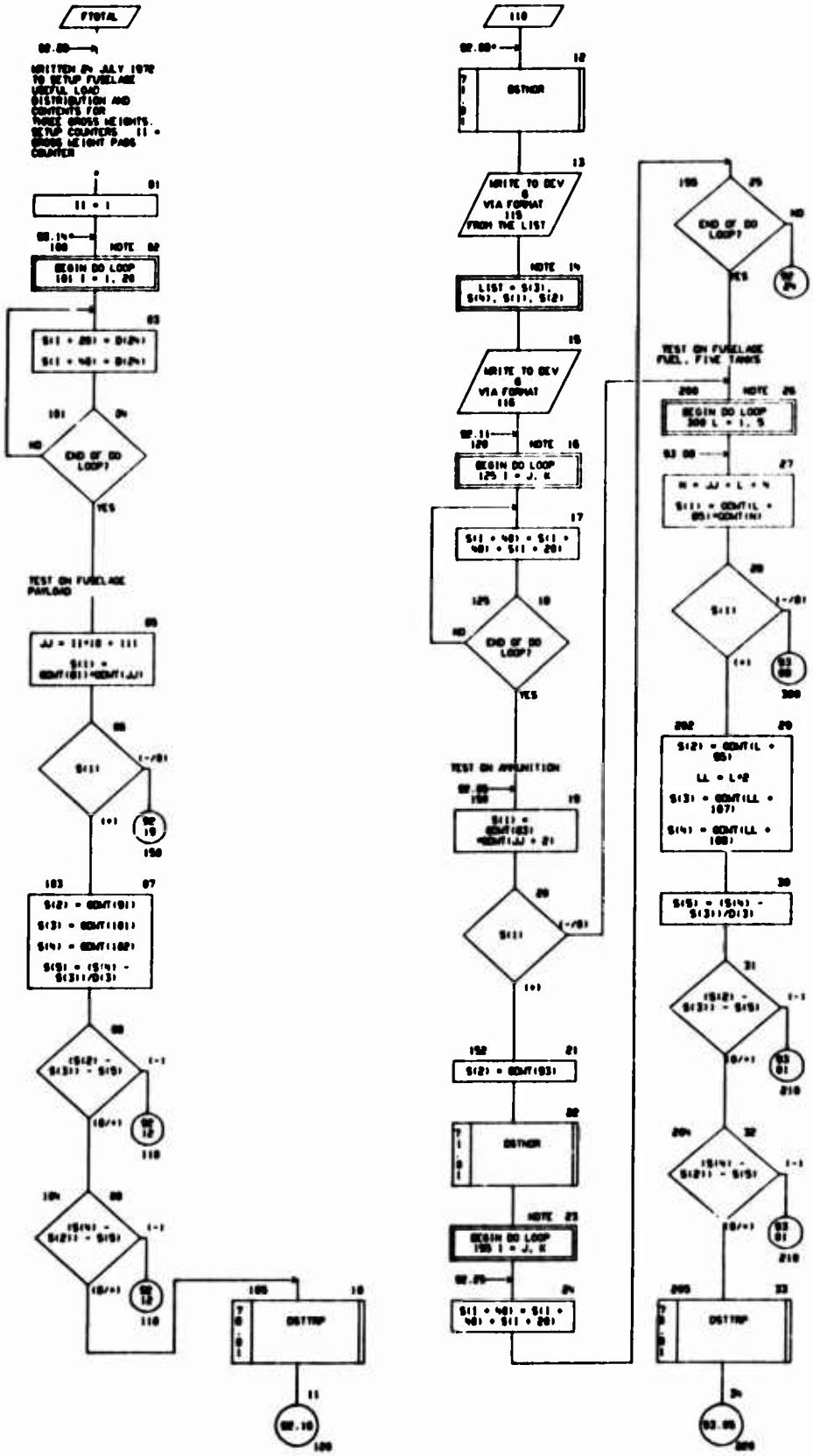


CHART TITLE - DERIVATIVE PROTAL

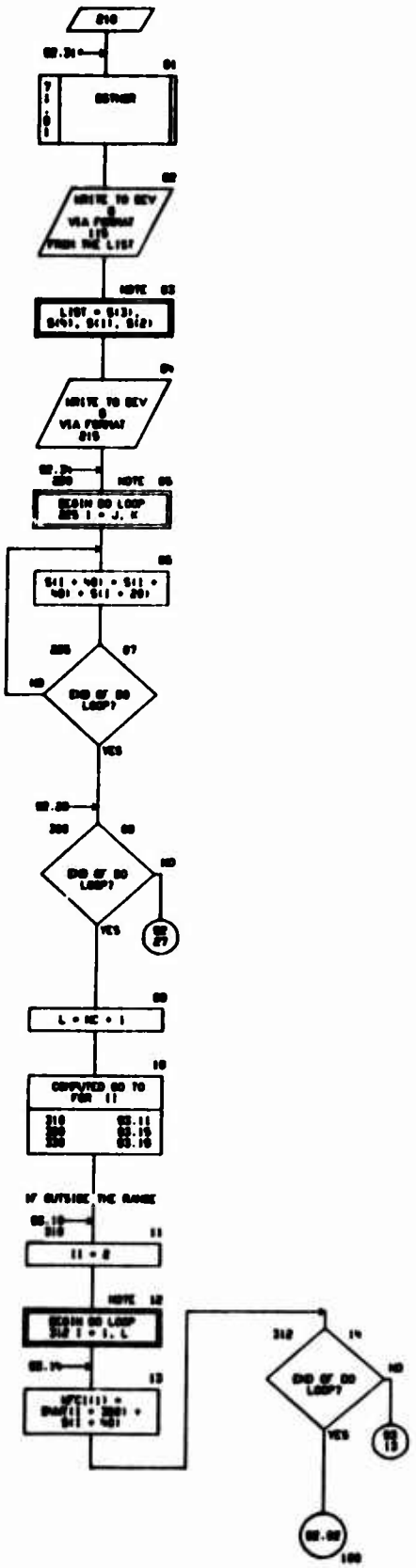


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION B(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION GDMT(100)
      DIMENSION DMAT(1000),WFC1(20),WFC2(20),WFC3(20)
      EQUIVALENCE (B(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GDMT(1),GD(91))
      EQUIVALENCE (DMAT(1),DV(1121)),(WFC1(1),DMAT(701)),
     (WFC2(1),DMAT(801)),(WFC3(1),DMAT(821))
      EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104)),
     (M,ND(105)),(II,ND(107)),(JJ,ND(108)),(LL,ND(110))
      EQUIVALENCE (NC,ND(115))
  110 FORMAT(2H0*** WARNING FROM FTOTAL *** /
     28X,33HPREVIOUS FORE AND AFT LIMITS WERE, 2F10.2/
     38X,F12.2,17H LBS AT FUS. STA.,F8.2,10X,21HDISTRIBUTED BY DSTNOR)
  115 FORMAT(1H+,19X,18HFUSELAGE PAYLOAD )
  215 FORMAT(1H+,22X,13HFUSELAGE FUEL )
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*****************************************************************
C                    SUBROUTINE FUSDST
C*****************************************************************
```

CHART TITLE - SUBROUTINE FUSDST

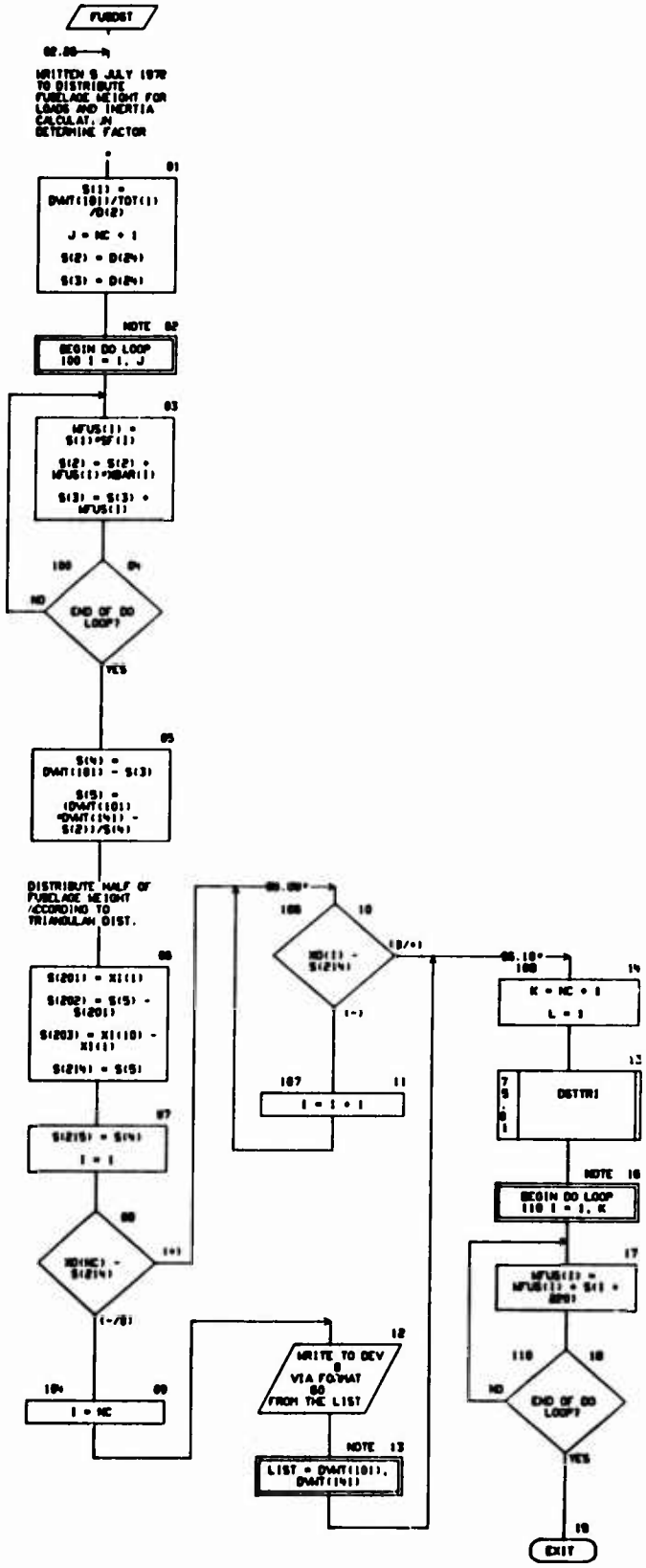


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4200)
      DIMENSION D(700),GD(700),DV(2329),S(400),ND(700)
      DIMENSION DVAT(500),NFUS(20)
      DIMENSION GDB(80),DVB(440)
      DIMENSION XI(10),XD(20),XBAR(20),SF(20),TOT(20)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (DVAT(1),DV(1121)), (NFUS(1),DVAT(3n1))
      EQUIVALENCE (GDB(1),GD(39)),(DVB(1),DV(443))
      EQUIVALENCE (XI(1),GDB(6)),(XD(1),GDB(56)),(XBAR(1),DVB(191)),
     (SF(1),DVB(101)),(TOT(1),DVB(381))
      EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104))
      EQUIVALENCE (NC,ND(119))
   88 FORMAT(21H0** FUSDST WARNING **    /10X,28HCHECK MT AND CD DATA,
     /10X,33HFUSELAGE MT DIST IS NOT REALISTIC,/10X,4HMT =,F10.2,
     3X,4HCD =,F8.2)
```

```
CHART TITLE - INTRODUCTORY COMMENTS


                    ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
                                         SUBROUTINE FUSGEO
                    ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
```

CHART TITLE - SUBROUTINE FUSEG

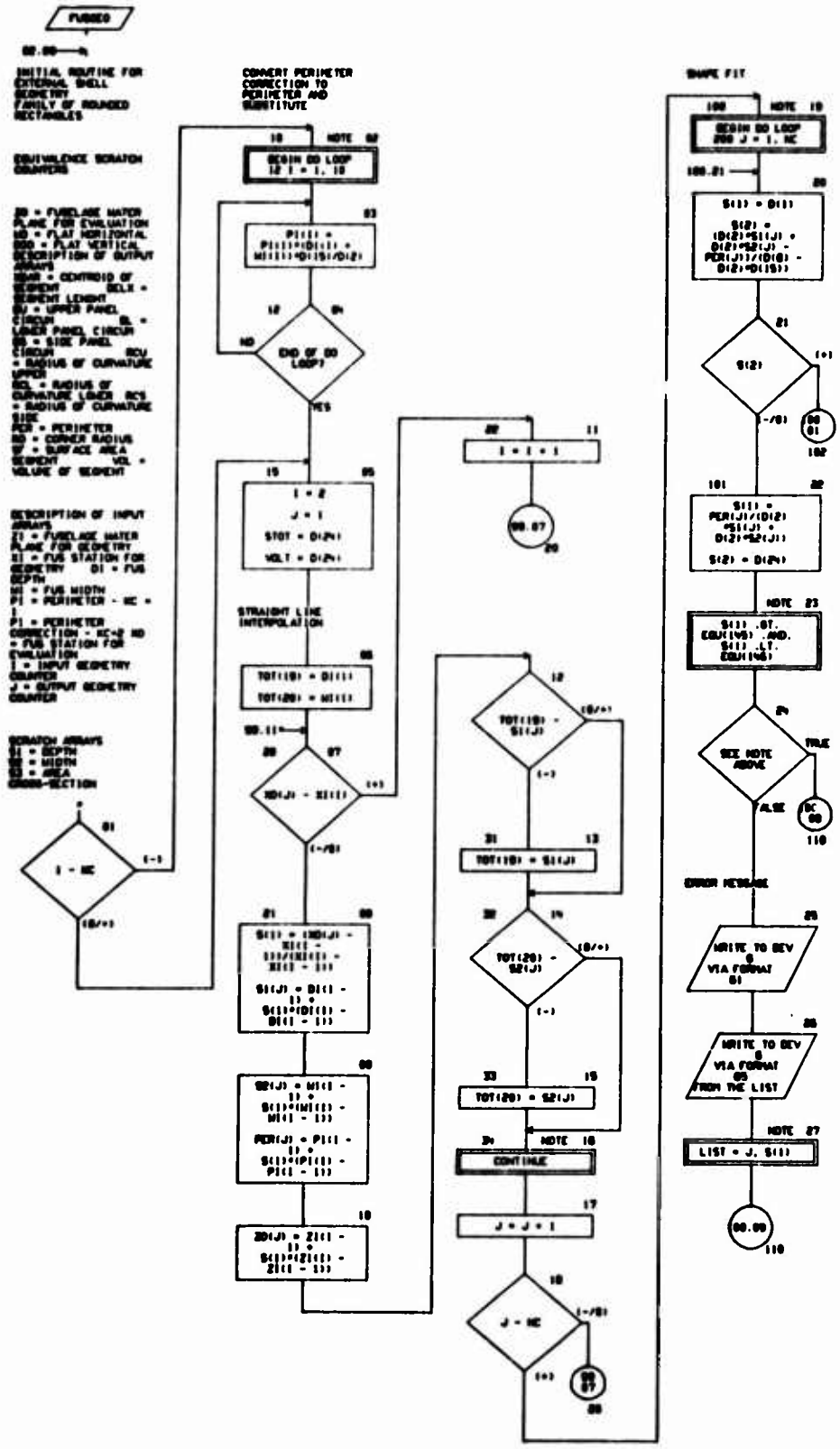


CHART TITLE - SUBROUTINE FUSEG

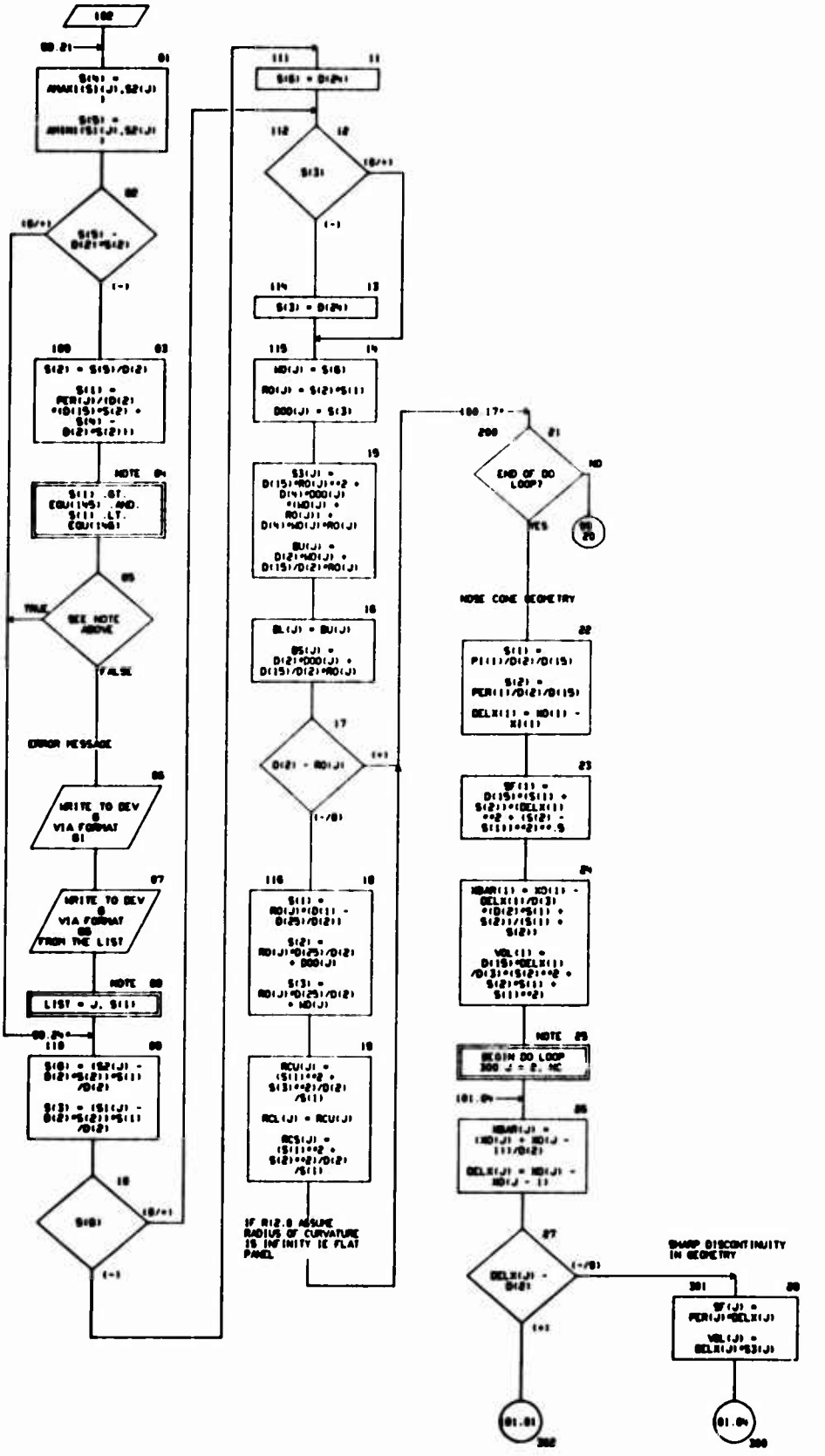


CHART TITLE - SUBROUTINE FUSEG

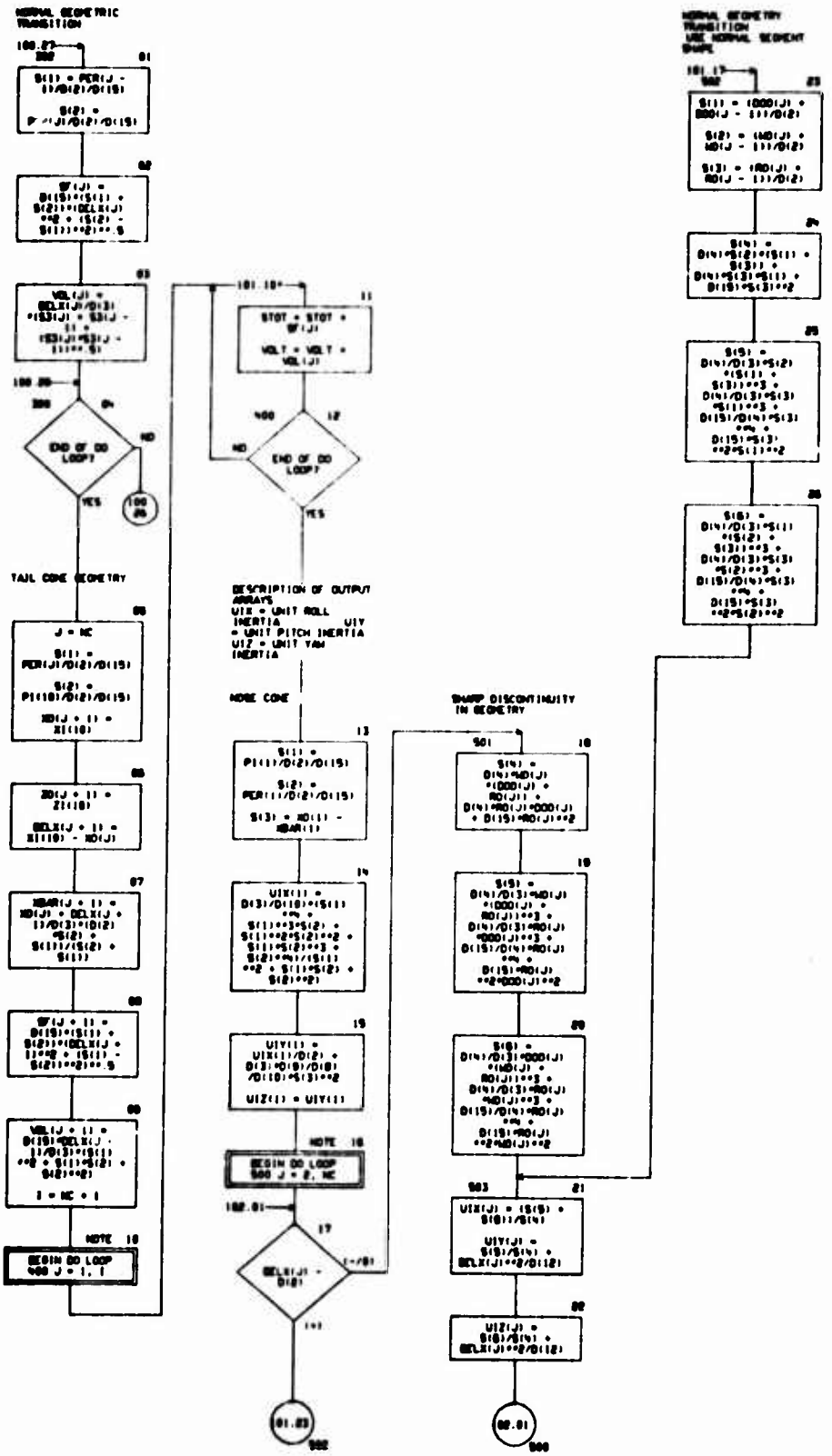


CHART TITLE - SUBROUTINE FUNKO

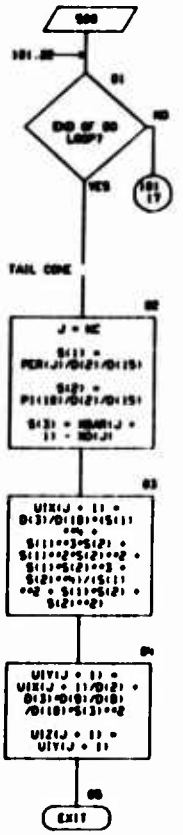


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),QD(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION QDB(80),DVB(440)
      DIMENSION X1(10),Z1(10),D1(10),M1(10),P1(10),RD(20)
      DIMENSION Z0(20),RCU(20),RCL(20),RCS(20),BU(20),BL(20),
     1 BS(20),XBAR(20),DELX(20),SF(20),VOL(20),DOO(20),WD(20),RO(20),
     2 PER(20)
      DIMENSION S1(20),S2(20),S3(20),TOT(20)
      DIMENSION U1X(20),U1Y(20),U12(20)
      EQUIVALENCE (D(1),TCOM(1)),(QD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (EQU(1),D(1))
      EQUIVALENCE (QDB(1),QD(381)),(DVB(1),DV(1431))
      EQUIVALENCE (X1(1),QDB(5)),(Z1(1),QDB(16)),(D1(1),QDB(26)),
     1 (M1(1),QDB(36)),(P1(1),QDB(46)),(RD(1),QDB(56))
      EQUIVALENCE (Z0(1),DVB(1)),(RCU(1),DVB(21)),(RCL(1),DVB(41)),
     1 (RCS(1),DVB(61)),(BU(1),DVB(81)),(BL(1),DVB(101)),
     2 (BS(1),DVB(121)),(XBAR(1),DVB(141)),(DELX(1),DVB(161)),
     3 (SF(1),DVB(181)),(VOL(1),DVB(201)),(DOO(1),DVB(221)),
     4 (WD(1),DVB(241)),(RO(1),DVB(261)),(PER(1), DVB(281))
      EQUIVALENCE (S1(1),DVB(301)),(S2(1),DVB(321)),(S3(1),DVB(341)),
     1 (TOT(1),DVB(361))
      EQUIVALENCE (U1X(1),DVB(381)),(U1Y(1),DVB(401)),(U12(1),DVB(421))
      EQUIVALENCE (TOT(1),STOT),(TOT(2),VOLT)
      EQUIVALENCE (ND(115),NC), (ND(116),KC)
      EQUIVALENCE (ND(101),I),(ND(102),J)
 61   FORMAT( 39H0WARNING FROM FUSGEO IN DATA MANAGEMENT )
 65   FORMAT( 8H SECTION, I13,3AH IS RECTANGULAR, CORR  FACTOR IS,F6 3)
 66   FORMAT( 8H SECTION,I13,3AH IS ROUNDED RECT , CORR  FACTOR IS,F6 3)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*************************************************************
C                    SUBROUTINE NAMES
C*************************************************************
```

CHART TITLE - SUBROUTINE NACGEO

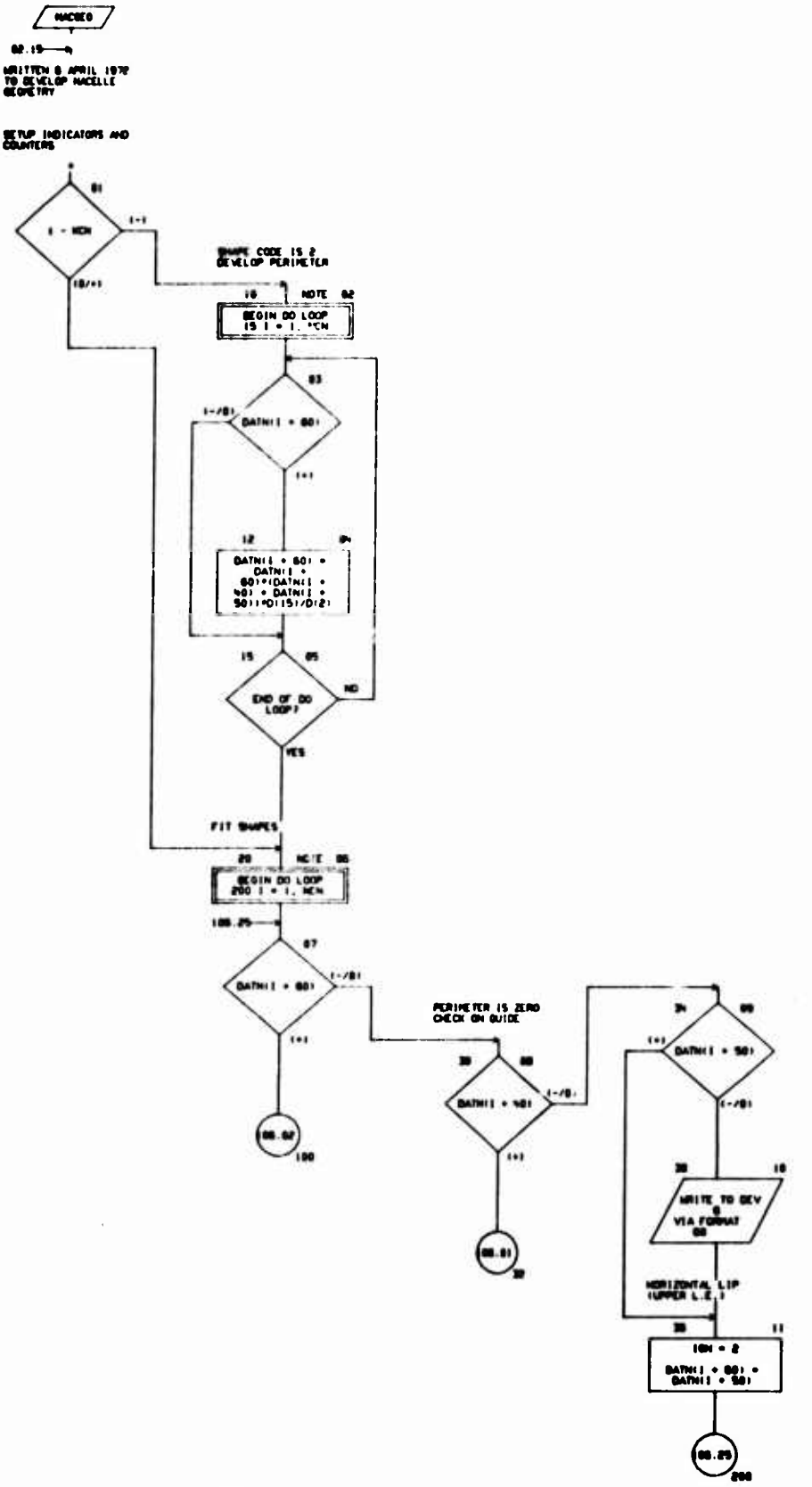


CHART TITLE - SUBROUTINE MACRO

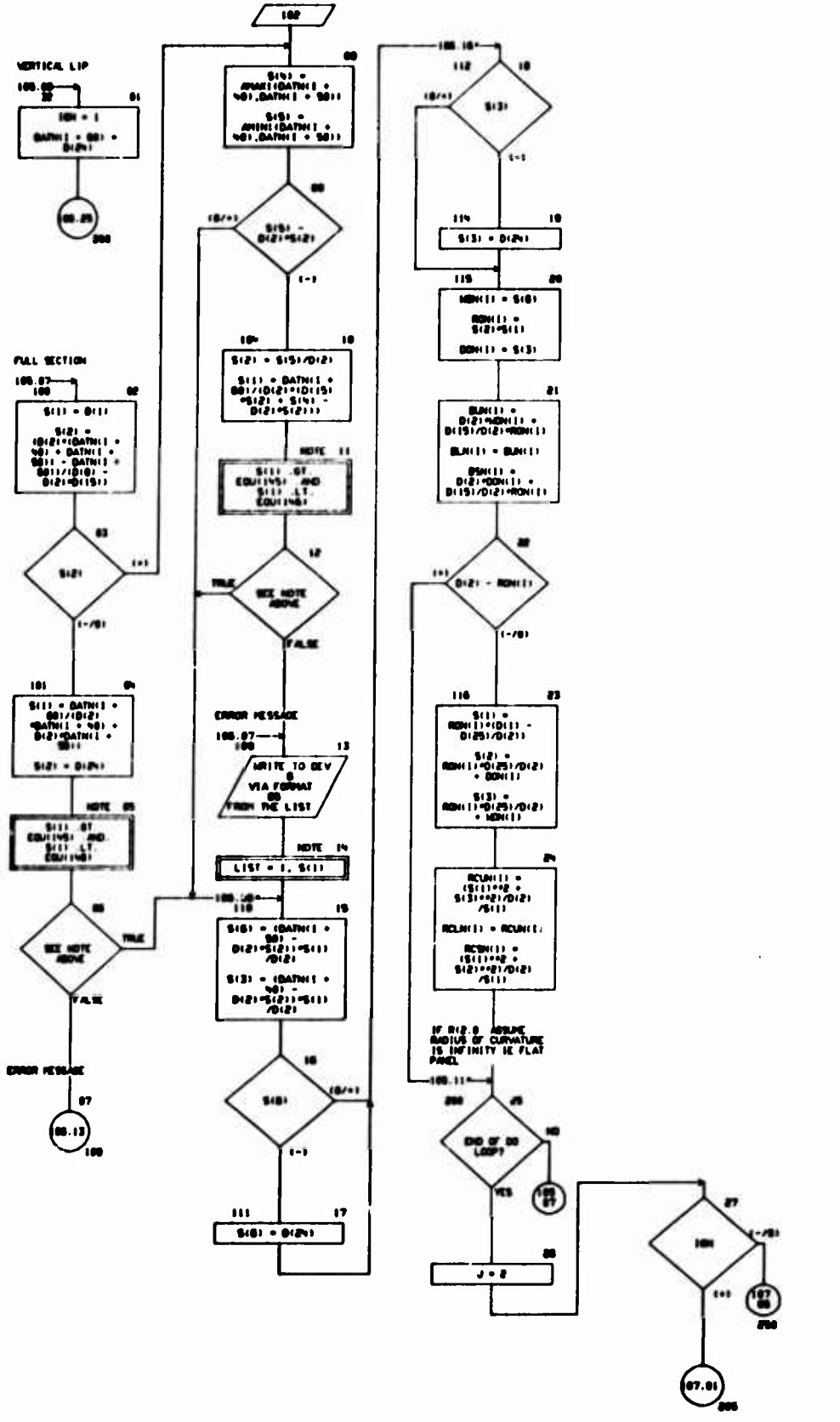


CHART TITLE - SUBROUTINE HACDEO

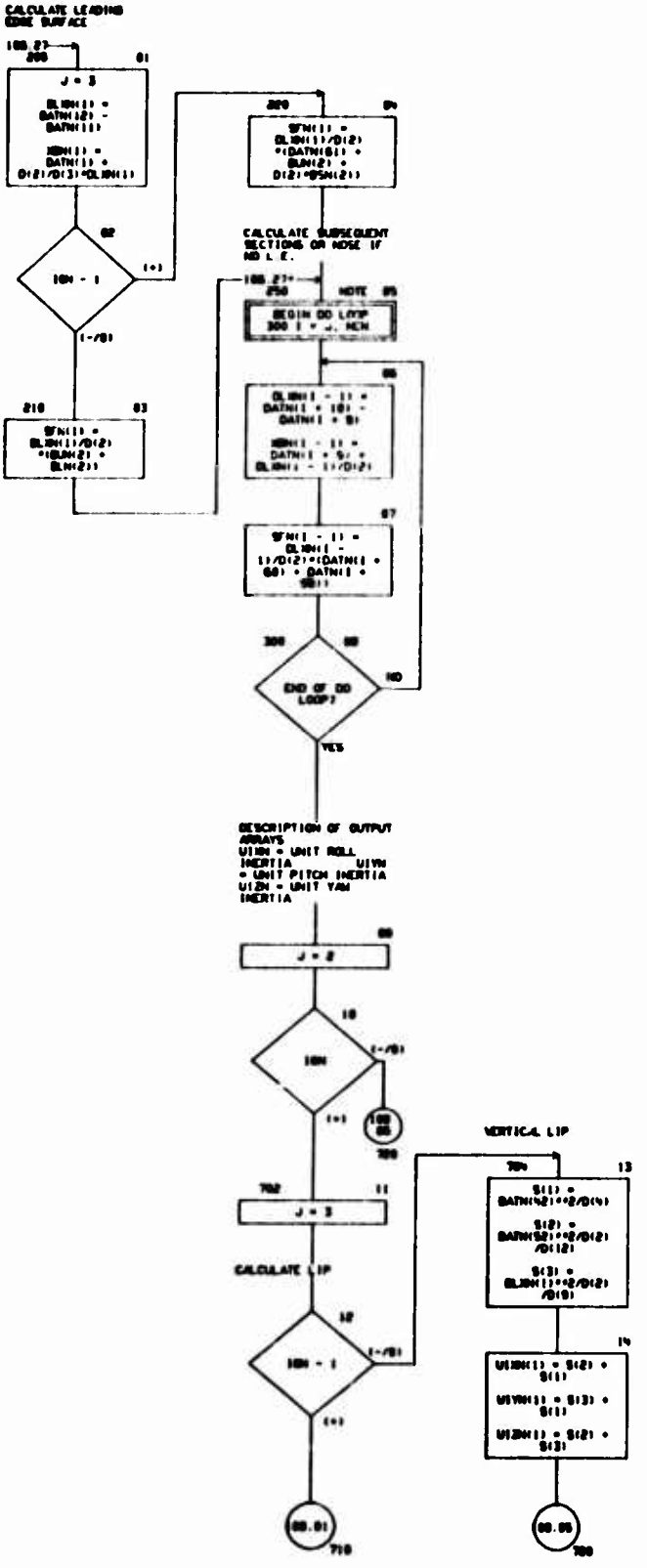


CHART TITLE - SUBROUTINE HACKED

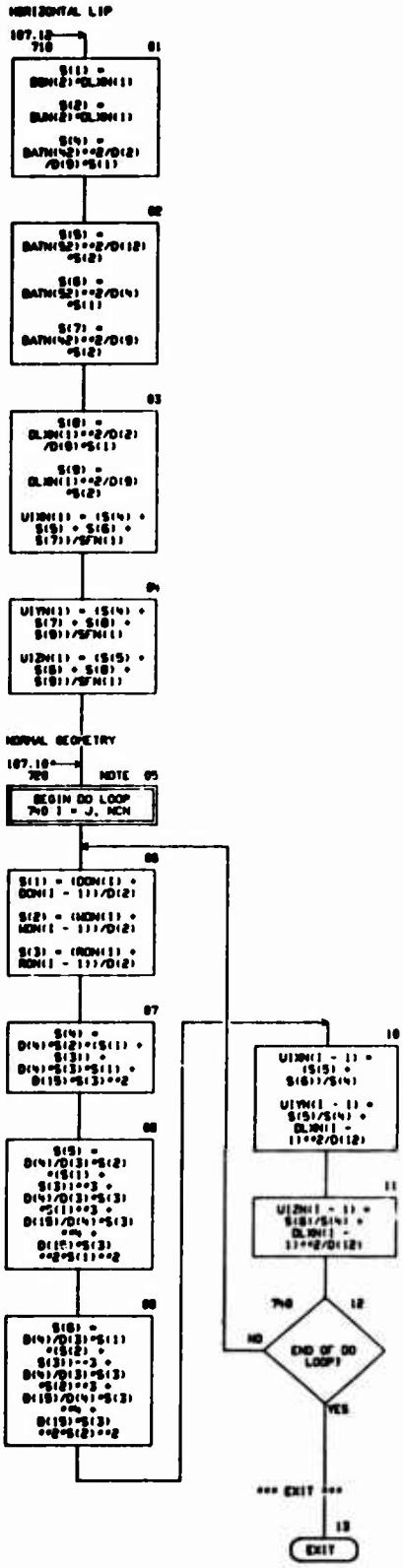


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),GD(700),DV(2320),$(400),ND(200)
      DIMENSION DATN(70)
      DIMENSION DVN(150)
      DIMENSION EQU(200)
      DIMENSION HDN(10),RDN(10),DDN(10),BUN(10),BLN(10),BSN(10),
     1 BLXN(10),BFN(10),RCUN(10),RCLN(10),RCSN(10),XBN(10)
      DIMENSION UIXN(10),UIYN(10),UIZN(10)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 ($(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (EQU(1),D(81))
      EQUIVALENCE (DATN(1),GD(581))
      EQUIVALENCE (DVN(1),DV(97))
      EQUIVALENCE (HDN(1),DVN(1)),(RDN(1),DVN(11)),(DDN(1),DVN(21)),
     1 (BUN(1),DVN(31)),(BLN(1),DVN(41)),(BSN(1),DVN(51)),
     1 (BLXN(1),DVN(61)),(BFN(1),DVN(71)),(RCUN(1),DVN(81)),
     1 (RCLN(1),DVN(91)),(RCSN(1),DVN(101)),(XBN(1),DVN(111))
      EQUIVALENCE (UIXN(1),DVN(121)),(UIYN(1),DVN(131)),
     1 (UIZN(1),DVN(141))
      EQUIVALENCE (ND(101),I),(ND(102),J)
      EQUIVALENCE (NCN,ND(119)),(KCN,ND(120)),(IGN,ND(121))
 88   FORMAT (*0HO WARNING FROM NACGEO IN DATA MANAGEMENT,10X,
     2HNACELLE LIP GEOMETRY ERROR)
 89   FORMAT (*0HO WARNING FROM NACGEO IN DATA MANAGEMENT /5H SECTION,
     1 I13,47H IS RECTANGLE OR ROUNDED RECT., CORR. FACTOR IS, 1F8.3 )
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*************************************************************
C                  SUBROUTINE NOSDEO
C*************************************************************
```

CHART TITLE - SUBROUTINE NOSEO

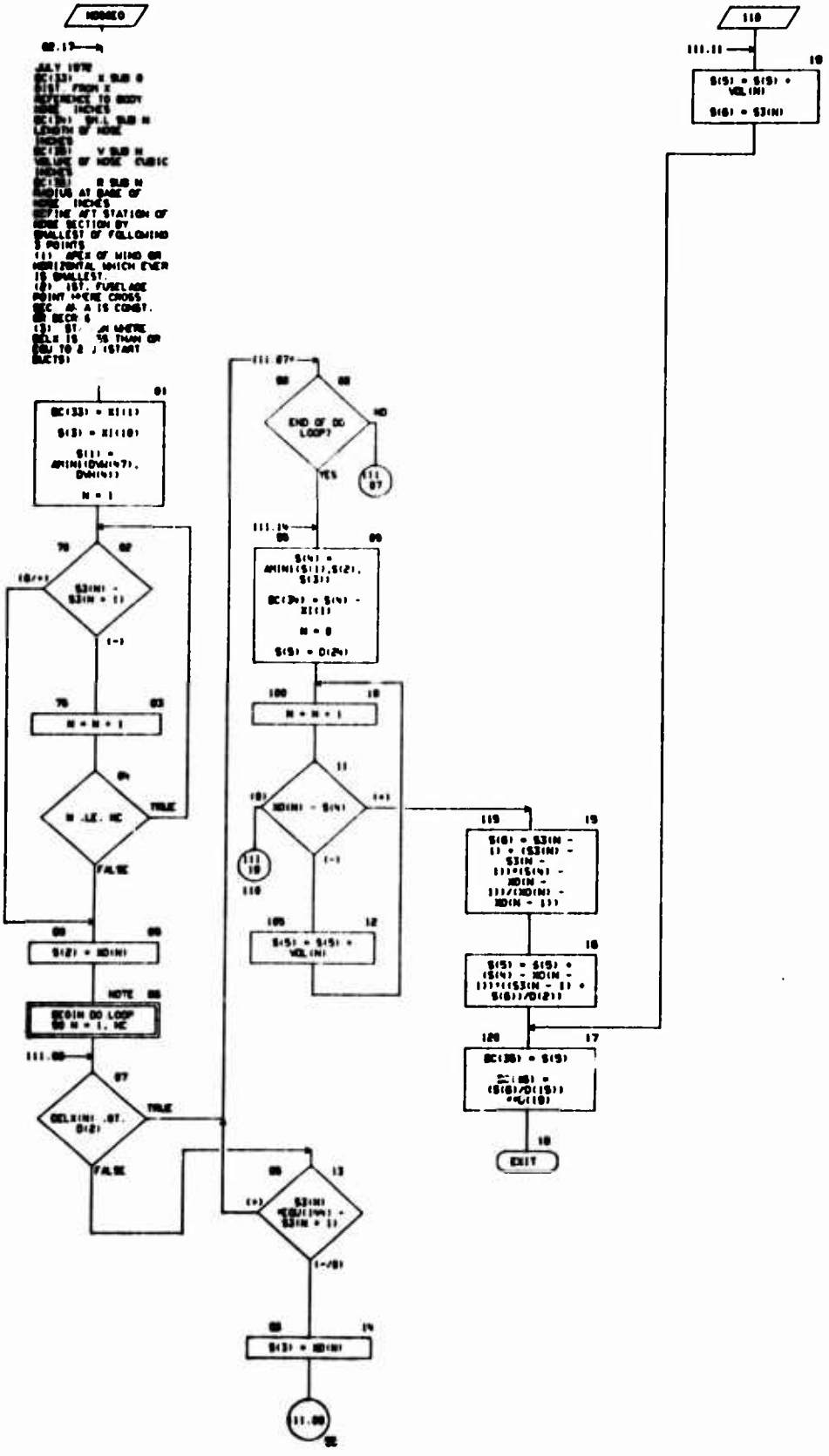


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
 COMMON TCOM(4320)
DIMENSION B(700),GD(700),DV(2320),S(400),ND(200)
, GDD(80), DVB(440), BC(200), DVA(50), DVH(30)
, BELX(20), VBL(20), S3(20), XD(20), XI(10), EQU(200)
EQUIVALENCE (TCOM(1),B(1)), (TCOM(701),GD(1)), (TCOM(1401),DV(1))
, (TCOM(3721),S(1)), (TCOM(4121),ND(1))
, (DV(321),DVA(1)), (DV(371),DVH(1)), (DV(431),DVB(1))
, (DVB(161),BELX(1)), (DVB(201),VBL(1)), (DVB(241),S3(1))
, (GD(301),GDD(1)), (GDD(6),XI(1)), (GDD(96),XD(1))
, (ND(119),NC)
, (DV(2121),BC(1)), (EQU(1),B(8))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*********************************************************************
C                      SUBROUTINE PRTOUT
C*********************************************************************
```

CHART TITLE - SUBROUTINE PRTDAT

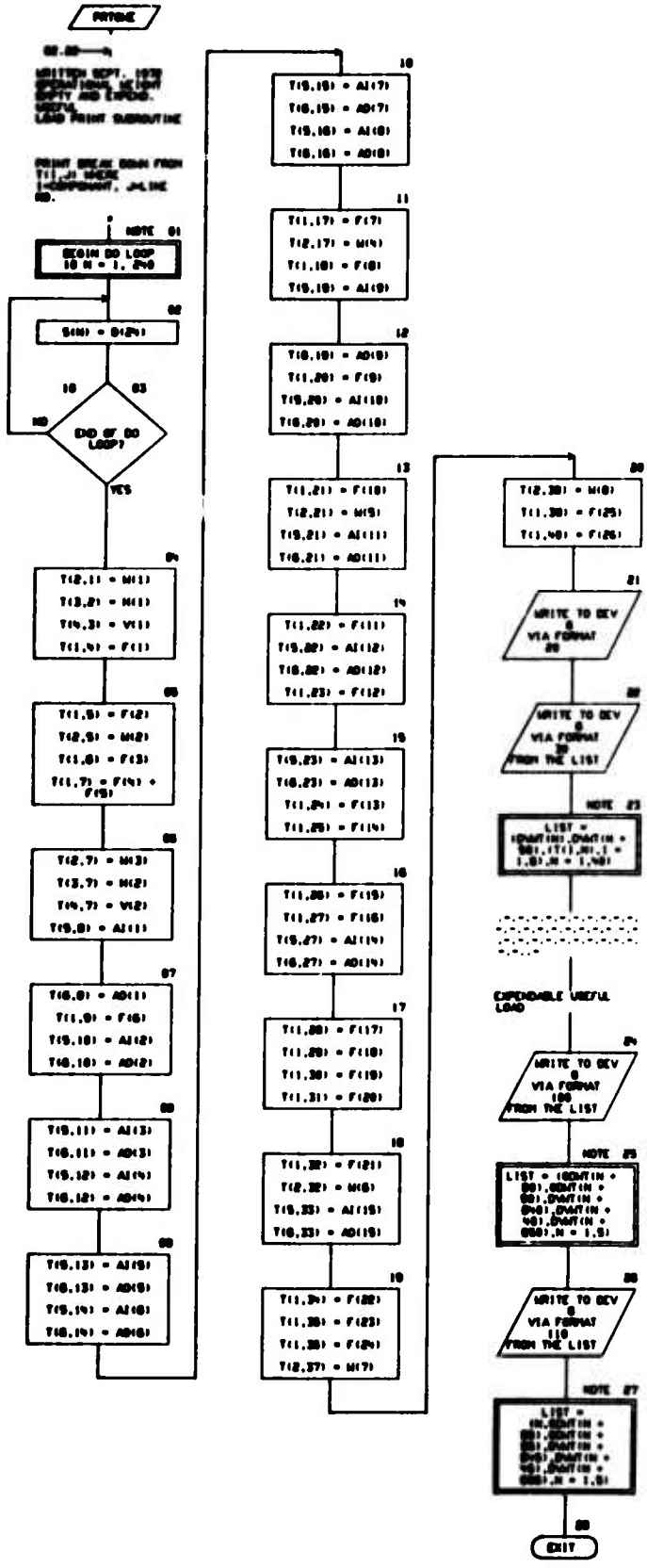


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700), GD(700), DV(2320), S(400), AD(200)
      DIMENSION GDMT(100)
      DIMENSION DVMT(1000)
      DIMENSION F(40), W(20), H(10), V(10), AJ(20), AD(20)
      DIMENSION T(6,40)
      EQUIVALENCE (D(1),TCOM(1)), (GD(1),TCOM(701)), (DV(1),TCOM(1401)),
     1(S(1),TCOM(3721)), (AD(1),TCOM(4121))
      EQUIVALENCE (GDMT(1),GD(9))
      EQUIVALENCE (DVMT(1),DV(1121))
      EQUIVALENCE (F(1),DVMT(101)), (W(1),DVMT(181)), (H(1),DVMT(221)),
     1(V(1),DVMT(241)), (AJ(1),DVMT(251)),(AD(1),DVMT(301))
      EQUIVALENCE (T(1,1),S(1))
   20 FORMAT(1H1,69X,21H** PRTOBE - IP(46) **/
     1       59H* DATA MANAGEMENT *** OPERATIONAL WEIGHT EMPTY *** /
     1 10X, 38HTOTAL AND MAJOR COMPONANT BREAK DOWN // 20X,
     1 18HTOTAL WT. 8X, 3HARM 4X, 8HFUSELAGE 8X, 4HWING 2X,
     1 18HHORIZONTAL 3X, 8HVERTICAL 11H IB NACELLE 11H OB NACELLE )
   30 FORMAT(5H WING 12X,6F11.1 /11H HORIZONTAL 6X,6F11.1 /9H VERTICAL
     1 8X,6F11.1 /9H BODY 12X,6F11.1 /10H MAIN GEAR 7X,6F11.1 /
     1 10H NOSE GEAR 7X,6F11.1 /14H SURF. CONTROL 3X,6F11.1 /
     1 13H ENG. SECTION 4X,6F11.1 /17H OTHER STRUCTURE 6F11.1 /
     1 9H ENGINES 9X,6F11.1 /17H ACCESSORY G BOX 6F11.1 /
     1 14H AIS STRUCTURE 3X,6F11.1 /17H AIS ACT AND REC .6F11.1 /
     1 8H EXHAUST 9X,6F11.1 / 17H COOL. AND DRNG. .6F11.1 /
     1 12H LUBE SYSTEM 5X,6F11.1 / 12H FUEL SYSTEM 5X,6F11.1 /
     1 17H ENGINE CONTROLS . 6F11.1 /
     1 14H START-IG SYS. 3X,6F11.1 / 8H A P U 11X,6F11.1/12H INSTRUMENTS
     1 9X,6F11.1/1 ' HYDRAULICS6X,6F11.1/11H ELECTRICAL 6X,6F11.1 /
     1 11H ELECTRONIC 6X,6F11.1 / 8H ARMAMENT 8X,6F11.1 /12H FURNISHINGS
     1 9X,6F11.1 /15H AIR CONDITION. 2X,6F11.1 / 7H PHOTO. 10X,6F11.1 /
     1 10H AUX. GEAR 7X,6F11.1 / 17H OTHER EQUIPMENT 6F11.1 / 5H CREW
     1 12X,6F11.1 / 11H TRAP. FUEL 6X,6F11.1 / 4H OIL 13X,6F11.1 /
     1 4H LUB 13X,6F11.1 / 14H MISCELLANEOUS 3X,6F11.1 / 5H GUNS 12X,
     1 6F11.1 / 10H M PYLONS 7X,6F11.1 /14H M. EXT. TANKS 3X,6F11.1 /
     1 10H F PYLONS 7X,6F11.1 /14H F. EXT. TANKS 3X,6F11.1 )
  100 FORMAT(1H1,69X,21H** PRTOBE - IP(46) **//
     1       44H DATA MANAGEMENT --- EXPENDABLE USEFUL LOAD //3HX,
     2 3HX------- CAPACITY -------, 8X,4HTOON,8X,4HTOON,8X,4HLOON / 3HX,
     1 6HWEIGHT,9X, 3HARM // 13X, 21HPASSENGERS OR PAYLOAD, 5F12.2 /
     1 13X, 18HWING PAYLOAD,9X, 5F12.2 / 13X, 18HAMMUNITION, 11X,
     1 5F12.2 / 13X, 17HWING FUEL TANK 1, 4X, 5F12.2 / 13X,
     1 17HWING FUEL TANK 2, 4X, 5F12.2 )
  110 FORMAT ( 13X, 18HFUSELAGE FUEL TANK, 12, X, 5F12.2 )
```

CHART TITLE - INTRODUCTORY COMMENTS

```
**********************************************************************
                        SUBROUTINE SWEEP
**********************************************************************
```

CHART TITLE - SUBROUTINE QUICK

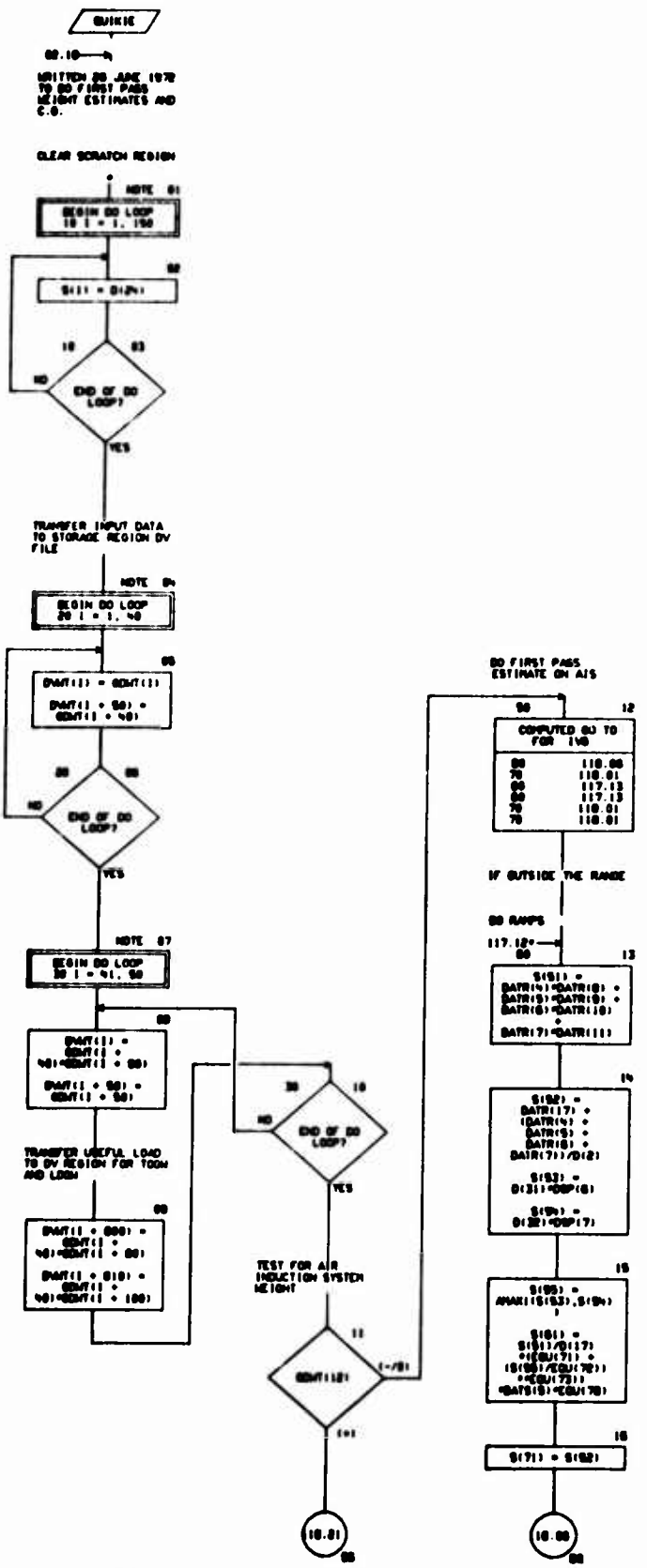


CHART TITLE - SUBROUTINE GUESS

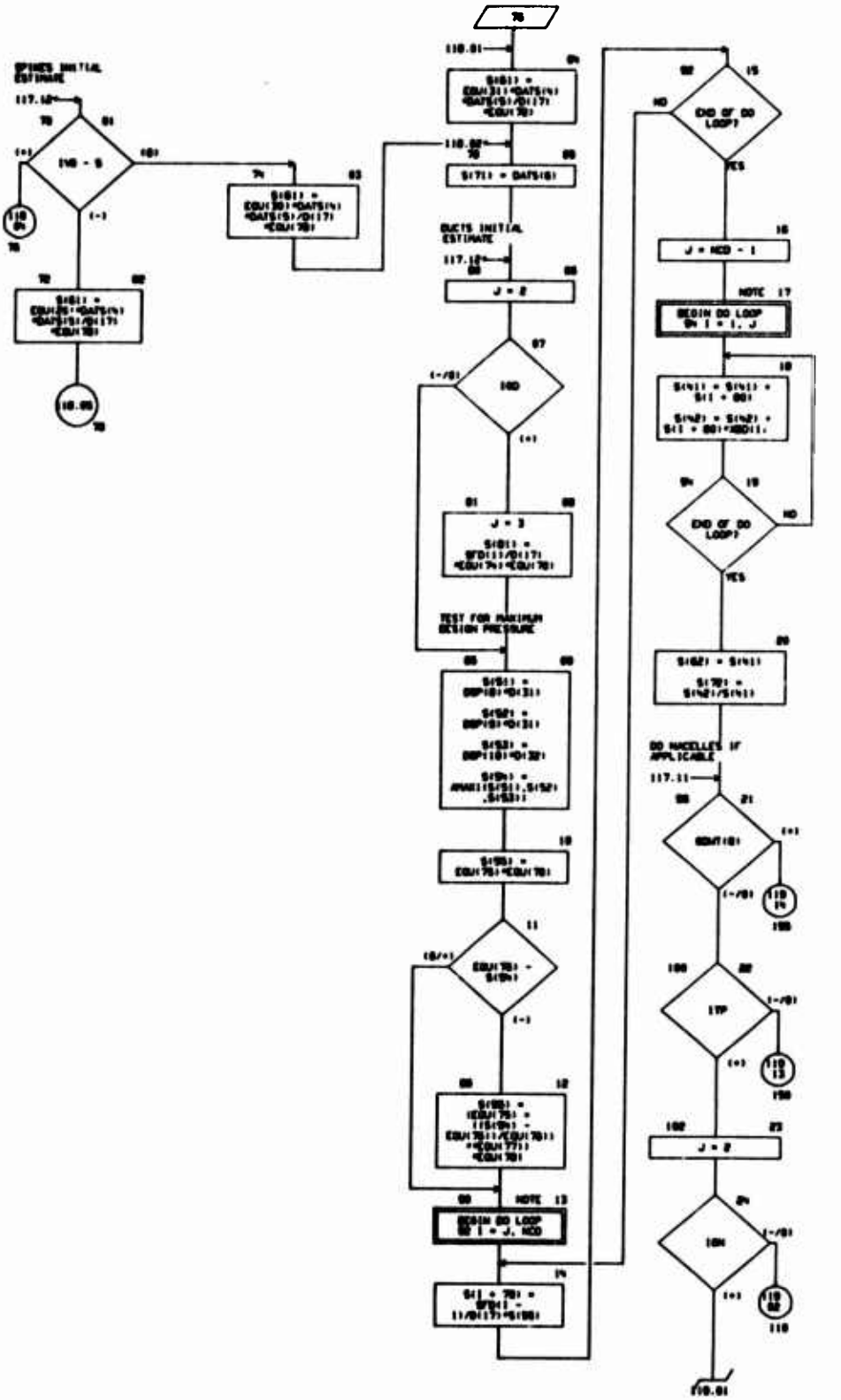


CHART TITLE - SUBROUTINE QUIKIE

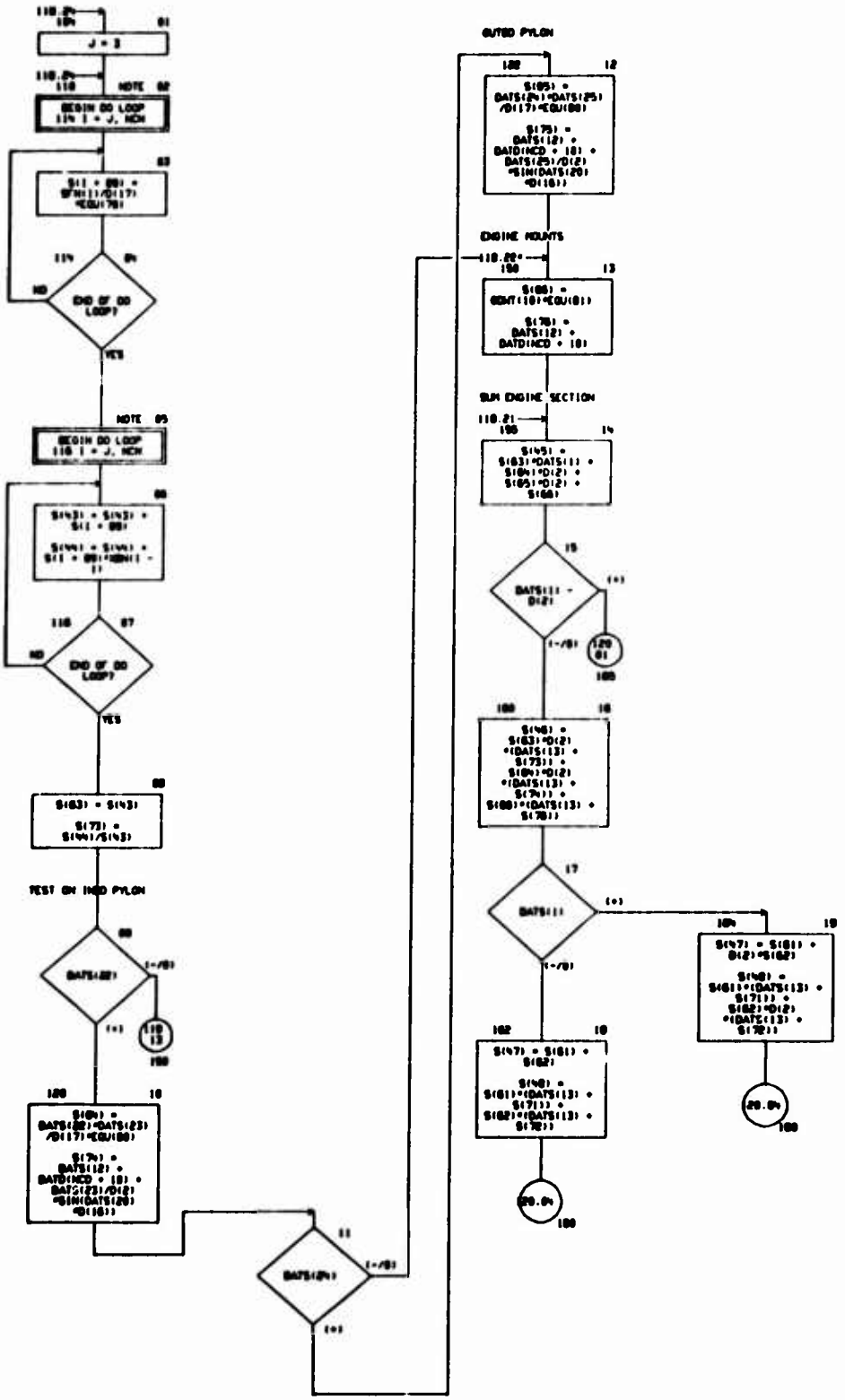


CHART TITLE - SUBROUTINE QUIKIE

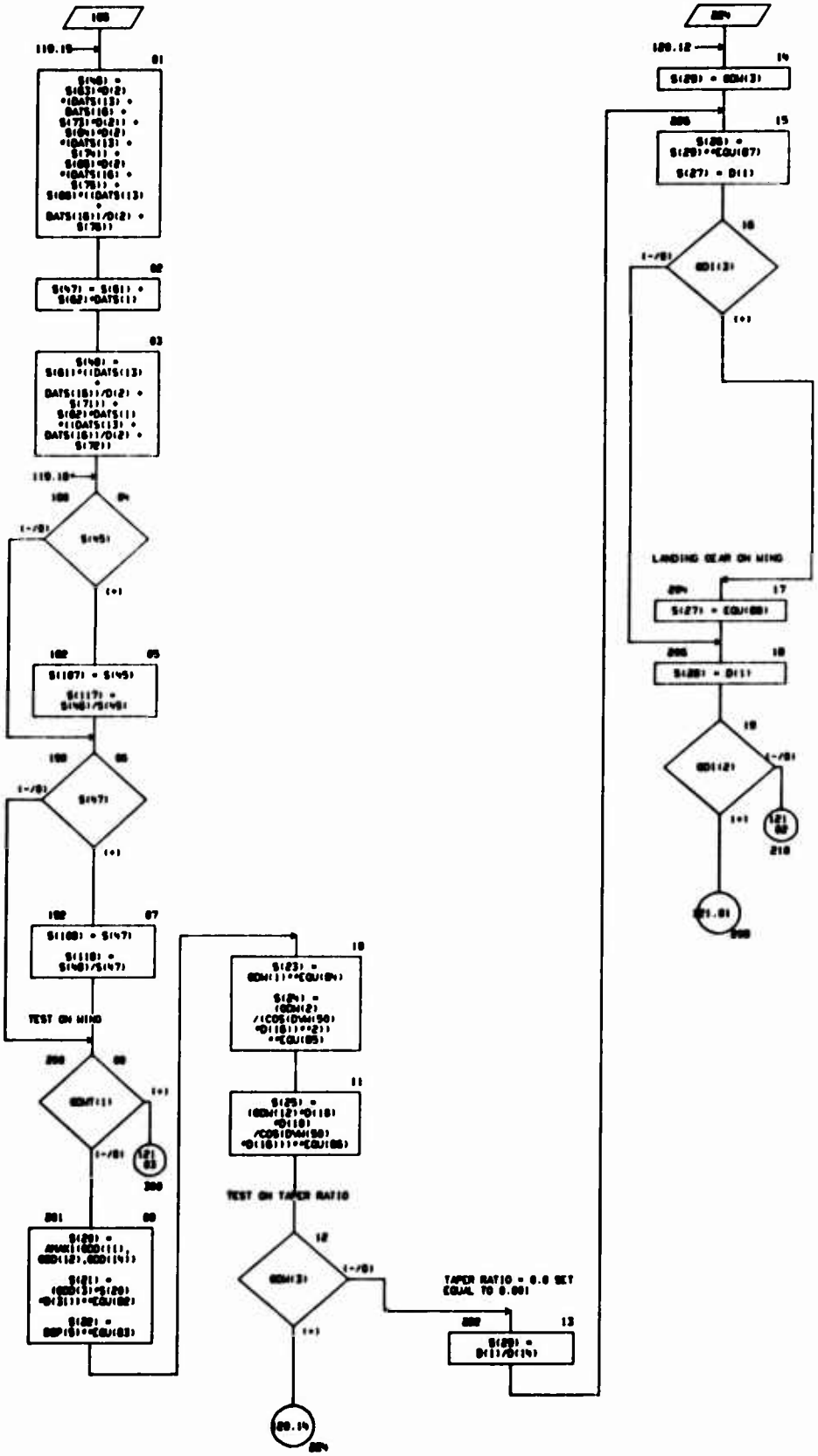


CHART TITLE - SUBROUTINE QUICK

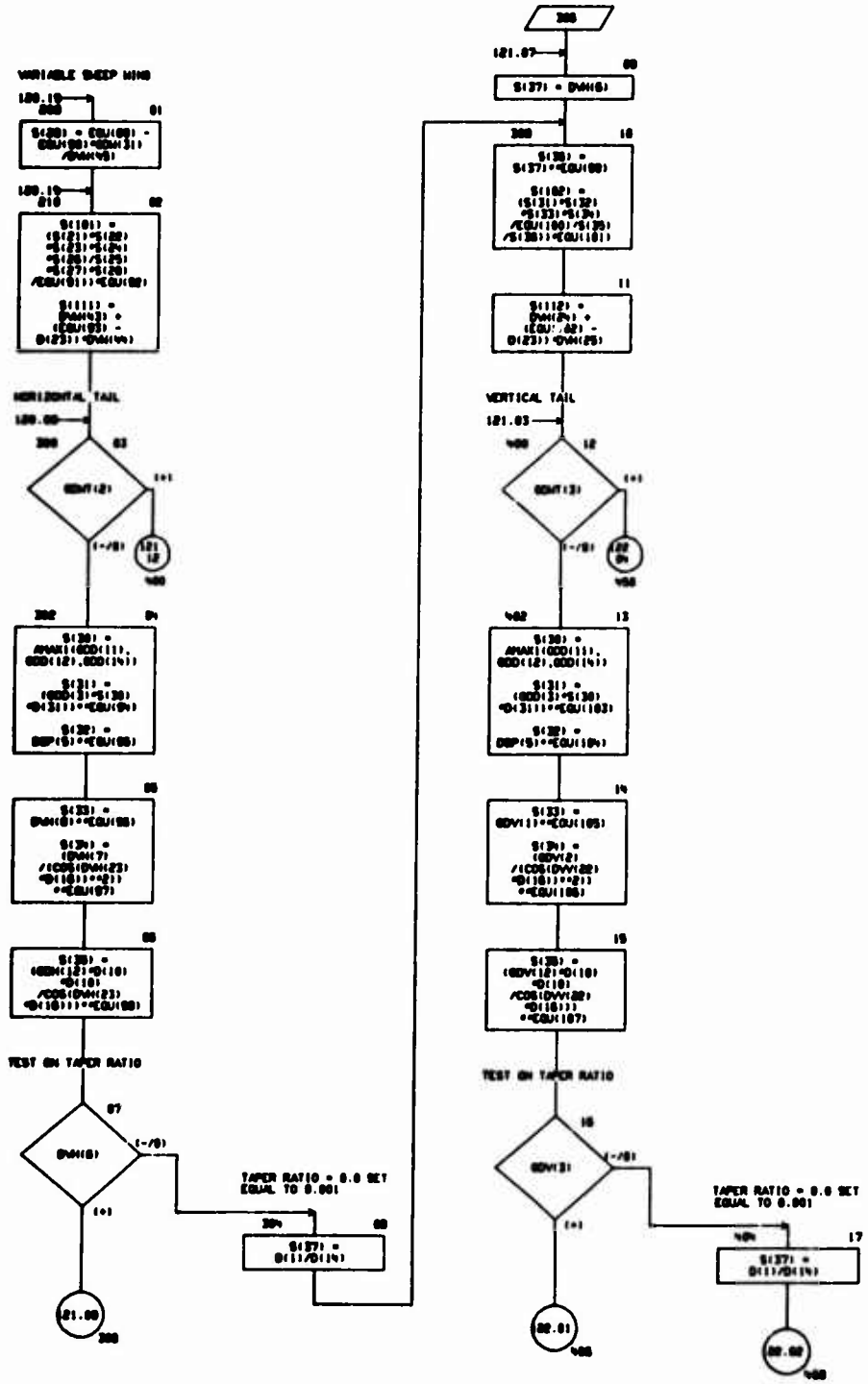


CHART TITLE - SUBROUTINE QUICK

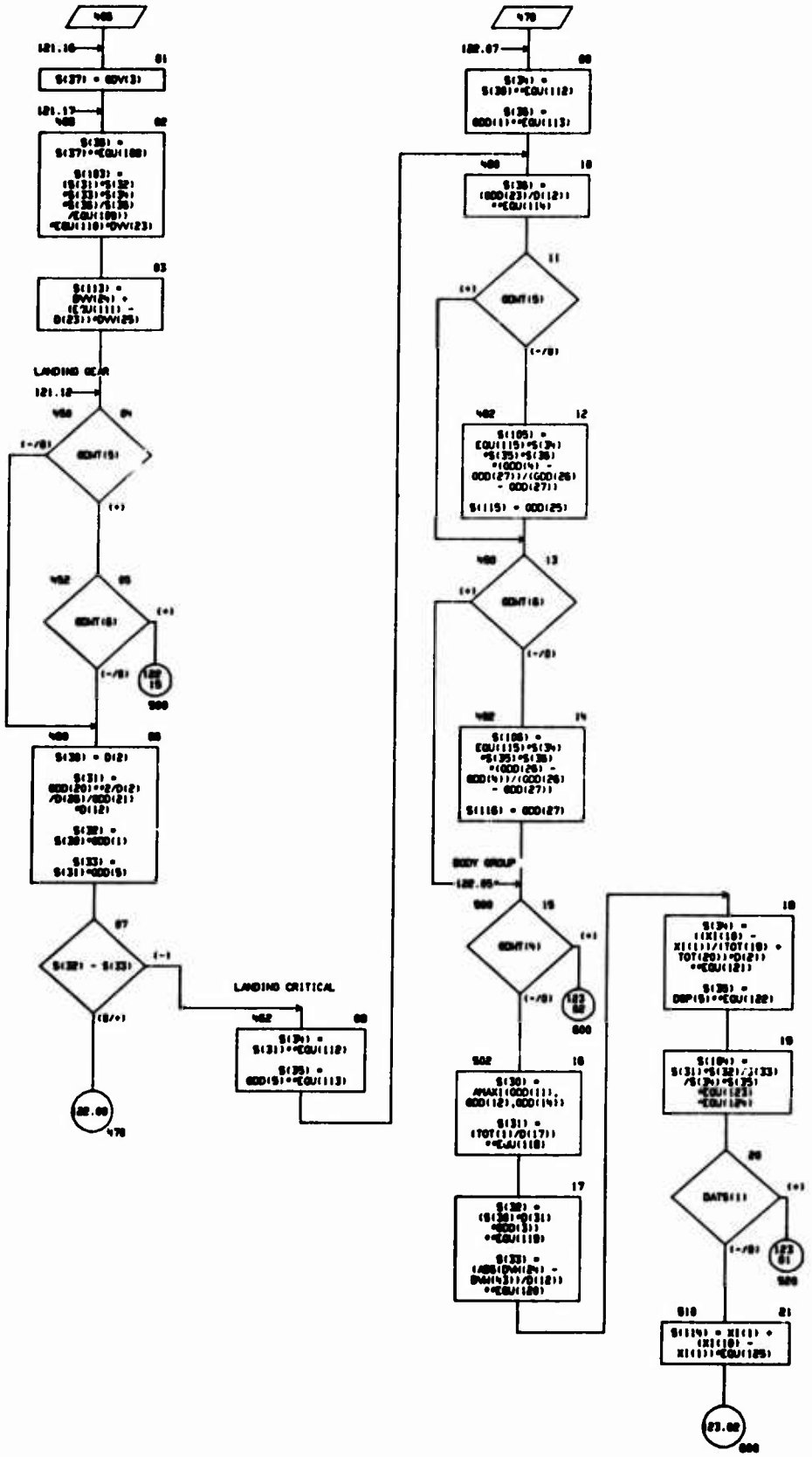


CHART TITLE - SUBROUTINE GUIKIE

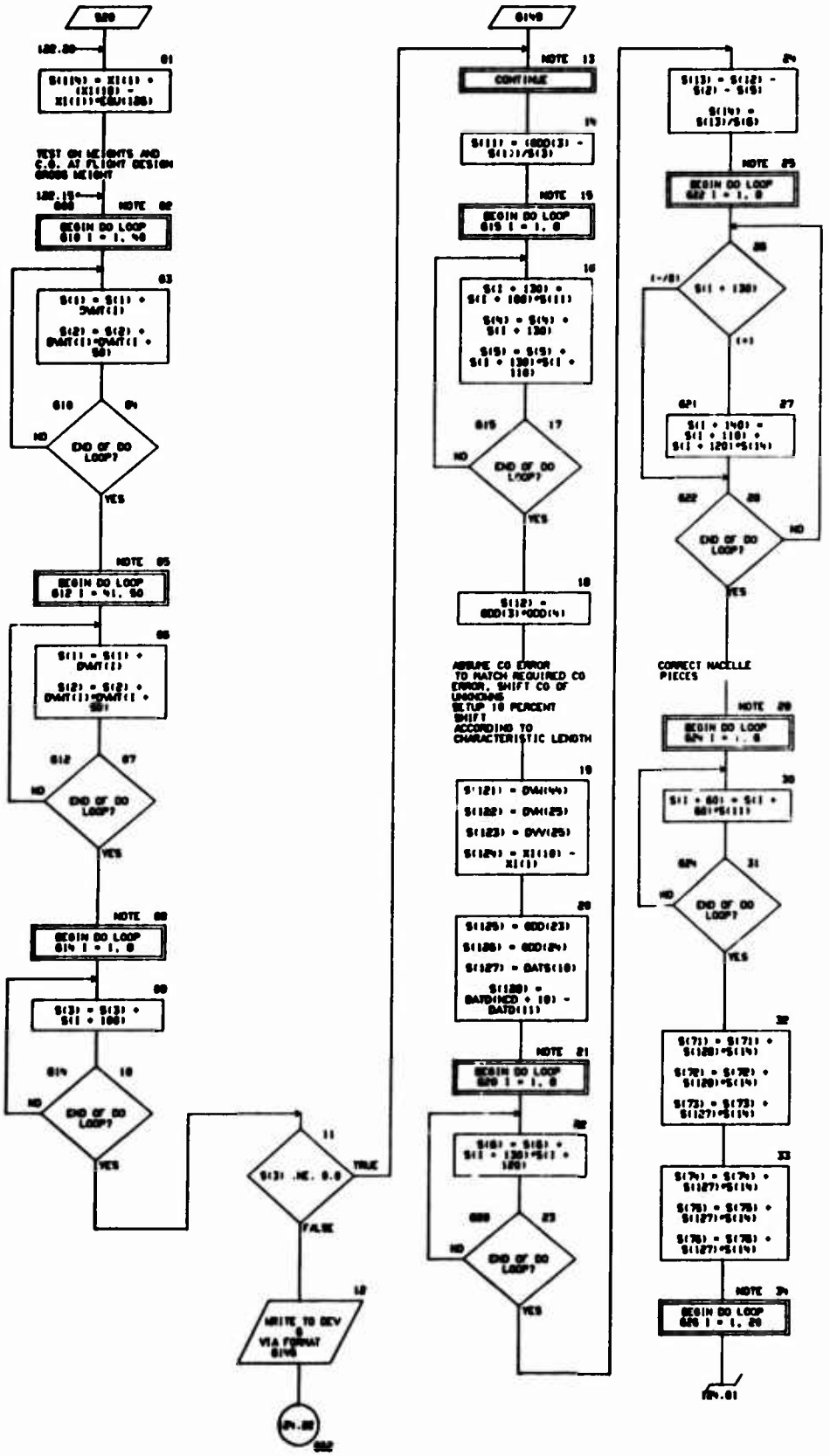


CHART TITLE - SUBROUTINE QUIKIE

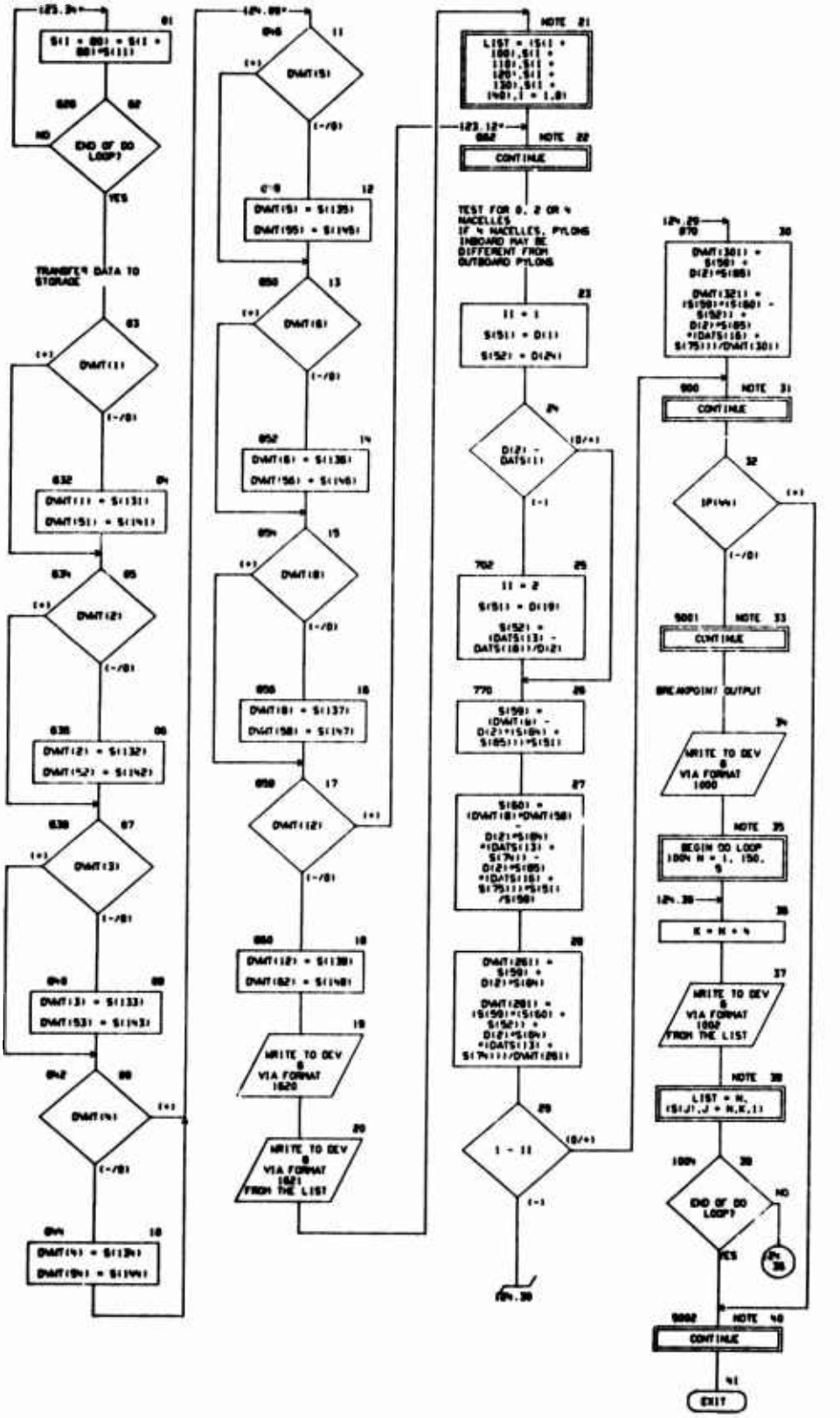


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION GDI(20),GDD(30),GDWT(160),GDM(50),GDH(40),GDV(40),
     GDB(80),DATS(40),DATD(70),DATR(20),DATN(70)
      DIMENSION DVH(50),DVW(30),DVV(30),DVB(440),DVD(100),DVN(150),
     DWAT(1000)
      DIMENSION DSP(10)
      DIMENSION X(10)
      DIMENSION TOT(20),XBD(10),SFD(10),XBN(10),SFN(10)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (EQU(1),D(81))
      EQUIVALENCE (GDI(1),GD(1)),(GDD(1),GD(21)),(GDWT(1),GD(81)),
     (GDM(1),GD(251)),(GDH(1),GD(301)),(GDV(1),GD(341)),
     (GDB(1),GD(381)),(DATS(1),GD(461)),(DATD(1),GD(501)),
     (DATR(1),GD(571)),(DATN(1),GD(591))
      EQUIVALENCE (X(1),GDB(6))
      EQUIVALENCE (DVH(1),DV(321)),(DVW(1),DV(371)),(DVV(1),DV(401)),
     (DVB(1),DV(431)),(DVD(1),DV(871)),(DVN(1),DV(971)),
     (DWAT(1),DV(1121))
      EQUIVALENCE (DSP(1),DV(31))
      EQUIVALENCE (TOT(1),DVB(361)),(SFD(1),DVD(71)),
     (XBD(1),DVD(81)),(SFN(1),DVN(71)),(XBN(1),DVN(111))
      EQUIVALENCE (I,ND(101)),(J,ND(102))
      EQUIVALENCE (II,ND(107))
      EQUIVALENCE (ITP,ND(111)),(IVG,ND(112)),(IGD,ND(114)),
     (NC,ND(115)),(NCD,ND(117)),(NCN,ND(119)),(ION,ND(121))
0145  FORMAT(/// 6X40********** ALL DETAIL WEIGHTS AND C.G.S WERE INPUT
     ********** ///)
1620  FORMAT(1HI,28X,52HSTRUCTURE WEIGHT DATA FROM QUIKIE IN DATA MANAGE
     MENT//32X,18HINITIAL ESTIMATE,4X,14HCHARACTERISTIC,8X,
     18HCORRECTED ESTIMATE/30X,6HWEIGHT,5X,9HHORIZ ARM,6X,6HLENGTH,
     13X,6HWEIGHT,5X,9HHORIZ ARM)
1621  FORMAT(9X,4HWING,11X,2F12.1,4X,F12.1,4X,2F12.1/
     9X,19HHORIZONTAL TAIL,2F12.1,4X,F12.1,4X,2F12.1/
     9X,13HVERTICAL TAIL,2X,2F12.1,4X,F12.1,4X,2F12.1/
     9X,6HFUSELAGE,7X,2F12.1,4X,F12.1,4X,2F12.1/
     9X,9HMAIN GEAR,6X,2F12.1,4X,F12.1,4X,2F12.1/
     9X,9HNOSE GEAR,6X,2F12.1,4X,F12.1,4X,2F12.1/
     9X,14HENGINE SECTION,1X,2F12.1,4X,F12.1,4X,2F12.1/
     9X,15HAIR IND. SYSTEM,2F12.1,4X,F12.1,4X,2F12.1)
1000  FORMAT(1HI,32X,11H  S-REGION,45X,21H** QUIKIE - IP(44) **)
1002  FORMAT(4X,I4,9F10.4)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*****************************************************************
C                    SUBROUTINE SPDAL7
C*****************************************************************
```

CHART TITLE - SUBROUTINE SPDALT

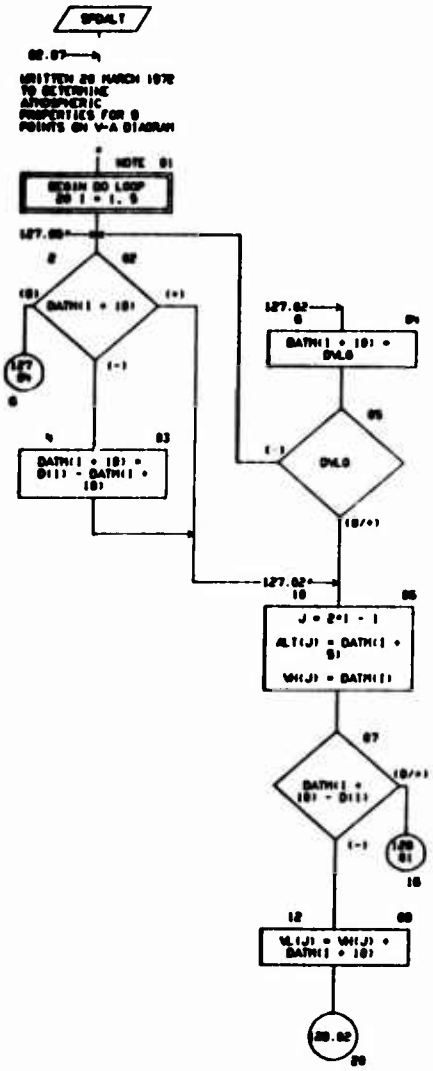


CHART TITLE - SUBROUTINE SPDALT

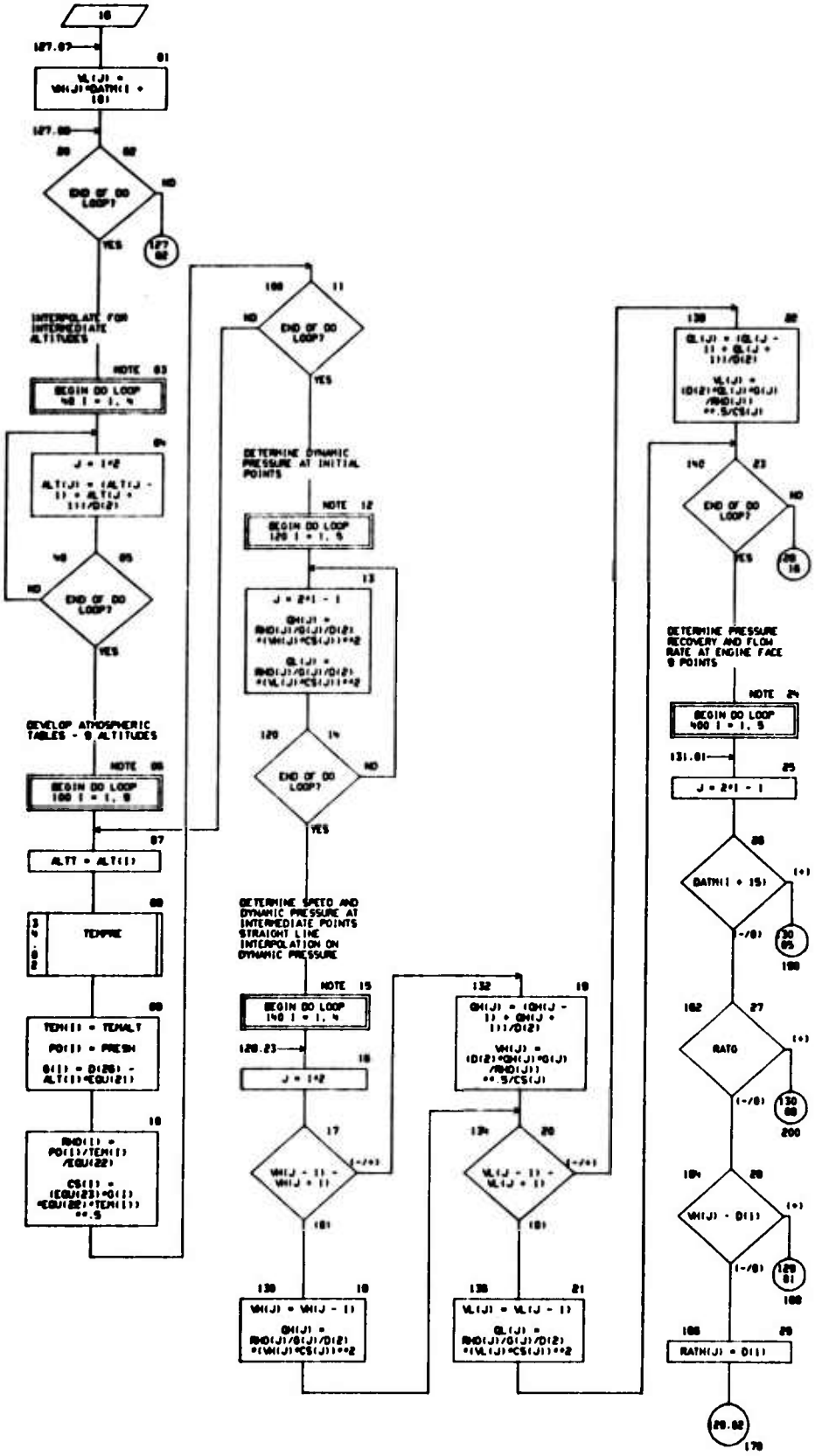


CHART TITLE - SUBROUTINE SPDALT

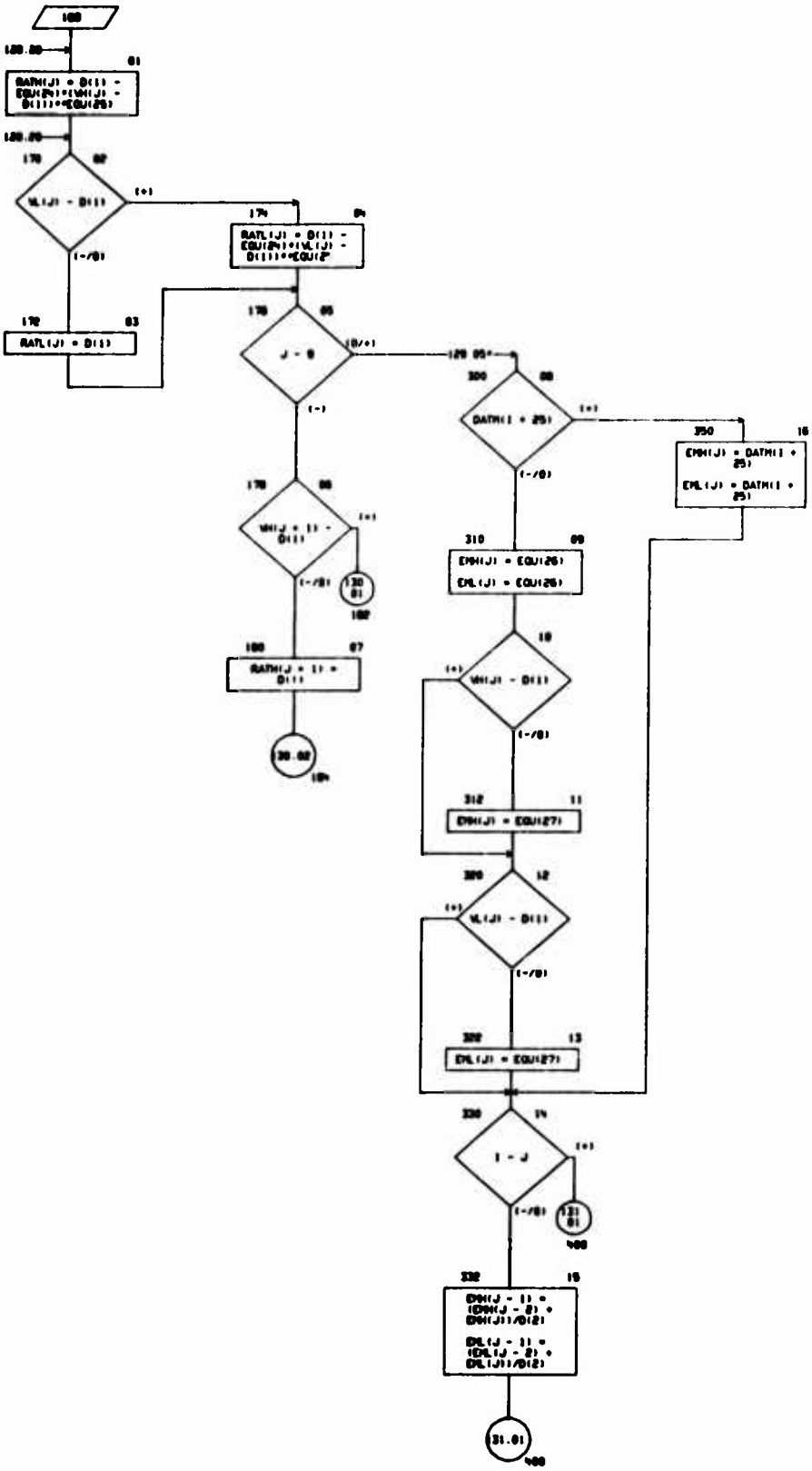


CHART TITLE - SUBROUTINE SPDALT

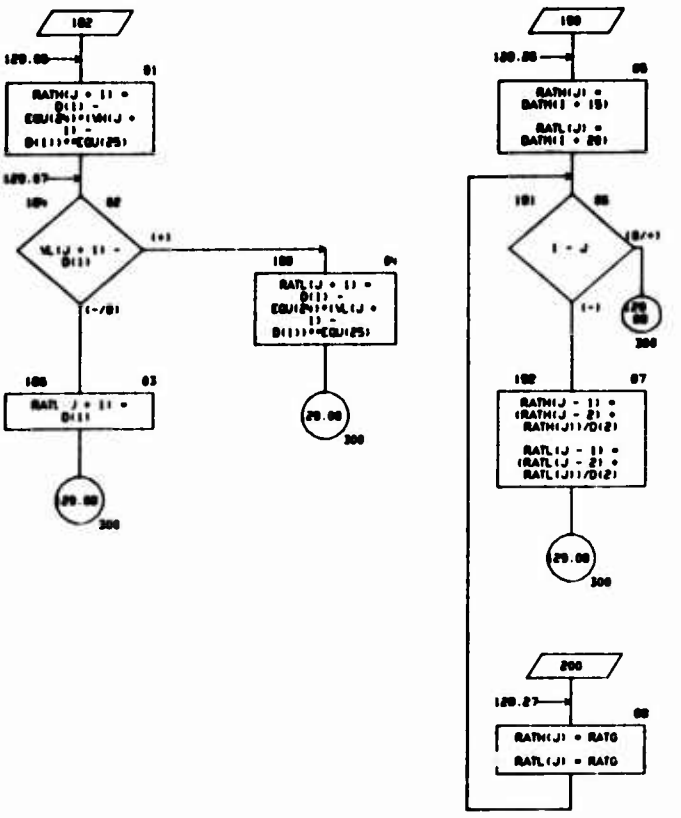


CHART TITLE - SUBROUTINE SPDALT

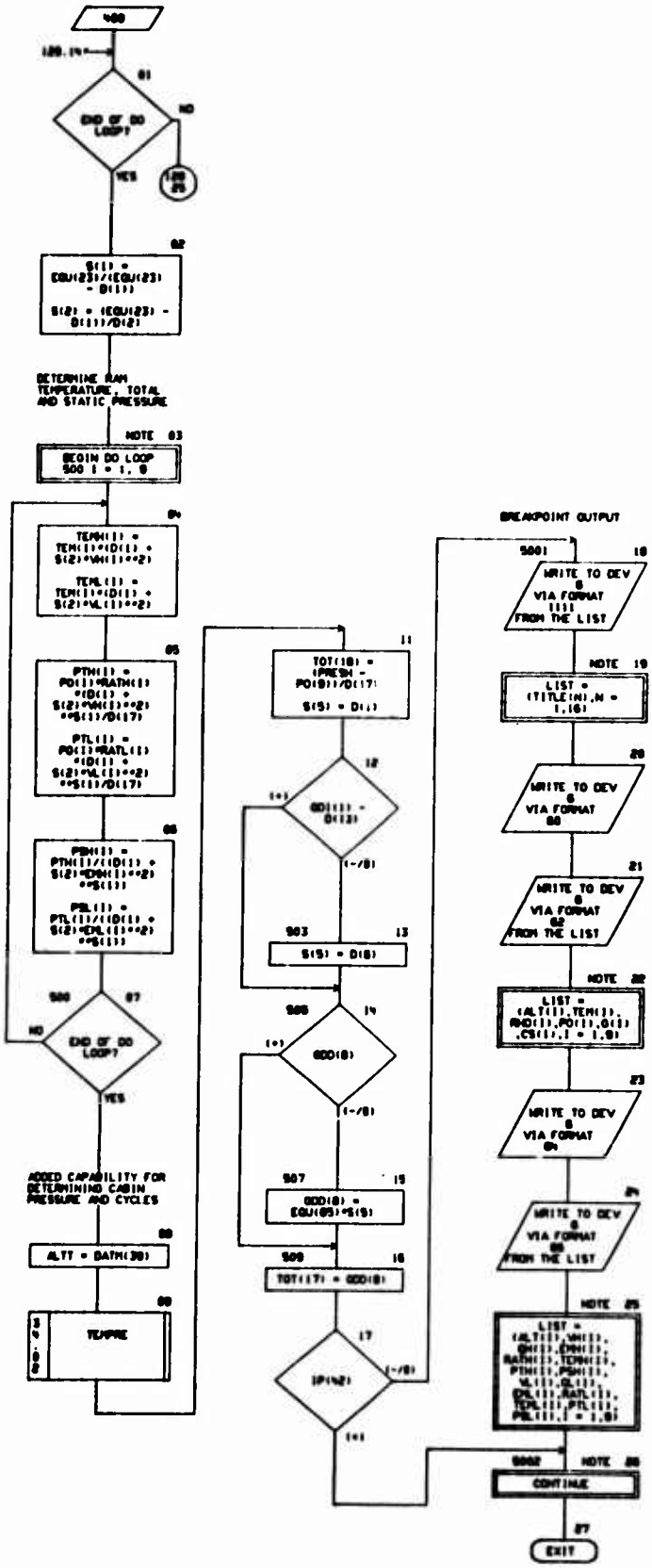


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
      COMMON /MISC/ XMISC(100)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION GD1(20),GDD(30)
      DIMENSION DATM(40)
      DIMENSION EQU(200)
      DIMENSION TITLE(16)
      DIMENSION ALT(10),TEM(10),PO(10),G(10),CS(10),RHO(10)
      DIMENSION VM(10),VL(10),QM(10),QL(10),EMM(10),EML(10),
     &RATM(10),RATL(10),TEMM(10),TEML(10),PTM(10),PTL(10),PSM(10),
     &PSL(10)
      DIMENSION DVB(440),TOT(20)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(70)),(DV(1),TCOM(140)),
     & (S(1),TCOM(372)),(ND(1),TCOM(412))
      EQUIVALENCE (D(61),EQU(1))
      EQUIVALENCE (GD1(1),GD(1)),(GDD(1),GD(21))
      EQUIVALENCE (DATM(1),GD(51))
      EQUIVALENCE (DATM(31),DVL0),(DATM(32),RATO)
      EQUIVALENCE (TITLE(1),XMISC(65))
      EQUIVALENCE (S(1),TEMALT),(S(2),PRESH),(S(4),ALTT)
      EQUIVALENCE (ALT(1),DV(1)),(TEM(1),DV(11)),(PO(1),DV(21)),
     & (G(1),DV(31)),(CS(1),DV(41)),(RHO(1),DV(51))
      EQUIVALENCE (VM(1),DV(61)),(VL(1),DV(71)),(QM(1),DV(81)),
     & (QL(1),DV(91)),(EMM(1),DV(101)),(EML(1),DV(111)),
     & (RATM(1),DV(121)),(RATL(1),DV(131)),(TEMM(1),DV(141)),
     & (TEML(1),DV(151)),(PTM(1),DV(161)),(PTL(1),DV(171)),
     & (PSM(1),DV(181)),(PSL(1),DV(191))
      EQUIVALENCE (DVB(1),DV(431)),(TOT(1),DVB(361))
      EQUIVALENCE (ND(101),II),(ND(102),J)
1111  FORMAT(1H1,88X,21H** SPDALT - IP(42) **/10X,8A10/10X,8A10)
99    FORMAT(1H0,38X,37H*** SPEED ALTITUDE PROFILE TABLES ***,
     &       //46X,19HSTANDARD ATMOSPHERE//15X,8HALTITUDE,3X,
     &11HTEMPERATURE,5X,7HDENSITY,6X,8HPRESSURE,6X,1HD,8X,
     &14HSPEED OF SOUND/17X,4HFEET,5X,11HDEG RANKINE,7X,3HPCF,10X,
     &3HPSF,5X,8HFT/SEC SQ,7X,6HFT/SEC)
98    FORMAT(12X,F11.1,F12.3,5X,F10.7, F12.2,F11.3,F14.2)
94    FORMAT(///46X,13HPROFILE TABLE/4X,
     &4HALT.,3X,4HVM(M),4X,4HQ(M),3X,3HM2,2X,7HPT2/PT0,2X,
     &9HRAM T,4X,3HPT2,4X,3HP2,4X,4HVM(L),4X,4HQ(L),3X,3HM2,2X,
     &7HPT2/PT0,2X,9HRAM T,4X,3HP12,4X,3HP2/4X,
     &4HFEET,4X,3HMN,5X,3HPSF,4X,3HMN,11X,5HDEG R,4X,3HPSI,4X,3HPSI,
     &4X,3HMN,5X,3HPSF,4X,3HMN,11X,5HDEG R,4X,3HPSI,4X,3HPSI)
95    FORMAT(F10.1,F8.2,F8.2,F8.2,F7.4,F8.2,2F7.2,F7.2,F8.2,F8.2,
     &F7.4,F8.2,2F7.2)
```

CHART TITLE - INTRODUCTORY COMMENTS

SUBROUTINE TEMPRE

CHART TITLE - SUBROUTINE TEMPRE

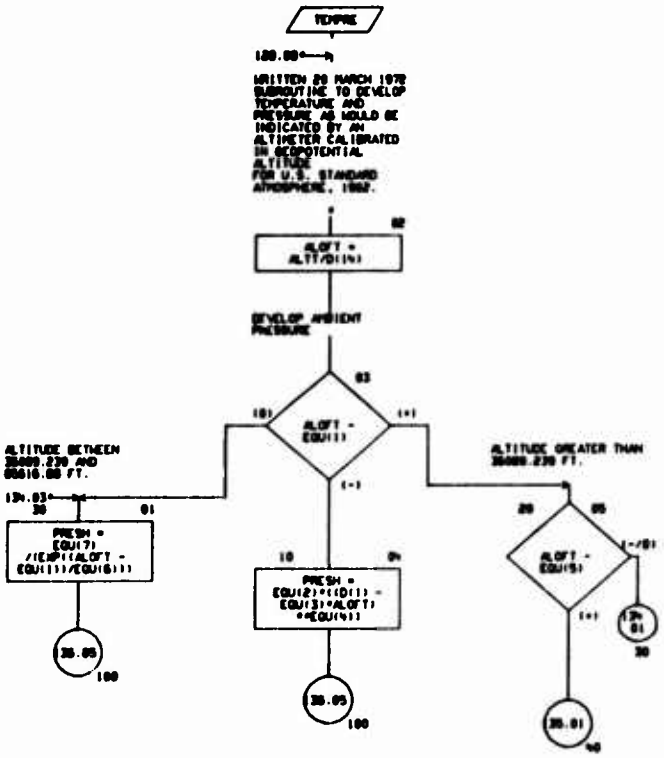


CHART TITLE - SUBROUTINE TOPPRE

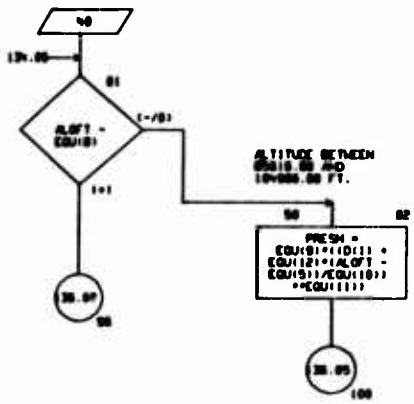


CHART TITLE - SUBROUTINE TEMPRE

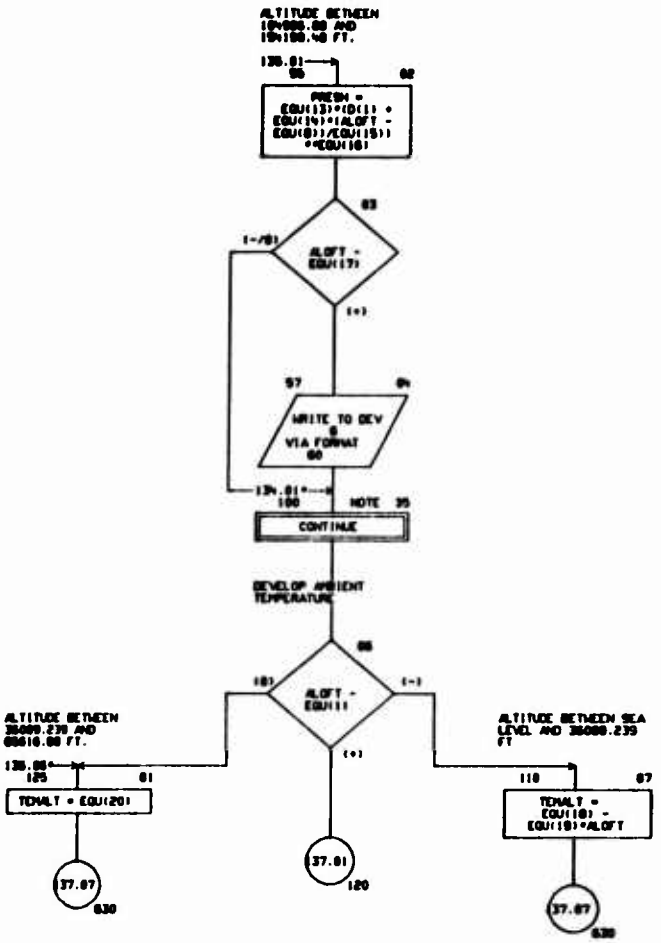


CHART TITLE - SUBROUTINE TEMPRE

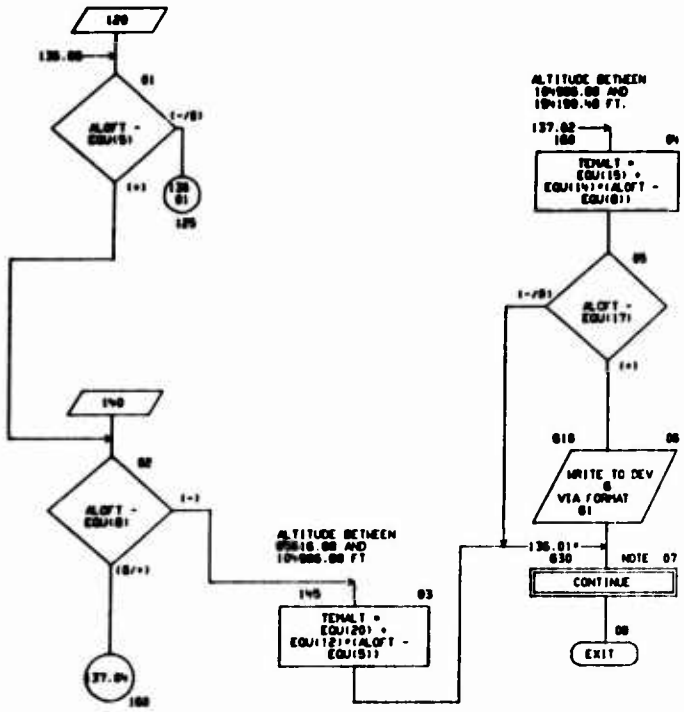


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (D(81),EQU(1))
      EQUIVALENCE (S(1),TEMALT),(S(2),PRESH),(S(3),ALOFT),(S(4),ALTI)
   88 FORMAT(1HB,6X,23H*** WARNING MESSAGE ***,16X,
     *9HALTITUDE IS BEYOND VALID RANGE OF PRESSURE)
   81 FORMAT(1HB,6X,23H*** WARNING MESSAGE ***,16X,
     *9HALTITUDE IS BEYOND VALID RANGE OF TEMPERATURE)
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*****************************************************************
C                    SUBROUTINE MEIOST
C*****************************************************************
```

CHART TITLE - SUBROUTINE WEIGHT

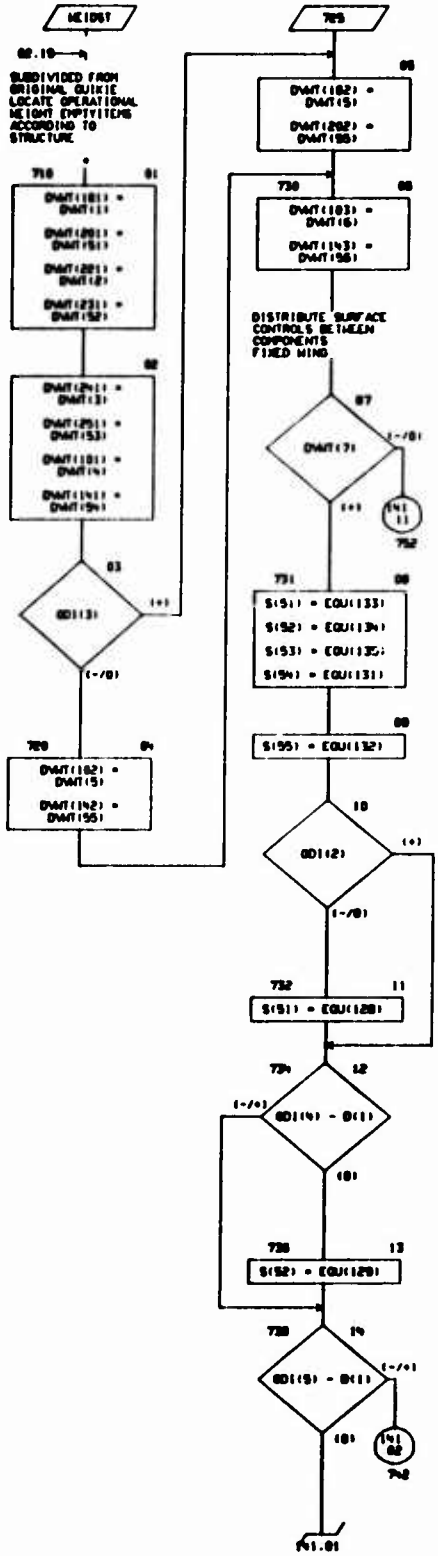


CHART TITLE - SUBROUTINE MEIOS7

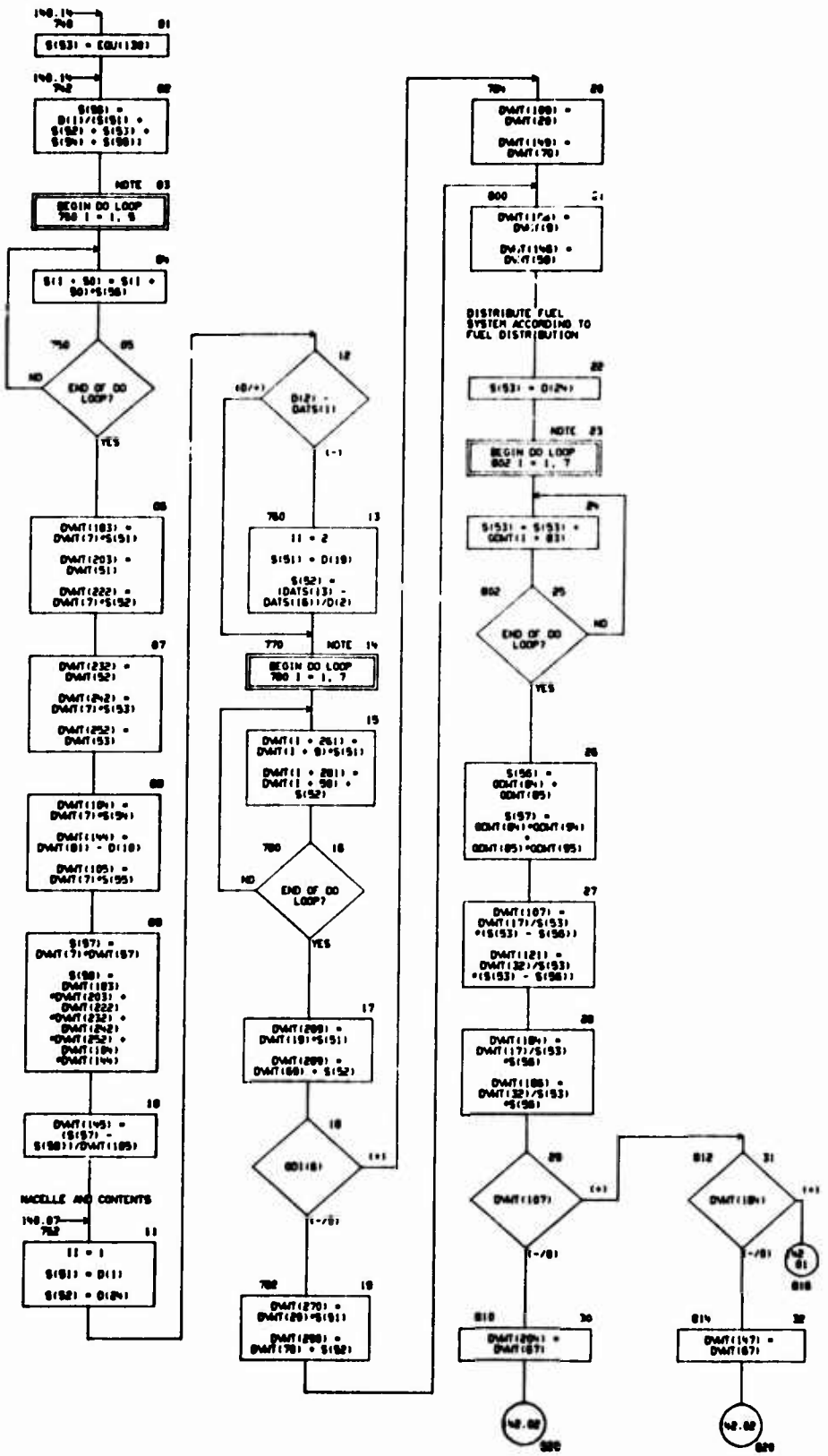


CHART TITLE - SUBROUTINE REQUEST

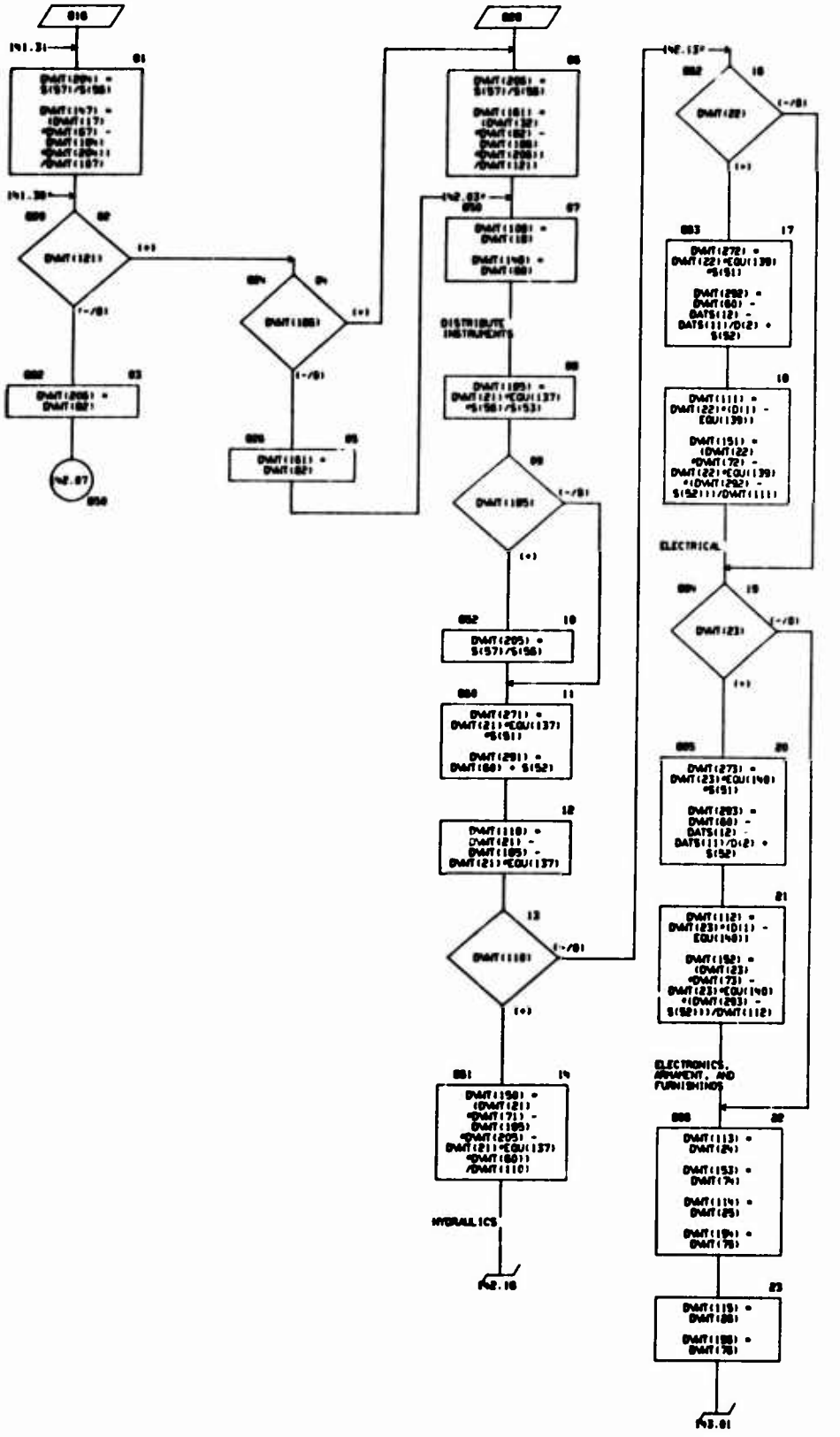


CHART TITLE - SUBROUTINE WEIGHT

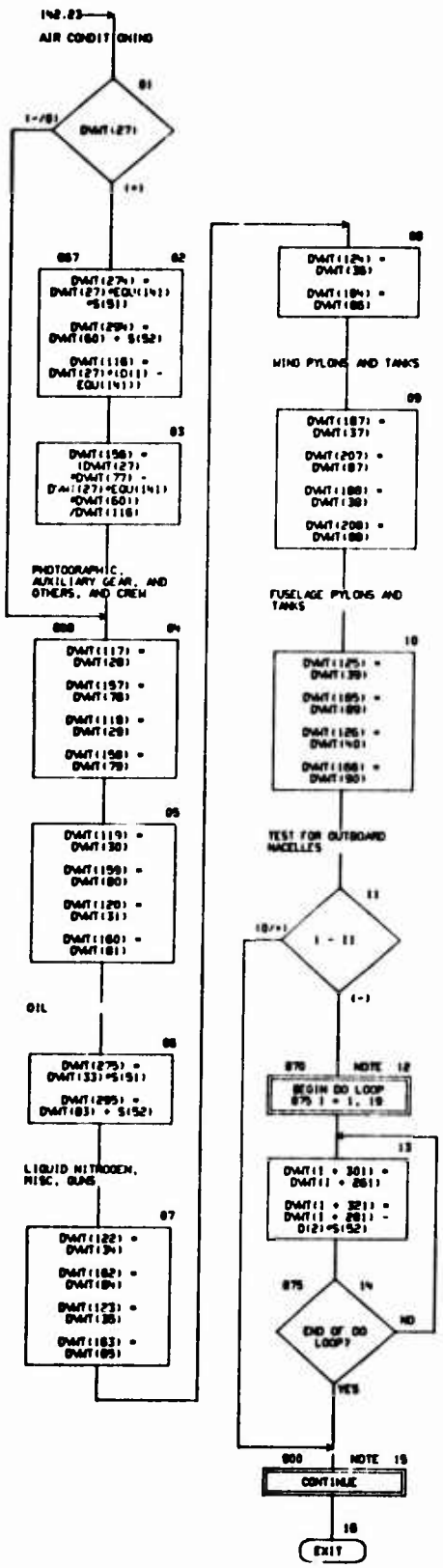


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION GD1(20),GDNT(160),DATS(40),DVNT(1000)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GD1(1),GD(1)),(GDNT(1),GD(81)),(DATS(1),GD(481))
     , (DVNT(1),DV(121))
      EQUIVALENCE (EQU(1),D(81))
      EQUIVALENCE (I,ND(181)),(J,ND(182)),(II,ND(187))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*************************************************************************
C                        SUBROUTINE MAVDCO
C*************************************************************************
```

CHART TITLE - SUBROUTINE WWWEO

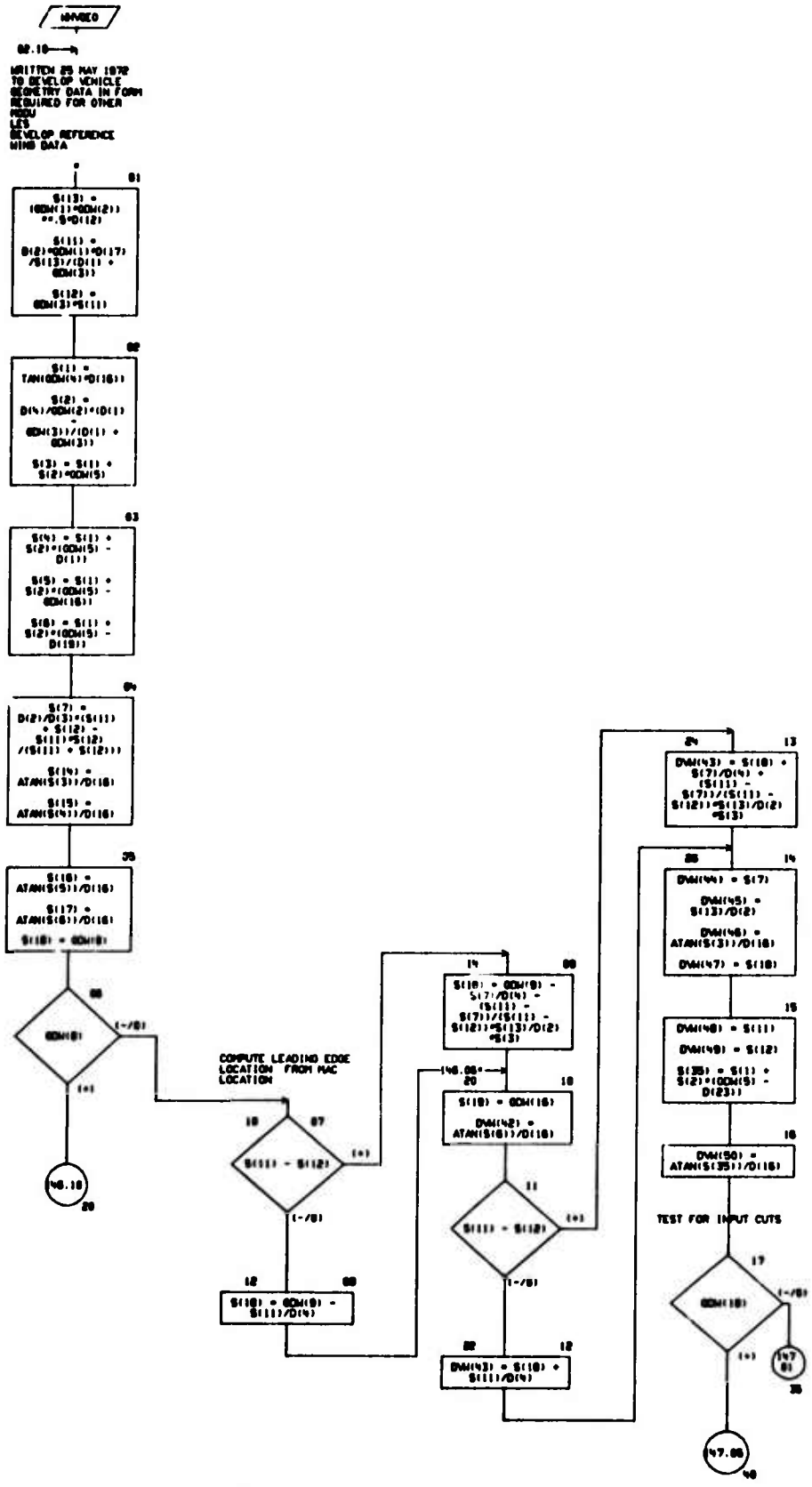


CHART TITLE - SUBROUTINE MAKEDS

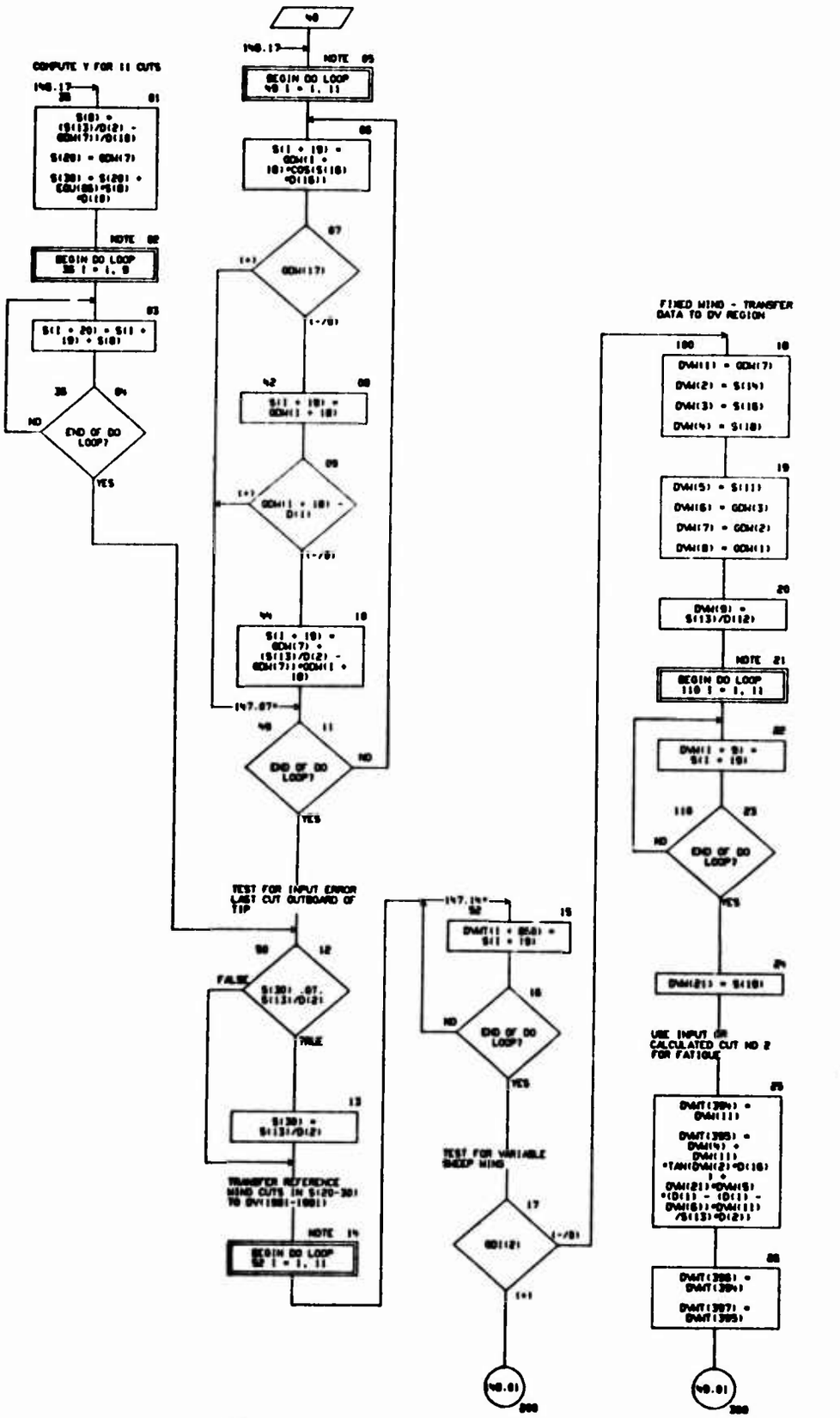


CHART TITLE - SUBROUTINE MAKDES

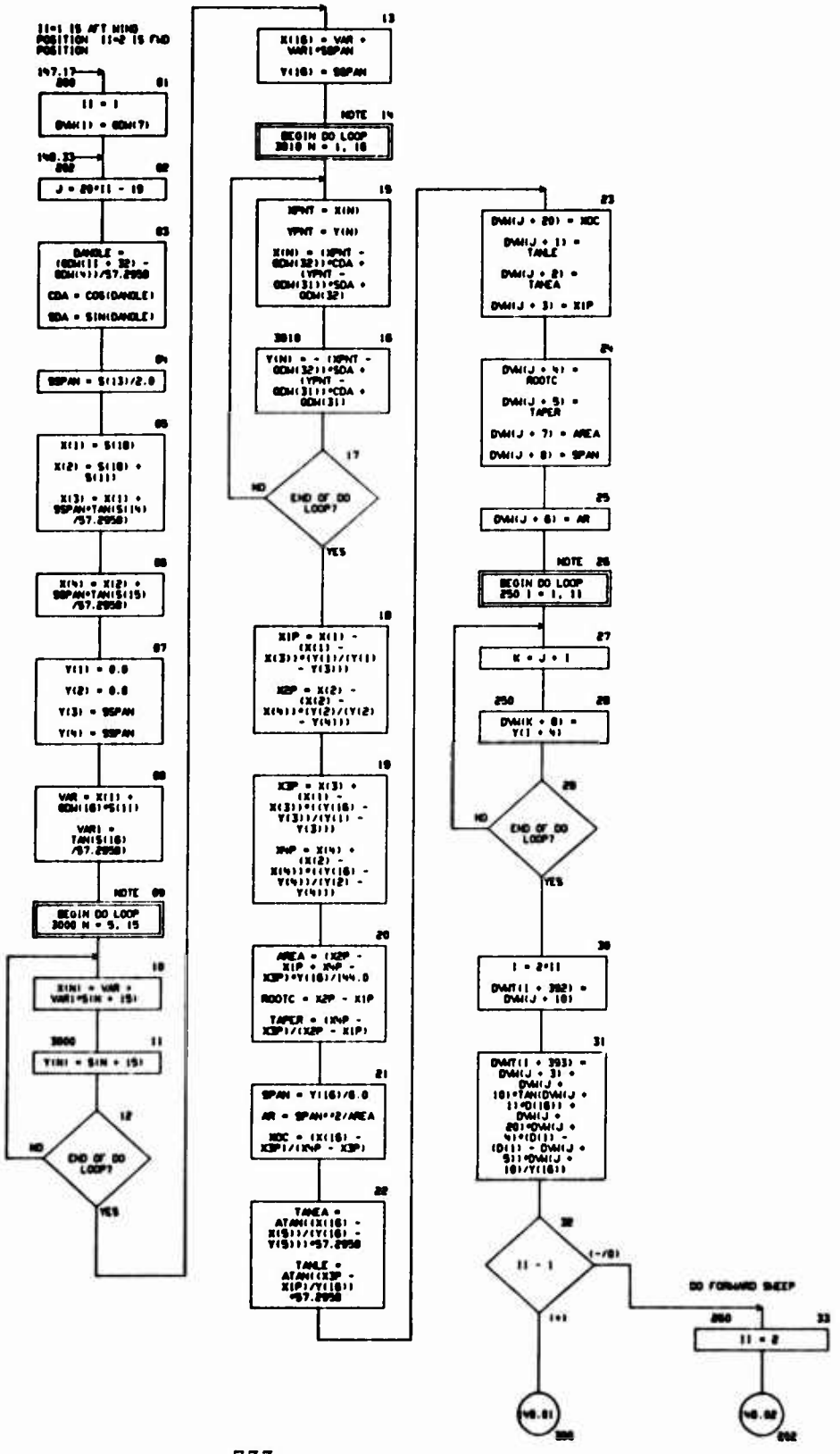


CHART TITLE - SUBROUTINE HAVGEO

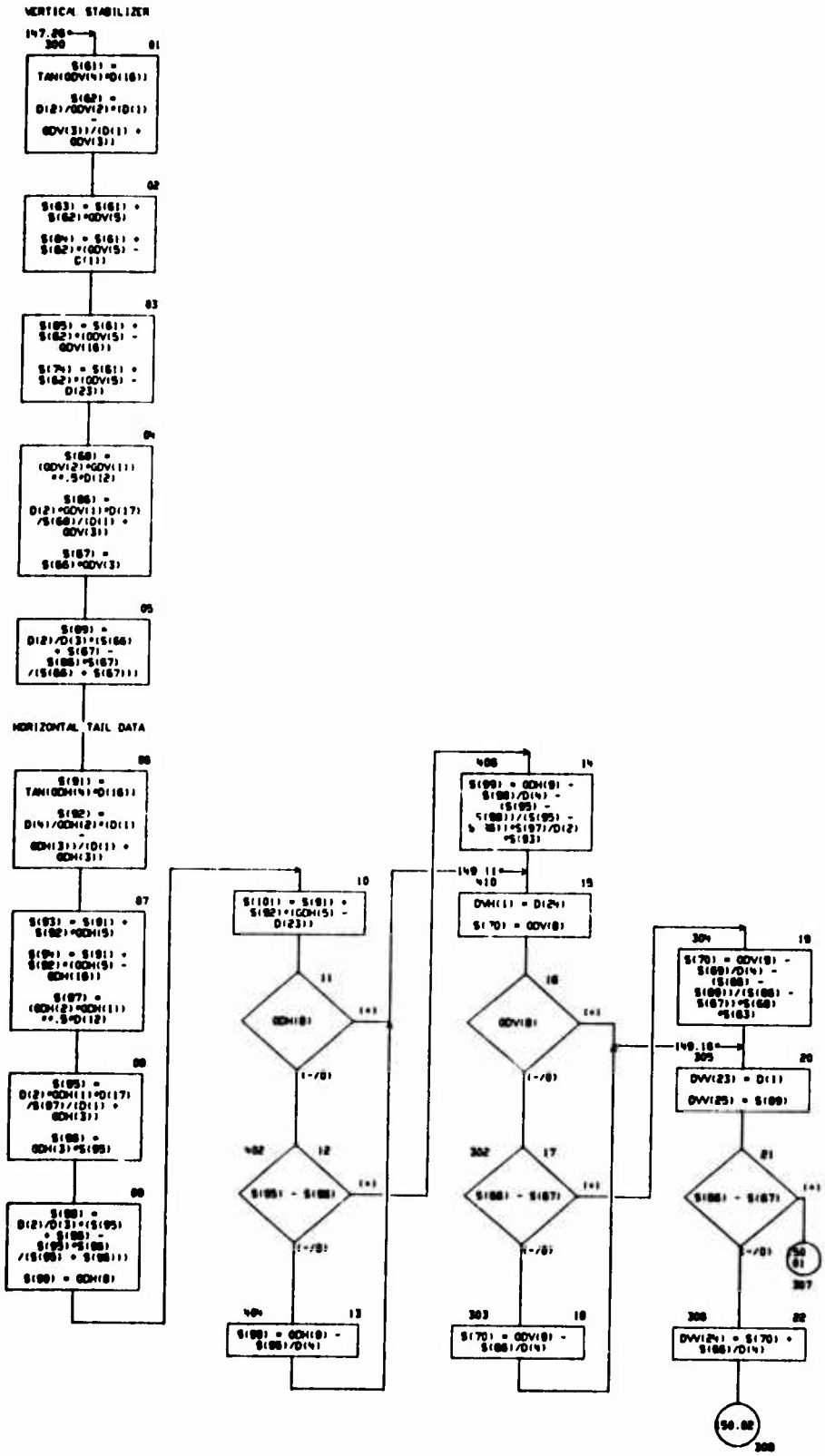


CHART TITLE - SUBROUTINE MAKEO

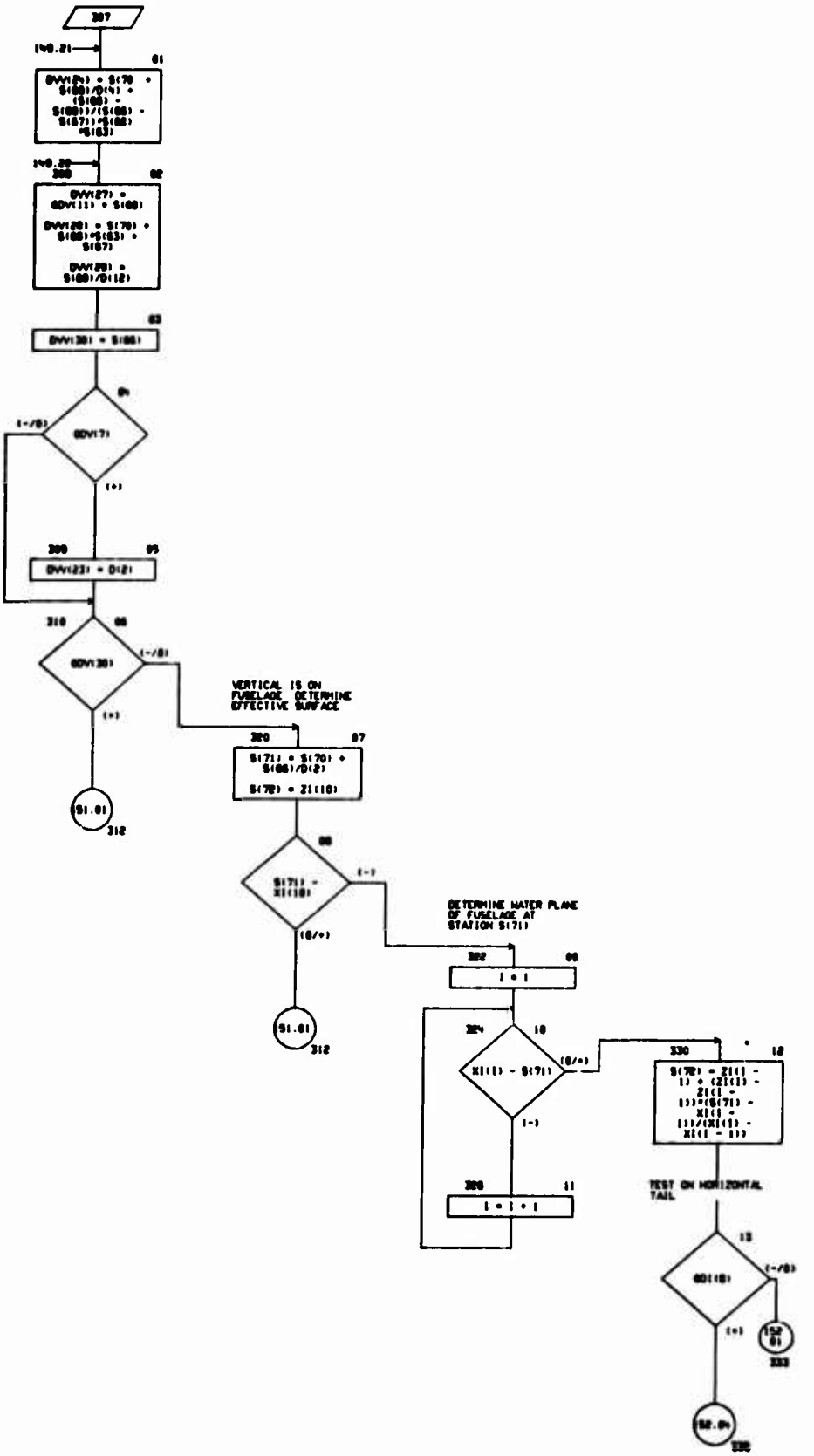


CHART TITLE - SUBROUTINE HANDEQ

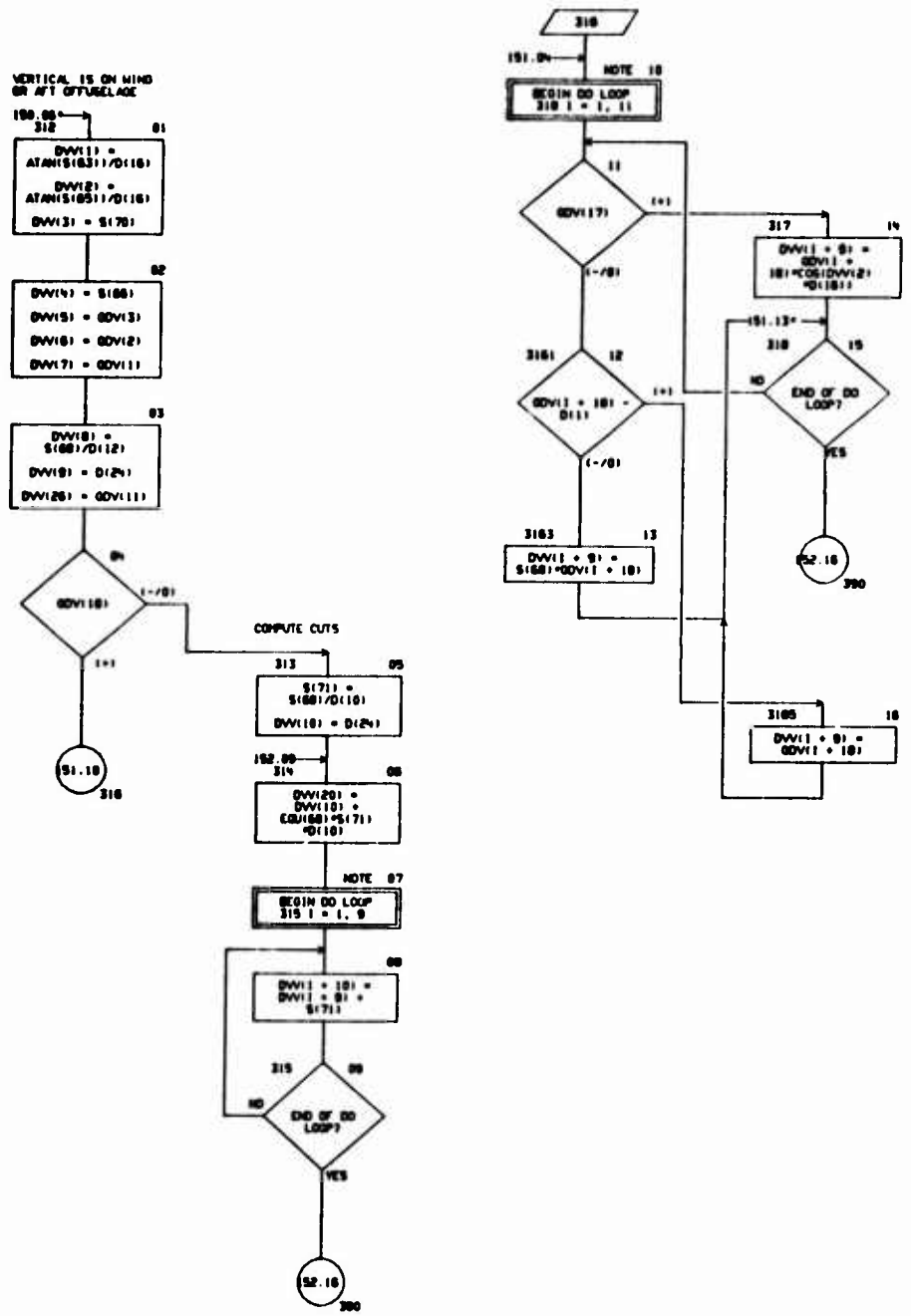


CHART TITLE - SUBROUTINE WINGED

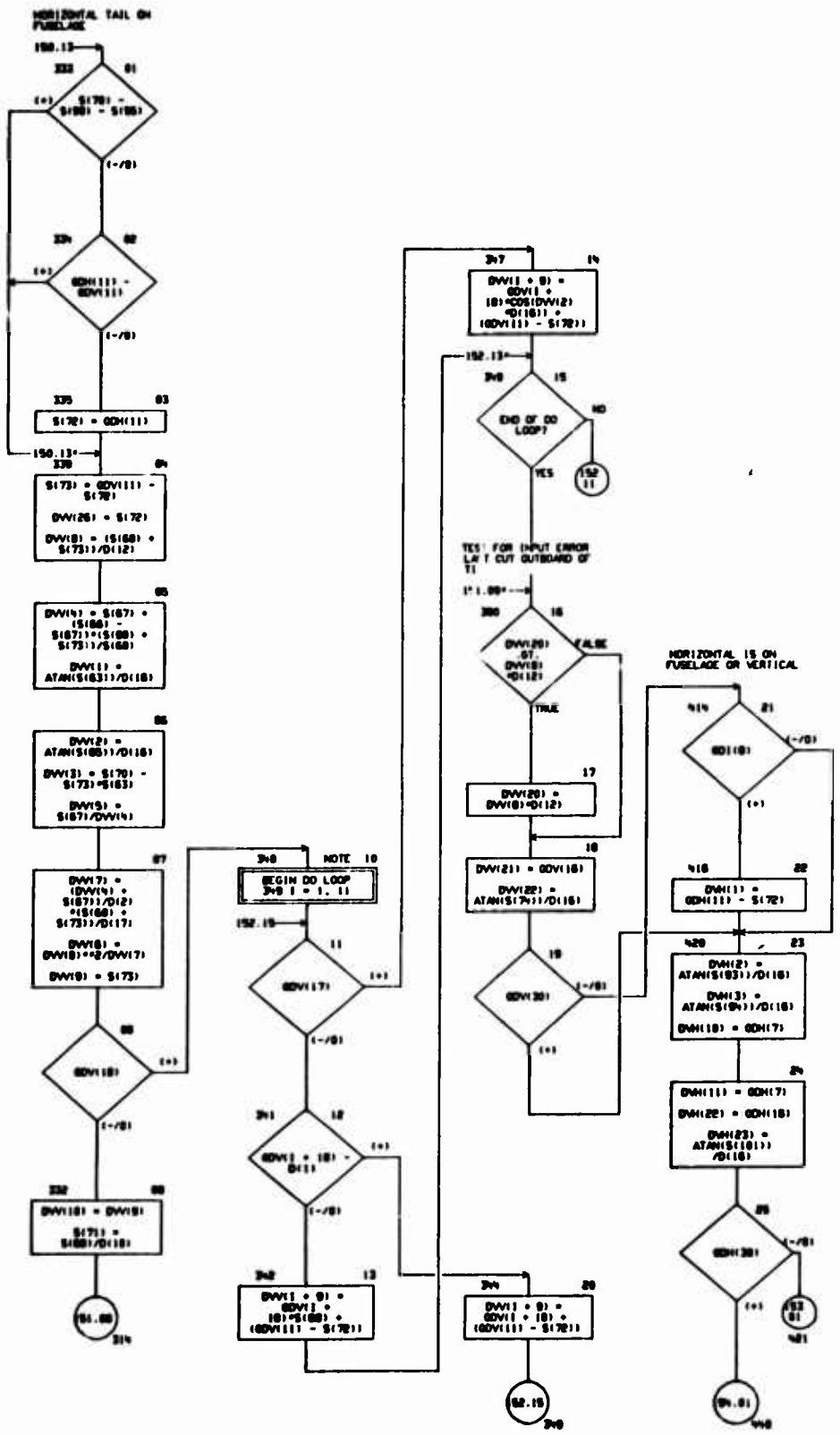


CHART TITLE - SUBROUTINE HAVEQ

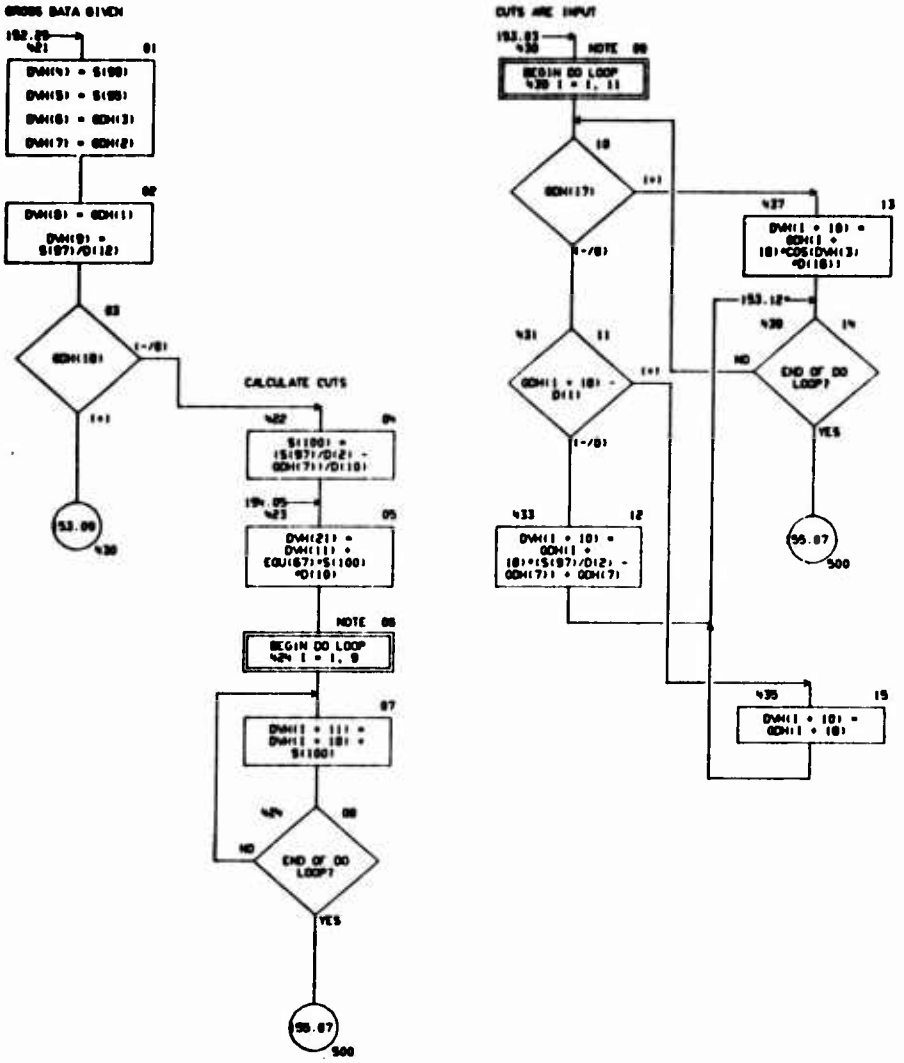


CHART TITLE - SUBROUTINE HAVGEO

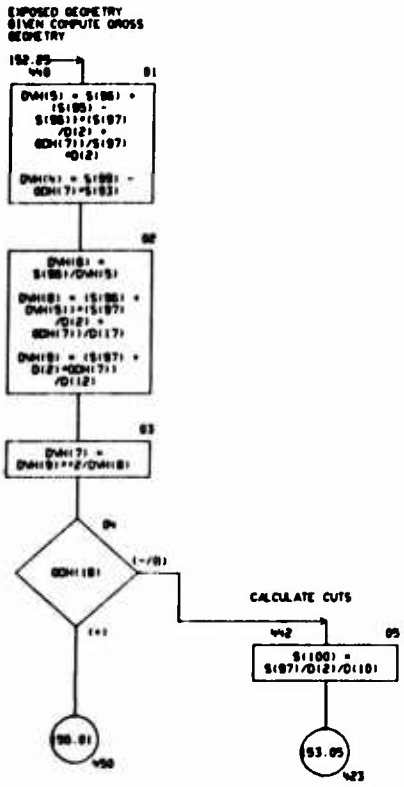


CHART TITLE - SUBROUTINE MMGEO

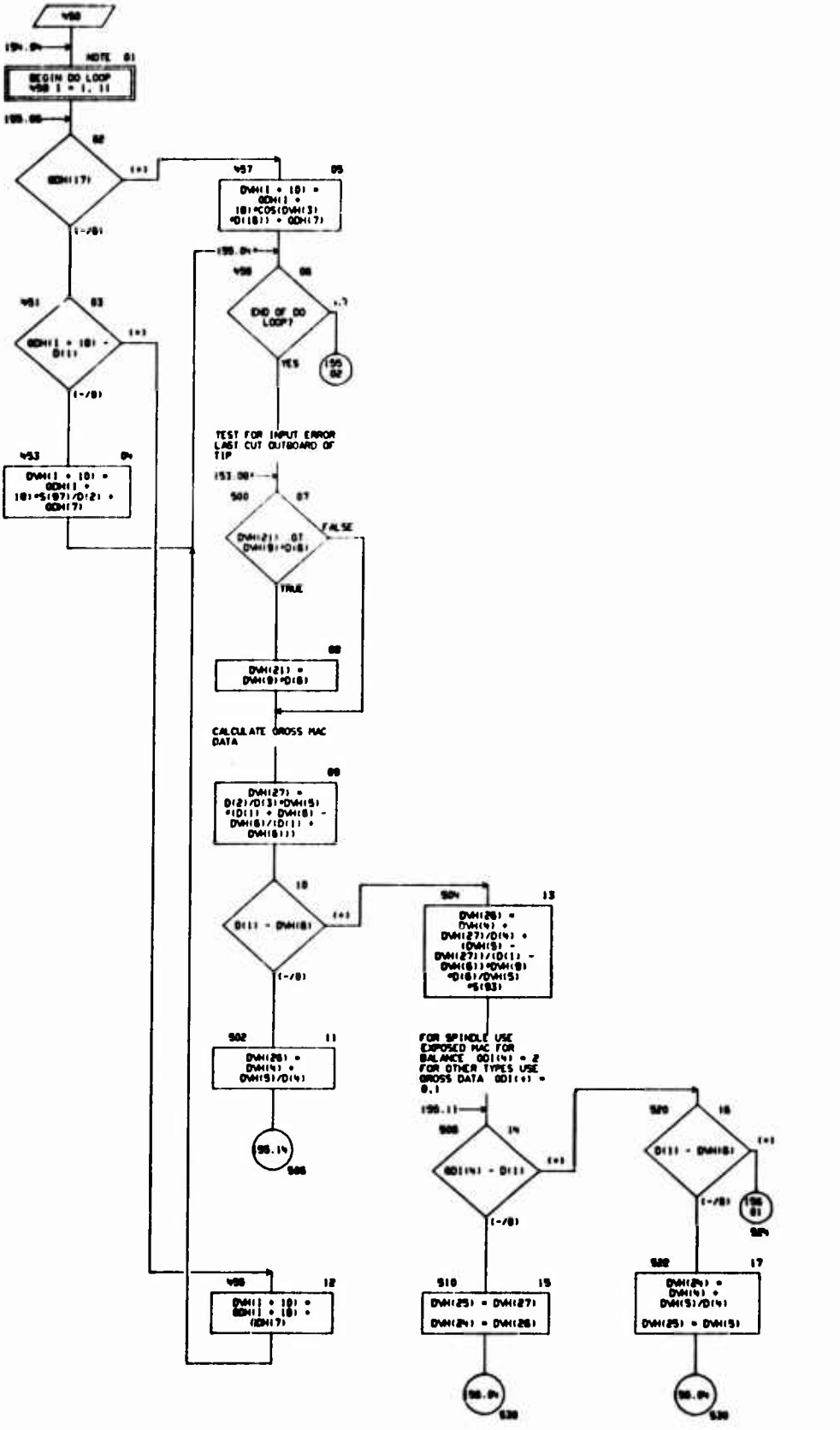


CHART TITLE - SUBROUTINE WINGED

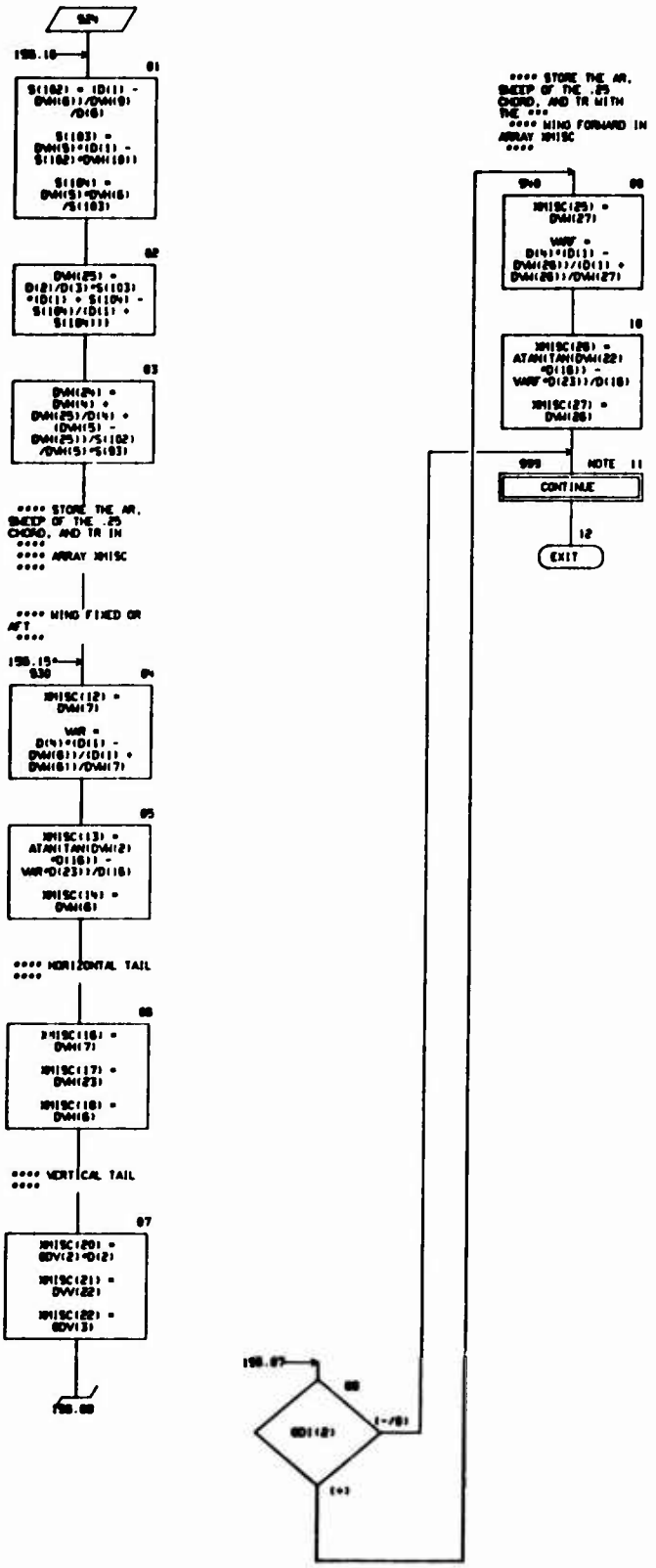


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
 COMMON TCOM(4320)
COMMON /MISC/ XMISC(100)
DIMENSION D(700),QD(700),DV(2320),S(400),ND(200)
DIMENSION EQU(200)
DIMENSION QD1(20),QDB(80),X1(10),Z1(10)
DIMENSION QDN(50),QDH(40),QDV(40),DVN(50),DVH(30),DVV(30)
DIMENSION DVAT(1000)
DIMENSION X(10),Y(10)
EQUIVALENCE (D(1),TCOM(1)),(QD(1),TCOM(701)),(DV(1),TCOM(1401)),
 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
EQUIVALENCE (EQU(1),D(1))
EQUIVALENCE (QD1(1),QD(1)),(QDB(1),QD(21))
 , (X1(1),QDB(6)),(Z1(1),QDB(16))
EQUIVALENCE (QDN(1),QD(25)),(QDH(1),QD(301)),(QDV(1),QD(39)),
 (DVN(1),DV(32)),(DVH(1),DV(371)),(DVV(1),DV(401))
EQUIVALENCE (DVAT(1),DV(1121))
EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(II,ND(107))
```

CHART TITLE - INTRODUCTORY COMMENTS

```
C*****/*************************************************************
C                    SUBROUTINE WNDDST
C*************************************************************
```

CHART TITLE - SUBROUTINE WNDDST

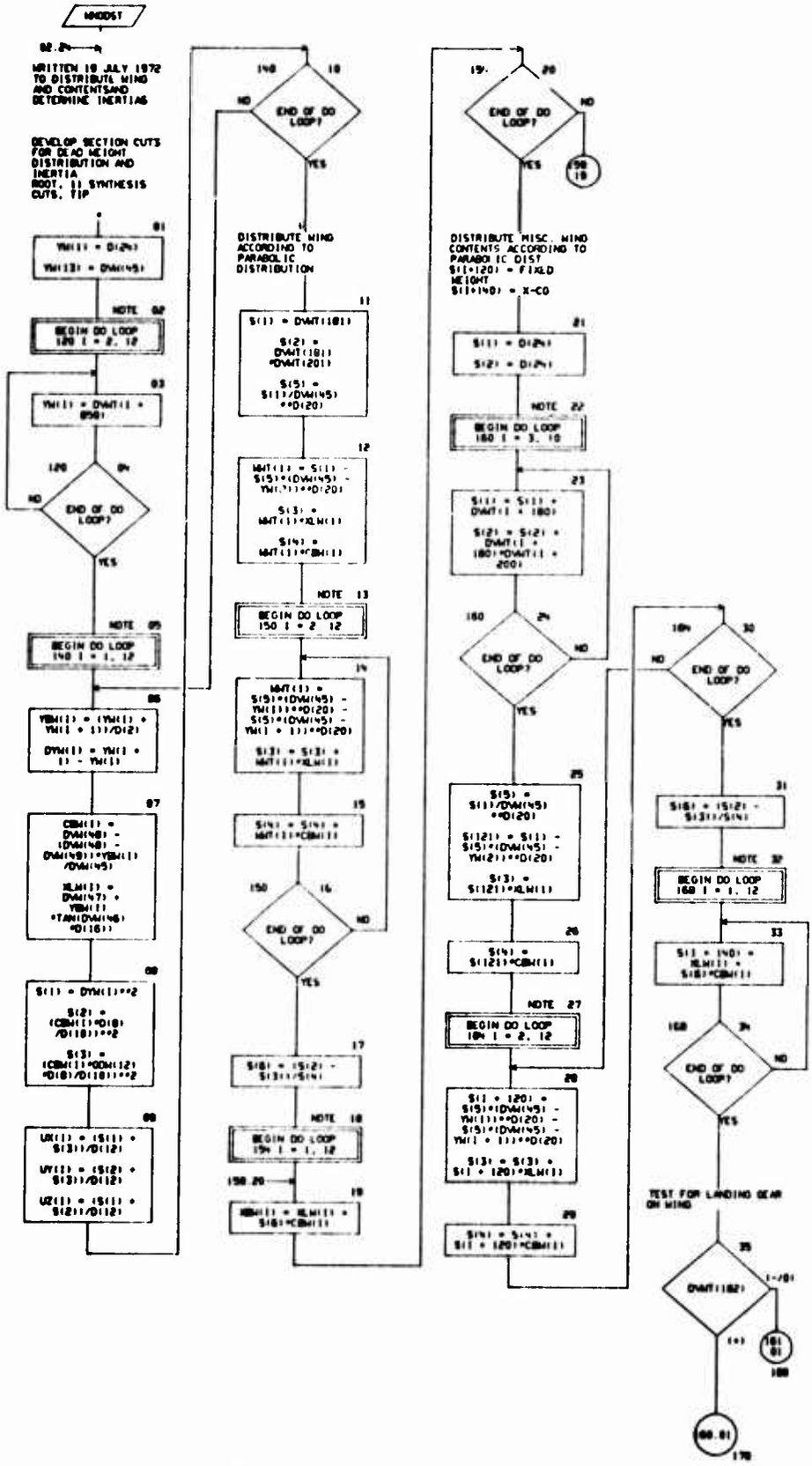


CHART TITLE - SUBROUTINE MNDBST

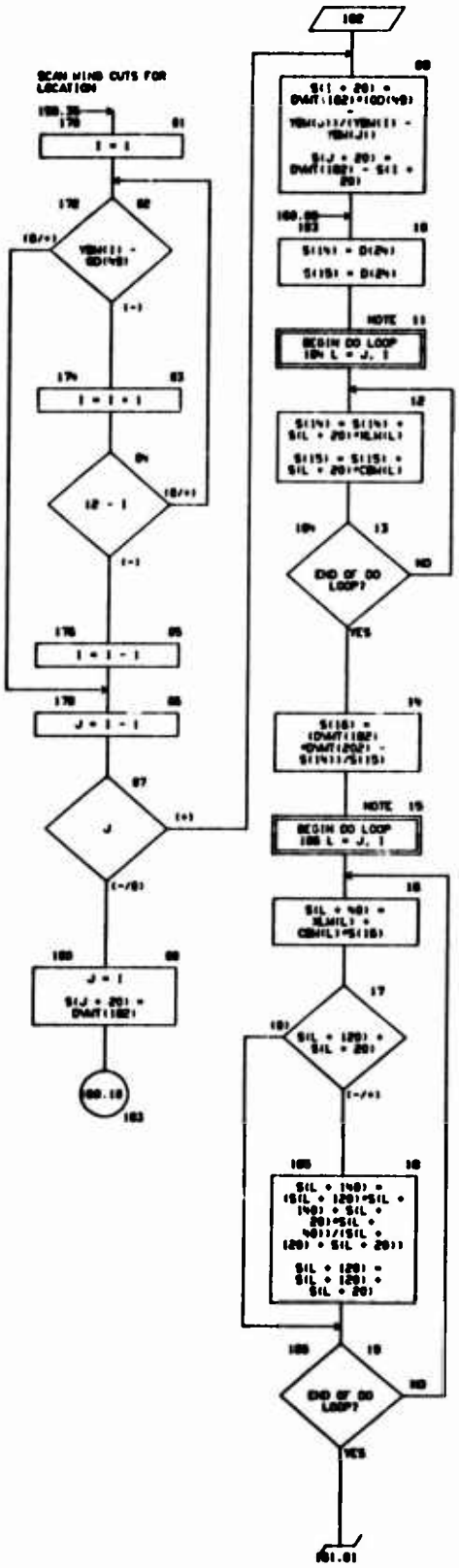


CHART TITLE - SUBROUTINE WNDDST

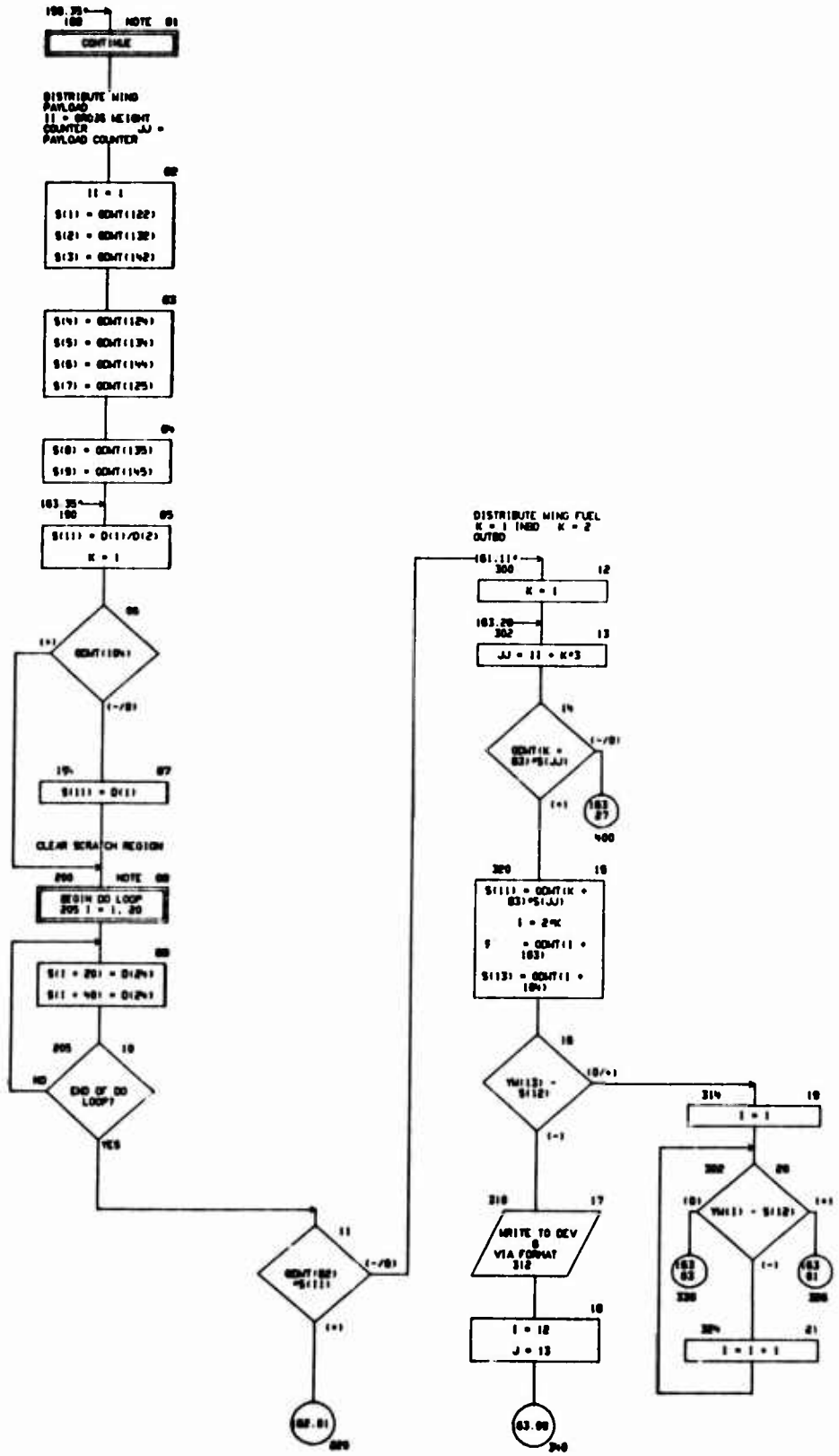


CHART TITLE - SUBROUTINE HADDST

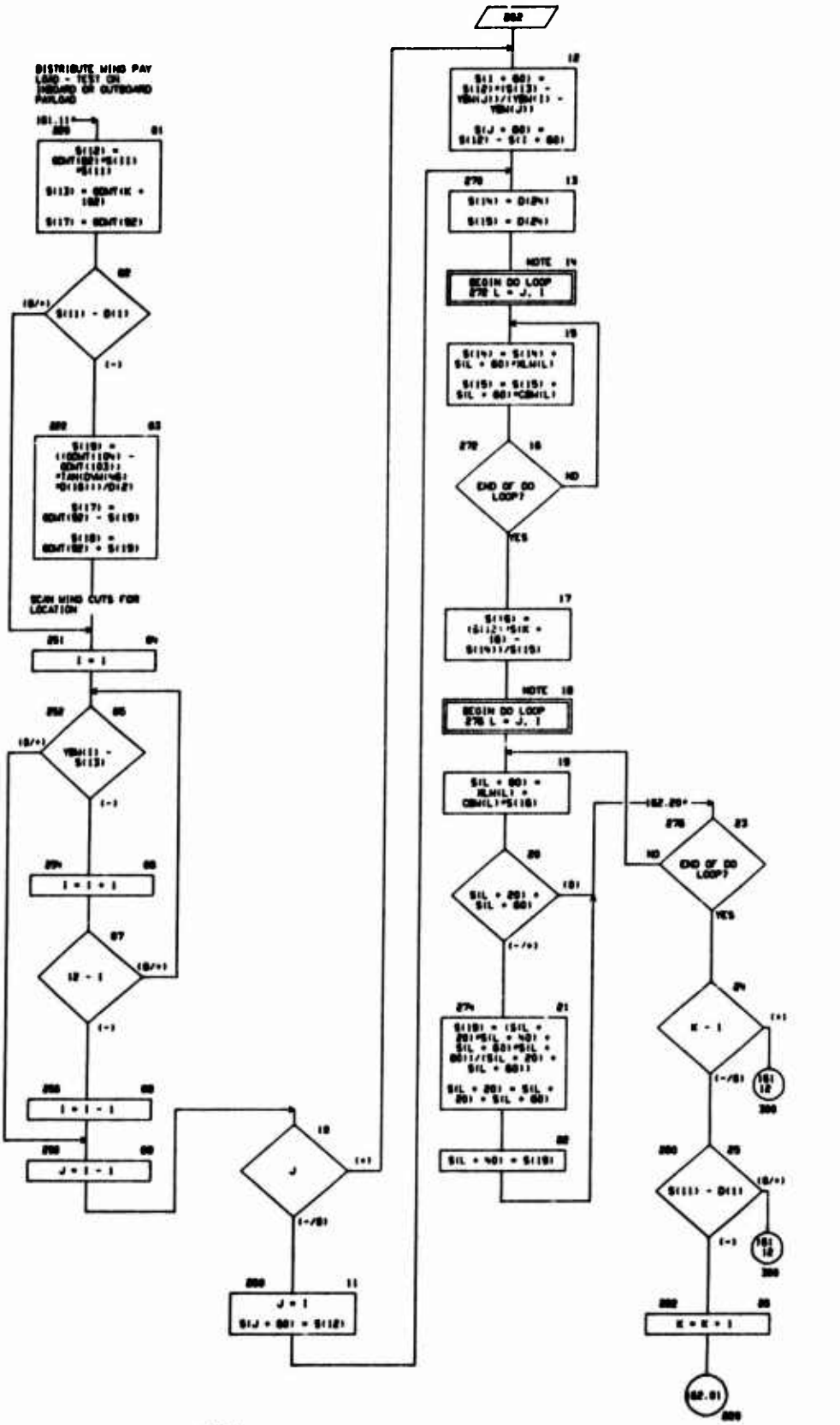


CHART TITLE - SUBROUTINE MMDIST

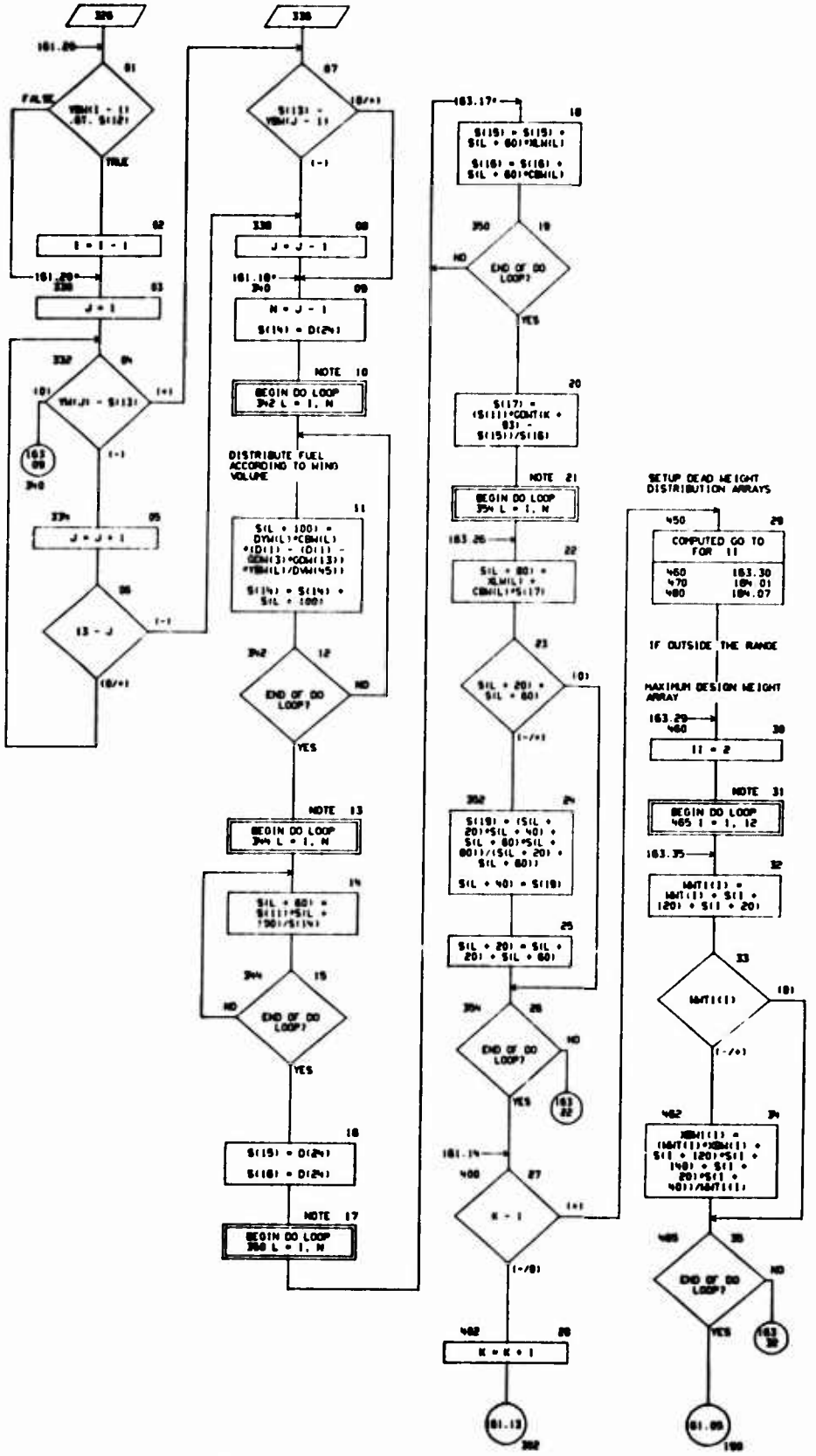


CHART TITLE - SUBROUTINE WINDSET

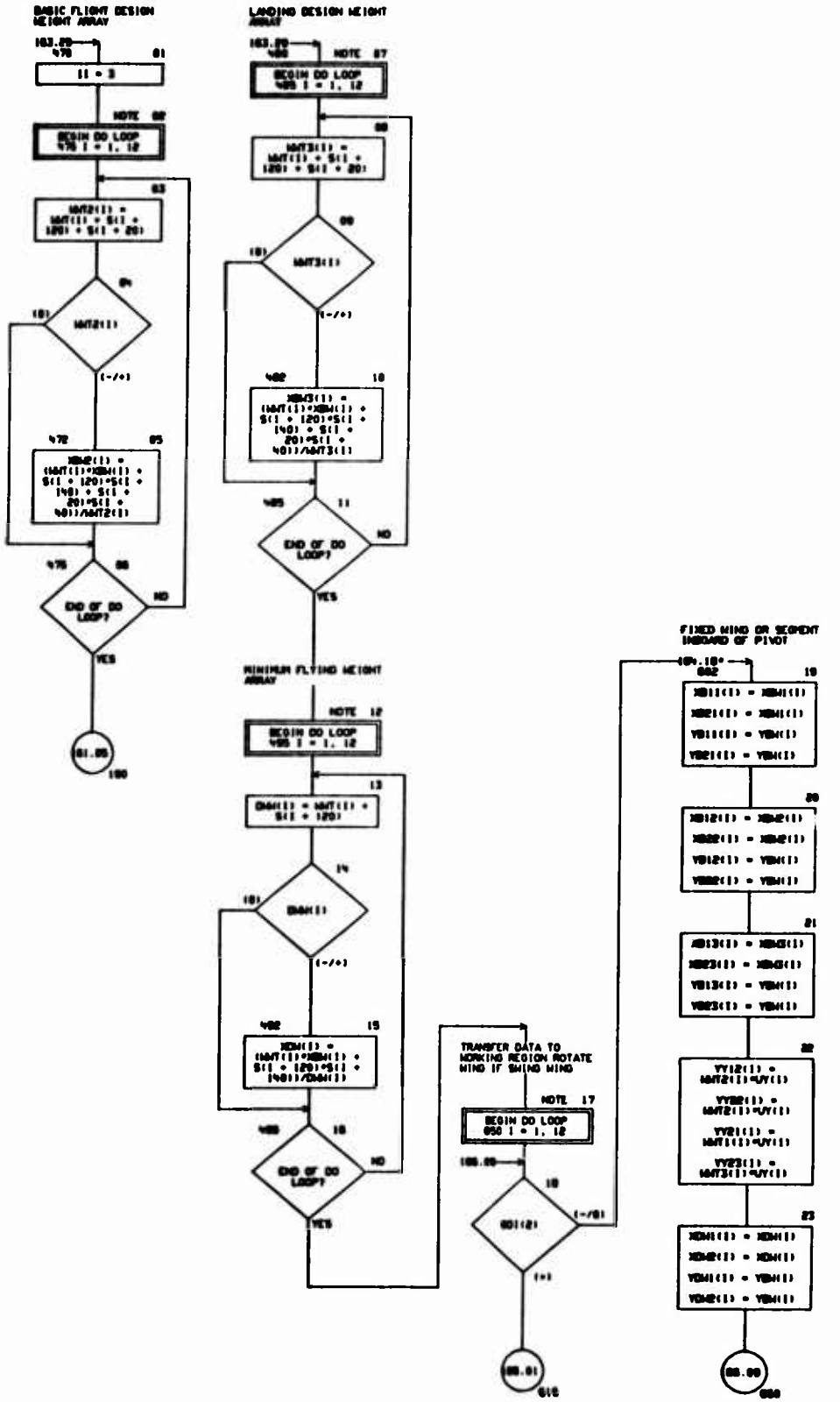


CHART TITLE - SUBROUTINE MMOST

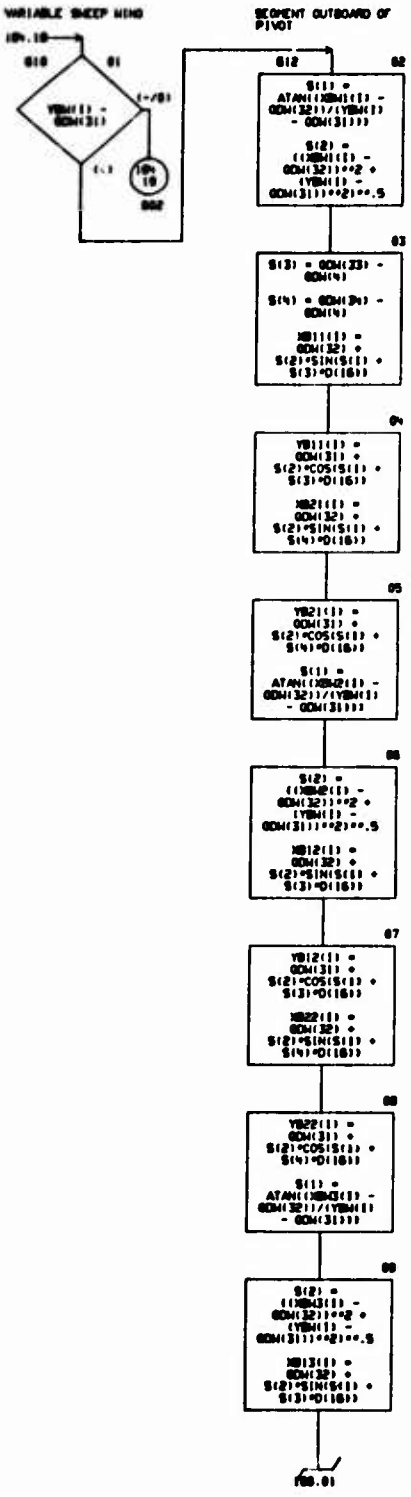


CHART TITLE - SUBROUTINE WINDST

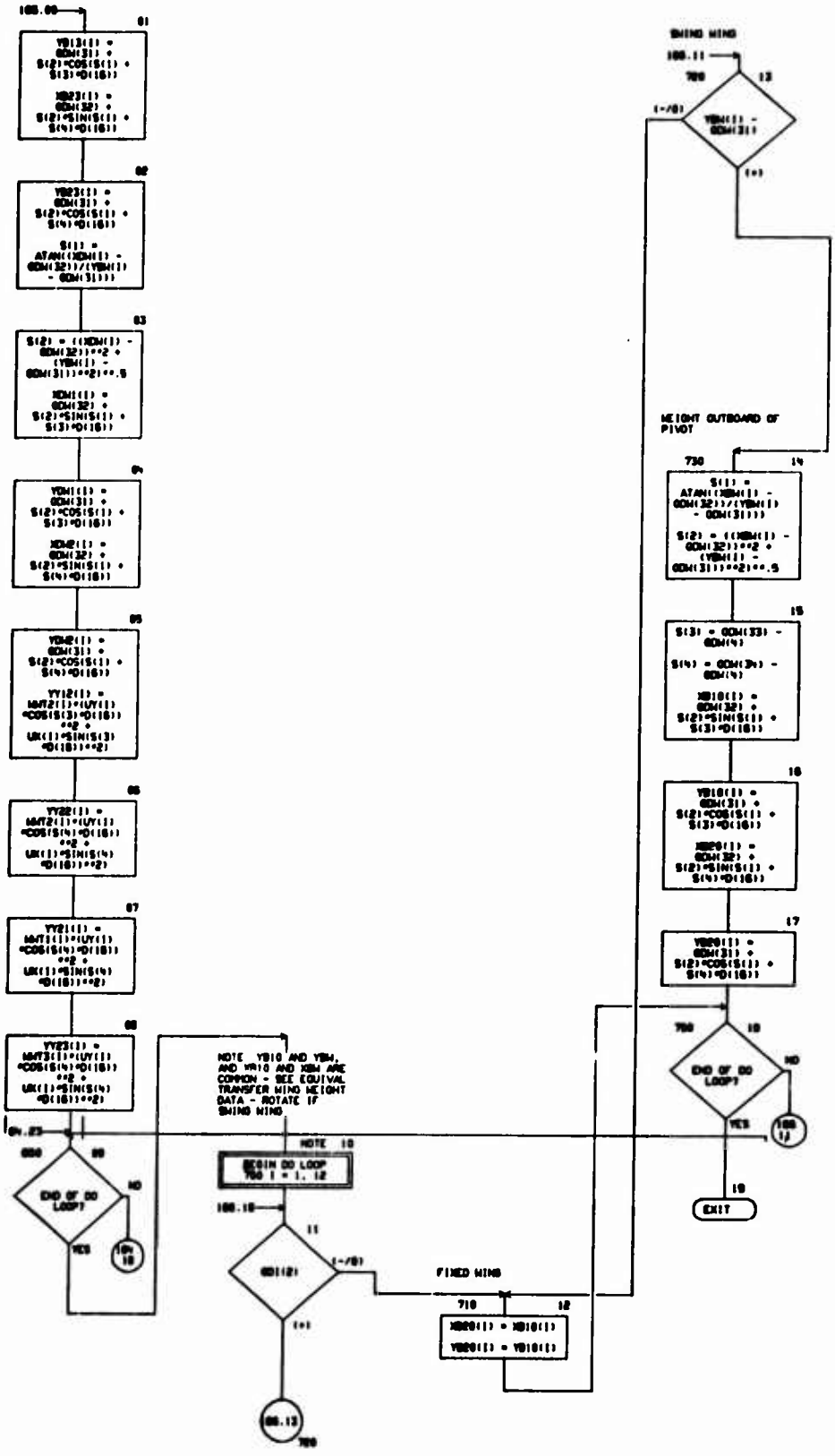


CHART TITLE - NON-PROCEDURAL STATEMENTS

```
      COMMON TCOM(4320)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION GD1(20),GDMT(160),GDM(50)
      DIMENSION DVM(50),DVMT(1000)
      DIMENSION YW(13),YBM(12),DYM(12),CBM(12),UX(12),UY(12),UZ(12)
      DIMENSION XLM(12)
      DIMENSION XB10(12),YB10(12),XB20(12),YB20(12)
      DIMENSION WMT(12),XBM(12),WMT1(12),XBM1(12),WMT2(12),XBM2(12),
     WMT3(12),XBM3(12),XB1(12),XB21(12),YB1(12),YB21(12),
     XB12(12),XB22(12),YB12(12),YB22(12),XB13(12),XB23(12),
     YB13(12),YB23(12),YY12(12),YY22(12),YY21(12),YY23(12)
      DIMENSION DXM(12),XDM(12),XDM1(12),XDM2(12),YDM1(12),YDM2(12)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (DVM(1),DV(321)),(DVMT(1),DV(1121))
     , (GD1(1),GD(1)), (GDMT(1),GD(91)), (GDM(1),GD(251))
      EQUIVALENCE (YW(1),DVMT(381)),(YBM(1),DVMT(401)),
     (DYM(1),DVMT(413)),(CBM(1),DVMT(425)),(UX(1),DVMT(437)),
     (UY(1),DVMT(449)),(UZ(1),DVMT(461))
      EQUIVALENCE (XLM(1),S(181))
      EQUIVALENCE (WMT(1),DVMT(473)),(XBM(1),DVMT(485)),
     (WMT1(1),DVMT(497)),(XBM1(1),DVMT(509)),(WMT2(1),DVMT(521)),
     (XBM2(1),DVMT(533)),(WMT3(1),DVMT(545)),(XBM3(1),DVMT(557)),
     (XB1(1),DVMT(569)),(XB21(1),DVMT(581)),(YB1(1),DVMT(593)),
     (YB21(1),DVMT(605)),(XB12(1),DVMT(617)),(XB22(1),DVMT(629)),
     (YB12(1),DVMT(641)),(YB22(1),DVMT(653)),(XB13(1),DVMT(665)),
     (XB23(1),DVMT(677)),(YB13(1),DVMT(689)),(YB23(1),DVMT(701)),
     (YY12(1),DVMT(713)),(YY22(1),DVMT(725)),(YY21(1),DVMT(737)),
     (YY23(1),DVMT(749))
      EQUIVALENCE (YB10(1),DVMT(401)),(XB20(1),DVMT(413)),
     (YB20(1),DVMT(425)),(XB10(1),DVMT(485))
      EQUIVALENCE (DXM(1),DVMT(872)),(XDM(1),DVMT(884))
     , (XDM1(1),DVMT(896)),(XDM2(1),DVMT(908))
     , (YDM1(1),DVMT(920)),(YDM2(1),DVMT(932))
      EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104)),
     (N,ND(105)),(II,ND(107)),(JJ,ND(108))
  312 FORMAT(1H0,20X,35HERROR INBD FUEL RIB IS OUTBD OF TIP)
```

```
FORTRAN MODULE      (LIST,AUTOSEQ)

 CARD NO        ****                         CONTENTS                         ****

          C
          C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
          C                          PROGRAM DATAIN
          C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
          C
                PROGRAM DATAIN
          C
          C       MAIN CONTROL DATA MANAGEMENT, DEAD WT DIST. AND INERTIA
          C       WRITTEN 25 MAY 1972
                 COMMON TCOM(4320)
                COMMON /IPRINT/ IP(80)
                COMMON /MISC/ XMISC(100)
          C
                DIMENSION D(700),DD(700),DV(2320),S(400),ND(200)
          C
                DIMENSION GDB(80),DATS(40),DATD(70),DATN(70)
          C
                DIMENSION BC(200)
          C
          C
                EQUIVALENCE (D(1),TCOM(1)),(DD(1),TCOM(701)),(DV(1),TCOM(1401)),
               1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
          C
                EQUIVALENCE (BC(1),DV(2121))
          C
                EQUIVALENCE (GDB(1),DD(381)),(DATS(1),DD(461)),
               1 (DATD(1),DD(501)) (DATN(1),DD(591))
          C
          C
                EQUIVALENCE (ITP,ND(111)),(IVG,ND(112))
                EQUIVALENCE (NC,ND(115)),(KC,ND(116)),(NCD,ND(117)),
               1 (KCD,ND(118)),(NCN,ND(119)),(KCN,ND(120))
          C
                DO 700 N=1,4320
            700 TCOM(N) = 0.0
          C
          C         READ GENERAL DATA IN RECORD 11
          C
                CALL READMS(1,D(1),1400,11)
          C
                ITP = DATS(1)
                IVG = DATS(3)
                NC = GDB(1)
                KC = GDB(2)
                NCD = DATD(1)
                KCD = DATD(2)
                CALL SPDALT
                CALL GRDWFR
                CALL FUSGEO
                CALL WINGEO
                CALL BUCGEO
                IF(ITP) 32,32,30
          C        NACELLE TYPE
             30 NCN = DATN(1)
                KCN = DATN(2)
                CALL NACGEO
             32 CONTINUE
          C
          C       ***********************************************************
          C       CALL NOSGEO TO SETUP FOREBODY GEOMETRY DATA IN BC ARRAY FOR
          C       USE IN AIRLOADS MODULE ROUTINE BLCNTL
          C       BC ARRAY = RECORD 22
          C
                CALL NOSGEO
          C
          C       ***********************************************************
          C
                CALL QUIKIE
                CALL WEIDST
          C       PRINT OPERATIONAL WEIGHT EMPTY AND USEFUL LOAD PRIOR TO
```

```
CARD NO ****                    CONTENTS                    ****
     71        C       FURTHER USEAGE
     72              IF(IP(40))5005,5005,5006
     73         5005 CONTINUE
     74              CALL PRTOBE
     75         5006 CONTINUE
     76              CALL WGBDST
     77              CALL FUSDST
     78              CALL CONDST
     79              CALL FTOTAL
     80        C
     81        C       ********************************************************
     82        C       CALL AVDATA TO CALCULATE TOTAL VEHICLE WEIGHT AND INERTIA
     83        C       SUMMARY
     84        C       AVDATA CALLS AVDINR TO SETUP FUSELAGE DEAD WEIGHT DISTRIBUTION
     85        C       DATA IN FUSDMI ARRAY FOR USE IN FUSELAGE MODULE ROUTINE FUSLD
     86        C       FUSDMI ARRAY = RECORD 34
     87        C
     88              CALL AVDATA
     89        C
     90        C       ********************************************************
     91        C
     92        C       ********************************************************
     93        C       CALL BBLCNT TO SETUP VEHICLE DATA IN BC ARRAY FOR USE IN
     94        C       AIRLOADS MODULE ROUTINE BLCNTL
     95        C       BC ARRAY = RECORD 22
     96        C
     97              CALL BBLCNT
     98        C
     99        C       ********************************************************
    100        C
    101        C       ********************************************************
    102        C       CALL BWAVGD TO SETUP SPEED-ALTITUDE AND H-TAIL INERTIA DATA
    103        C       IN SPAL ARRAY FOR USE IN FLUTTER AND TEMPERATURE MODULE
    104        C       ROUTINE WAVGD
    105        C       SPAL ARRAY = RECORD 38
    106        C
    107              CALL BWAVGD
    108        C
    109        C       ********************************************************
    110        C
    111        C       ********************************************************
    112        C       CALL BCONTL TO SETUP GEOMETRY AND INERTIA DATA IN WD ARRAY
    113        C       FOR USE IN WING AND EMPENNAGE MODULE ROUTINE CONTL
    114        C       WD ARRAY = RECORD 21
    115        C
    116              CALL BCONTL
    117        C
    118        C       ********************************************************
    119        C
    120        C       ********************************************************
    121        C       CALL BFATHS TO SETUP INERTIA LOADS DATA IN BC ARRAY FOR
    122        C       USE IN AIRLOADS MODULE FATIGUE EVALUATION ROUTINE FATHS
    123        C       BC ARRAY = RECORD 22
    124        C
    125              CALL BFATHS
    126        C
    127        C       ********************************************************
    128        C
    129        C       ********************************************************
    130        C       CALL BWAILD TO SETUP INERTIA DATA IN WLD ARRAY FOR USE IN
    131        C       AIRLOADS MODULE SURFACE DESIGN LOAD CALCULATION ROUTINE WAILDS
    132        C       WLD ARRAY = RECORD 18
    133        C
    134              CALL BWAILD
    135        C
    136        C       ********************************************************
    137        C
    138              IF(IP(40))5001,5001,5002
    139         5001 CONTINUE
    140        C       BREAKPOINT OUTPUT
    141              WRITE(6,84)
```

```
CARD NO  ****                          CONTENTS                          ****
   142        24  FORMAT(1H1,3HX,41H** TCON AT END OF DATA MANAGEMENT LINK **,
   143           1  13X,21H** DATAIN - IP(49) **)
   144              WRITE(6,88)
   145         88 FORMAT(1H0,32X,43H*** VEHICLE GEOMETRY AND MISC DATA FILE ***)
   146              DO 84 N=1,9120,5
   147              IF( TCON(N)+TCON(N+1)+TCON(N+2)+TCON(N+3)+TCON(N+4) .EQ. 0.0 )
   148             *      GO TO 84
   149              WRITE(6,82) N,TCON(N),TCON(N+1),TCON(N+2),TCON(N+3),TCON(N+4)
   150         82 FORMAT(4X,I4,5F16.4)
   151         84 CONTINUE
   152              DO 88 N=1,200,5
   153              K=9120+N
   154              IF(ND(N)+ND(N+1)+ND(N+2)+ND(N+3)+ND(N+4) .EQ. 0 ) GO TO 88
   155              WRITE(6,85) K, ND(N),ND(N+1),ND(N+2),ND(N+3),ND(N+4)
   156         85  FORMAT(4X,I4,5I10 )
   157         88  CONTINUE
   158        5002 CONTINUE
   159        C
   160        C       ***************************************************************
   161        C       WRITE DV ARRAY IN RECORD 19 FOR USE IN OUTPUT SUMMARY
   162        C       MODULE ROUTINE OUTPUT
   163        C
   164              CALL WRITMS(1,DV(1),2320,19)
   165        C
   166        C       ***************************************************************
   167        C
   168              IF(IP(47)) 5003,5003,5004
   169        5003 WRITE(6,400)(N,BC(N),BC(N+1),BC(N+2),BC(N+3),BC(N+4),N=1,200,5)
   170         400 FORMAT(1H1,2HX,45H*** BC ARRAY --- LOADS DATA - - RECORD 22 ***,
   171            1 19X,21H** DATAIN - IP(47) **//
   172            * (10X,I3,5F16.4) )
   173        5004 CONTINUE
   174        C
   175        C       ***************************************************************
   176        C       WRITE BC ARRAY IN RECORD 22 FOR USE IN AIRLOADS MODULE
   177        C
   178              CALL WRITMS(1,BC(1),195,22)
   179        C
   180        C       ***************************************************************
   181        C
   182        C
   183        C       ***************************************************************
   184        C       IMISC(24) = NDH IN LANDING GEAR INPUT DATA SET
   185        C       CALL DLNDGR TO SETUP DESIGN DATA FOR USE IN LANDING GEAR
   186        C       MODULE ROUTINE LANDGR
   187        C       D ARRAY = RECORD 25
   188        C
   189              IF(IMISC(24)) 70,70,80
   190        C
   191           70 CALL DLNDGR
   192        C
   193        C       ***************************************************************
   194        C
   195           80 CONTINUE
   196        C
   197              END
   198        C
   199        C ********************************************************************** ***********
   200        C                          SUBROUTINE AVDATA
   201        C ********************************************************************** )
   202        C
   203              SUBROUTINE AVDATA
   204        C       WRITTEN 24 JULY 1970
   205        C       TO DEVELOP TOTAL VEHICLE WEIGHT, CG, AND INERTIA DATA
   206        C
   207              COMMON TCON(9320)
   208              COMMON /IPRINT/ IP(80)
   209        C
   210              DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
   211        C
   212              DIMENSION GDH(50),GDH(40),GDV(40)
```

```
CARD NO  ****                          CONTENTS                          ****
  813          DIMENSION DATS(48)
  814      C
  815          DIMENSION DVN(30),DVV(30),DVB(4401,DVA(150),DVAT(1000)
  816          DIMENSION ZO(20),XBAR(20),UIY(20),UIZ(20)
  817          DIMENSION SFN(10),UIYN(10),UIZN(10),XBN(10)
  818          DIMENSION WFUS(20)
  819          DIMENSION WFC1(20),WFC2(20),WFC3(20)
  820      C
  821          EQUIVALENCE (D(1),TCON(1)),(GD(1),TCON(70))),(DV(1),TCON(148))),
  822         1 (S(1),TCON(378))),(ND(1),TCON(412)))
  823      C
  824          EQUIVALENCE (GDN(1),GD(25))),(GDN(1),GD(30))),(GDV(1),GD(34)))
  825      C
  826          EQUIVALENCE (DATS(1),GD(46))
  827      C
  828          EQUIVALENCE (DVN(1),DV(37))),(DVV(1),DV(40))),(DVB(1),DV(43))),
  829         1 (DVA(1),DV(97))),(DVAT(1),DV(112)))
  830      C
  831          EQUIVALENCE (ZO(1),DVB(1))),(XBAR(1),DVB(14))),
  832         1 (UIY(1),DVB(40))),(UIZ(1),DVB(42)))
  833      C
  834          EQUIVALENCE (SFN(1),DVA(7))),(UIYN(1),DVA(13))),
  835         1 (UIZN(1),DVA(16))),(XBN(1),DVA(19)))
  836      C
  837          EQUIVALENCE (WFUS(1),DVAT(34))
  838      C
  839          EQUIVALENCE (WFC1(1),DVAT(78))),(WFC2(1),DVAT(80))),
  840         1 (WFC3(1),DVAT(82)))
  841      C
  842          EQUIVALENCE (ITP,ND(111)),(NC,ND(115)),(NCN,ND(119))
  843      C
  844          DO 101 I=1,900
  845          S(I) = D(2)
  846      101 CONTINUE
  847      C ..........  ..........
  848      C       SUM WING AND CONTENTS
  849          CALL AVDWG
  850      C ..... ..... .... ..... . ..... .....
  851      C                   SUM FUSELAGE AND CONTENTS
  852          J = NC + 1
  853          DO 150 I=1,J
  854          S(32) = S(32) + WFUS(I) + WFC1(I)
  855          S(33) = S(33) + WFUS(I)*XBAR(I) +WFC1(I)*XBAR(I)
  856          S(34) = S(34) + WFUS(I) + WFC2(I)
  857          S(35) = S(35) + WFUS(I)*XBAR(I) +WFC2(I)*XBAR(I)
  858          S(36) = S(36) + WFUS(I) + WFC3(I)
  859          S(37) = S(37) + WFUS(I)*XBAR(I) +WFC3(I)*XBAR(I)
  860          S(83) = S(83) + WFUS(I)*ZO(I) + WFC1(I)*ZO(I)
  861          S(84) = S(84) + WFUS(I)*ZO(I) + WFC2(I)*ZO(I)
  862          S(85) = S(85) + WFUS(I)*ZO(I) + WFC3(I)*ZO(I)
  863      150 CONTINUE
  864          S(38) = S(33)/S(32)
  865          S(39) = S(35)/S(34)
  866          S(40) = S(37)/S(36)
  867          S(86) = S(83)/S(32)
  868          S(87) = S(84)/S(34)
  869          S(88) = S(85)/S(36)
  870          S(41) = DVAT(221) + DVAT(222)
  871          S(42) = DVAT(231)
  872      C       BASE INERTIA ON EXPOSED GEOMETRY
  873          S(101) = DVN(8)*D(8)
  874          S(102) = DVN(9) - (D(1) - DVN(8))*DVN(9)/S(101)*DVN(8)
  875          S(43) = DVN(10) + (S(101) - DVN(11))/D(3)
  876          S(44) = S(41)*S(102)**2/D(3)/D(8)
  877          S(45) = S(41)*(S(102)**2 + (S(101)-DVN(10))**2)/D(3)/D(8)
  878          S(46) = DVAT(24) + DVAT(242)
  879          S(47) = DVAT(251)
  880          S(101) = DVV(8)*D(12)
  881          S(102) = DVV(9) - (D(1) - DVV(8))*DVV(9)/S(101)*DVV(8)
  882          S(48) = DVV(10) + (S(101) - DVV(11))/D(3)
  883          S(49) = S(48)*(S(102)**2 + (S(101) - DVV(11))**2)/D(3)/D(8)
```

CARD NO	**** CONTENTS ****
204	S(58) = .5(46)*S(102)**2*(3)/D(8)
205	IF(ITP) 200,300,200
206	C NACELLE MOUNTED - IND ARD NACELLES
207	200 DO 210 I=1,20
208	S(51) = S(51) + DVAT(I+200)
209	S(52) = S(52) + DVAT(I+200)*DVAT(I+200)
210	210 CONTINUE
211	S(53) = S(52)/S(51)
212	S(54) = DATS(14)
213	C ASSUME WEIGHT TO BE UNIFORMLY DISTRIBUTED ACCORDING TO NACELLE
214	C WETTED AREA
215	J = NCN - 1
216	DO 220 I=1,J
217	S(63) = S(63) + SFN(I)
218	220 CONTINUE
219	DO 230 I=1,J
220	S(103) = DATS(13) + XBN(I) - S(53)
221	S(55) = S(55) + S(51)*SFN(I)/S(63)*(UIYN(I) + S(103)**2)
222	S(56) = S(56) + S(51)*SFN(I)/S(63)*(UIZN(I) + S(103)**2)
223	230 CONTINUE
224	IF(2 - ITP) 240,300,300
225	240 DO 250 I=1,20
226	S(57) = S(57) + DVAT(I+300)
227	S(58) = S(58) + DVAT(I+300)*DVAT(I+320)
228	250 CONTINUE
229	S(59) = S(58)/S(57)
230	S(60) = DATS(17)
231	DO 260 I=1,J
232	S(103) = DATS(16) + XBN(I) - S(59)
233	S(61) = S(61) + S(57)*SFN(I)/S(63)*(UIYN(I) + S(103)**2)
234	S(62) = S(62) + S(57)*SFN(I)/S(63)*(UIZN(I) + S(103)**2)
235	260 CONTINUE
236	C ASSEMBLE TOTAL VEHICLE
237	300 S(64) = S(1) + S(32) + S(41) + S(46) + S(51) + S(57)
238	S(65) = S(2) + S(33) + S(41)*S(42) + S(46)*S(47) + S(52) + S(58)
239	S(66) = S(65) - S(2) + S(3)
240	S(67) = S(65)/S(64)
241	S(68) = S(66)/S(64)
242	S(69) = S(6) + S(34) + S(41) + S(46) + S(51) + S(57)
243	S(70) = S(7) + S(35) + S(41)*S(42) + S(46)*S(47) + S(52) + S(58)
244	S(71) = S(70) - S(7) + S(8)
245	S(72) = S(70)/S(69)
246	S(73) = S(71)/S(69)
247	C LANDING DESIGN WEIGHT WINGS FORWARD OR FIXED
248	S(80) = S(1) + S(35) + S(41) + S(46) + S(51) + S(57)
249	S(81) = S(13) + S(37) + S(41)*S(42) + S(46)*S(47) + S(52) + S(58)
250	S(82) = S(81)/S(80)
251	S(97) = S(1)*XBAR(1) + S(83) + S(41)*XBAR(1) + S(46)*S(48) +
252	1 S(51)*DATS(15) + S(57)*DATS(18)
253	S(98) = S(97)/S(64)
254	S(99) = S(8)*XBAR(1) + S(84) + S(41)*XBAR(1) + S(46)*S(48) +
255	1 S(51)*DATS(15) + S(57)*DATS(18)
256	S(100) = S(99)/S(80)
257	S(101) = S(1)*XBAR(1) + S(85) + S(41)*XBAR(1) + S(46)*S(48) +
258	1 S(51)*DATS(15) + S(57)*DATS(18)
259	S(102) = S(101)/S(80)
260	C DEVELOP VEHICLE INERTIA AT FDOM START WITH FUSELAGE
261	J = NC + 1
262	DO 310 I=1,J
263	S(74) = S(74) + (WFUS(I) + WFC2(I))*(UIY(I) +
264	1 (S(72) - XBAR(I))**2 + (Z0(I) - S(100))**2)
265	S(75) = S(75) + (WFUS(I) + WFC2(I))*(UIY(I) +
266	1 (S(73) - XBAR(I))**2 + (Z0(I) - S(100))**2)
267	S(76) = S(76) + (WFUS(I) + WFC2(I))*(UIZ(I) +
268	1 (S(72) - XBAR(I))**2)
269	S(77) = S(77) + (WFUS(I) + WFC2(I))*(UIZ(I) +
270	1 (S(73) - XBAR(I))**2)
271	S(103) = S(83) + (WFUS(I) + WFC1(I))*(UIY(I) +
272	1 (S(80) - XBAR(I))**2 + (Z0(I) - S(80))**2)
273	S(84) = S(84) + (WFUS(I) + WFC3(I))*(UIY(I) +
274	1 (S(82) - XBAR(I))**2 + (Z0(I) - S(100))**2)

```
CARD NO  ****                          CONTENTS                          ****
     $(95) = $(95) + (WEVS(I) + WFC(I,1))*(UIZ(I) +
    1 ($(88) - XBAR(I))**2)
     $(96) = $(96) + (WEVS(I) + WFC3(I))*(UIZ(I) +
    1 ($(82) - XBAR(I))**2)
  310 CONTINUE
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C       ADD OTHER COMPONENTS
      CALL AVDADC
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      CALL AVDINR
      DVAT(944) = $(84)
      DVAT(945) = $(87)
      DVAT(946) = $(88)
      DVAT(947) = $(89)
      DVAT(948) = $(93)
      DVAT(950) = $(95)
      DVAT(951) = $(88)
      DVAT(952) = $(72)
      DVAT(953) = $(73)
      DVAT(954) = $(100)
      DVAT(957) = $(74)
      DVAT(958) = $(75)
      DVAT(959) = $(76)
      DVAT(960) = $(77)
      DVAT(961) = $(80)
      DVAT(963) = $(82)
      DVAT(964) = $(102)
      DVAT(966) = $(94)
      DVAT(967) = $(96)
      DVAT(968) = $(41)
      DVAT(969) = $(42)
      DVAT(970) = $(43)
      DVAT(972) = $(44)
      DVAT(973) = $(45) + $(4,)*$(43)**2
      DVAT(974) = $(51)
      DVAT(975) = $(53)
      DVAT(977) = $(55)
      DVAT(978) = $(56)
      DVAT(979) = $(57)
      DVAT(980) = $(58)
      DVAT(982) = $(61)
      DVAT(983) = $(62)
C
      IF(IP(40))5001,5001,5002
 5001 CONTINUE
      WRITE (6,400)(I,$(I),$(I+1),$(I+2),$(I+3),$(I+4),I=1,105,5)
  400 FORMAT(1H1 $(I) FROM AVDATA,7I3,2IN** AVDATA - IP(40) **/
     1   (110, 5F10.4))
 5002 CONTINUE
C
      RETURN
      END
C
C ******************************************************************************
C                          SUBROUTINE AVDADC
C ******************************************************************************
C
      SUBROUTINE AVDADC
C
C ADD OTHER COMPONENTS
C     SEPARATED FROM AVDATA BECAUSE OF COMPILER TROUBLES
C
      COMMON TCOM(4320)
C
      DIMENSION D(700),GD(700),DV(2320),$(400),ND(200)
      DIMENSION  GDM(50),GDM(40),GDV(40)
      DIMENSION  DAT$(40)
C
      EQUIVALENCE(D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 ($(1),TCOM(3721)),(ND(1),TCOM(4121))
C HOLDING TEMPORARY FOR  SQUARES OF X DISTANCES TO NEW CG.
```

CARD NO **** CONTENTS ****

```
      EQUIVALENCE (GDH(1),GD(25)),(GDH(1),GD(30)),(GDV(1),GD(34))
      EQUIVALENCE (DATS(1),GD(48))
      EQUIVALENCE (XAH,S(110)), (XAH,S(111)), (XAV,S(112)),
     1 (XANI,S(113)), (XAND,S(114))
      EQUIVALENCE                (YAH,S(115)), (YAV,S(116)),
     1 (YANI,S(117)), (YAND,S(118))
      EQUIVALENCE (ZAH,S(119)), (ZAH,S(120)), (ZAV,S(121)),
     1 (ZANI,S(122)), (ZAND,S(123))
      EQUIVALENCE (X,S(124)), (Z,S(125)), (M,S(126))
C
C  10Y SUM OF HORIZONTAL, VERTICAL, NAC.1B., NAC 0B
      S(78) = S(44) + S(48) + S(55) + S(61)
C  10Z SUM OF M,V,NI,ND
      S(79) = S(45) + S(50) + S(56) + S(62)
      YAH = S(43)**2
      YAV = GDV(7)**2
      YANI= S(54)**2
      YAND= S(60)**2
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C       BFDH WITH WINDS AFT
      N=1
      X = S(72)
      Z = S(100)
      M = S(24)
   10 XAH = (X-M)**2
      XANI = (X-S(42))**2
      XAV = (X-S(47))**2
      XANI= (X-S(53))**2
      XAND= (X-S(59))**2
      ZAH = (Z-GDH(1))**2
      ZAH = (Z-GDH(1))**2
      ZAV = (Z-S(48))**2
      ZANI= (Z-DATS(15))**2
      ZAND= (Z-DATS(18))**2
      GO TO (20,30,40,50),N
C 1YY
   20 S(74) = S(74) + S(78) + S(18) + S(6)*(XAH+ZAH) + S(41)*(XAH+ZAH)
     1       + S(46)*(XAV+ZAV) + S(51)*(XANI+ZANI) + S(57)*(XAND+ZAND)
C 1ZZ
      S(75) = S(75) + S(79) + S(18) + S(6)*XAH
     1        + S(41)*(XAH + YAH) + S(46)*(XAV + YAV)
     2        + S(51)*(XANI + YANI) + S(57)*(XAND + YAND)
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C       BFDH WITH WINDS FORWARD
      N=2
      X = S(73)
      M = S(25)
      GO TO 10
C 2YY
   30 S(75) = S(75) + S(79) + S(18) + S(17) + S(6)*(XAH+ZAV) + S(41)*(XAH+ZAH)
     1       + S(46)*(XAV+ZAV) + S(51)*(XANI+ZANI) + S(57)*(XAND+ZAND)
C 2ZZ
      S(77) = S(77) + S(79) + S(18) + S(6)*XAH
     1        + S(41)*(XAH + YAH) + S(46)*(XAV + YAV)
     2        + S(51)*(XANI + YANI) + S(57)*(XAND + YAND)
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C       MAXIMUM DESIGN WEIGHT WITH WINDS FORWARD
C  TAKE OFF GROSS WEIGHT WITH WING FWD
      N=3
      X = S(69)
      Z = S(69)
      M = S(21)
      GO TO 10
C 3YY
   40 S(69) = S(69) + S(79) + S(69) + S(1)*(XAH+ZAH) + S(41)*(XAH+ZAH)
     1       + S(46)*(XAV+ZAV) + S(51)*(XANI+ZANI) + S(57)*(XAND+ZAND)
C 3ZZ
      S(69) = S(69) + S(79) + S(9) + S(1)*XAH
     1        + S(41)*(XAH + YAH) + S(46)*(XAV + YAV)
     2        + S(51)*(XANI + YANI) + S(57)*(XAND + YAND)
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
C       LANDING DESIGN WEIGHT WITH WINGS FORWARD
      N=4
      X = S(82)
      Z = S(182)
      W = S(20)
      GO TO 10
C IYY
   50 S(94) = S(94) + S(78) + S(80) + S(11)*(XAW+ZAW) + S(41)*(XAW+ZAW)
     1      + S(46)*(XAV+ZAV) + S(51)*(XAN1+ZAN1) + S(57)*(XAND+ZAND)
C IZZ
      S(95) = S(95) + S(78) + S(82) + S(11)*XAW
     1      + S(41)*(XAW + YAW) + S(46)*(XAV + YAV)
     2      + S(51)*(XAN1 + YAN1) + S(57)*(XAND + YAND)
  999 RETURN
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                           SUBROUTINE AVDINR
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE AVDINR
C
C       THIS ROUTINE SETS UP FUSELAGE WEIGHT DISTRIBUTION AND SPEED-
C       ALTITUDE PROFILE DATA FOR USE IN FUSELAGE MODULE ROUTINE FUSLD
C
       COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
C
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
C
      DIMENSION FUSDWI(400)
C
      DIMENSION GDI(20)
      DIMENSION GDW(50),GDH(40),GDV(40)
      DIMENSION DATS(40)
      DIMENSION DWT(1000)
      DIMENSION ALT(10),VL(10),QL(10)
      DIMENSION WFUS(20),WFC1(20),WFC2(20),WFC3(20)
C
      DIMENSION RT(35),RW(35),RH(35),RV(35),RA(35),RO(35)
C
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
C
      EQUIVALENCE (GDI(1),GD(1))
      EQUIVALENCE (GDW(1),GD(251)),(GDH(1),GD(301)),(GDV(1),GD(341))
      EQUIVALENCE (DATS(1),GD(461))
C
      EQUIVALENCE (DWT(1),DV(1121))
      EQUIVALENCE (ALT(1),DV(1)),(VL(1),DV(71)),(QL(1),DV(91))
      EQUIVALENCE (WFUS(1),DWT(341)),(WFC1(1),DWT(781))
     * , (WFC2(1),DWT(801)),(WFC3(1),DWT(821))
C
      EQUIVALENCE (RT(1),S(181)), (RW(1),S(226)),(RH(1),S(261)),
     1 (RV(1),S(296)), (RA(1),S(331)), (RO(1),S(366))
*
C WORKING STORAGE FOR COMBINING CALCS   WT, XCG, YCG,ZCG
      EQUIVALENCE ( WT,S(110)),( XCG,S(111)),( YCG,S(112)),(ZCG,S(113))
      EQUIVALENCE (VYCG,S(115)), (AZCG,S(114))
C
C .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
C SET UP INERTIA AND WEIGHT DISTRIBUTION DATA RECORD
C       STARTING FROM DOW THRU STOZ ----- 6 5*7 BLOCKS
C       IN S(181) THRU S(400)
C         SAVE INBOARD NACELLE DATA
      RT(1) = S(80)
      RT(2) = S(80)
      RT(3) = S(84)
      RT(5) = S(84)
      RT(4) = S(80)
      RT(6) = S(72)
```

```
CARD NO     ****                          CONTENTS                          ****
                 RT(7) = S(73)
                 RT(8) = S(88)
                 RT(10)= S(88)
                 RT(9) = S(82)
           C TOTAL HAS ZERO YCG(11 THRU 15), AND TIXX(21 THRU 25)
                 RT(16)= S(100)
                 RT(17)= S(100)
                 RT(18)= S(98)
                 RT(20)= S(98)
                 RT(19)= S(102)
                 RT(26)= S(74)
                 RT(27)= S(75)
                 RT(28)= S(93)
                 RT(30)= S(93)
                 RT(29)= S(94)
                 RT(31)= S(76)
                 RT(32)= S(77)
                 RT(33)= S(95)
                 RT(35)= S(96)
                 RT(34)= S(96)
           C ......................................................................
           C BASIC WING STORED IN RW FIRST, COMPONANTS ADDED IF NECESSARY
                 RW(1) = S(6)
                 RW(2) = S(6)
                 RW(3) = S(1)
                 RW(5) = S(1)
                 RW(4) = S(11)
                 RW(6) = S(24)
                 RW(7) = S(25)
                 RW(8) = S(21)
                 RW(10) = S(21)
                 RW(9) = S(20)
                 RW(11) = S(26)
                 RW(12) = S(27)
                 RW(13) = S(23)
                 RW(15) = S(23)
                 RW(14) = S(31)
                 RW(16) = GDW(11)
                 RW(17) = GDW(11)
                 RW(18) = GDW(11)
                 RW(19) = GDW(11)
                 RW(20) = GDW(11)
                 RW(26) = S(16)
                 RW(27) = S(17)
                 RW(31) = S(18)
                 RW(32) = S(19)
                 RW(28) = S(88)
                 RW(30) = S(88)
                 RW(33) = S(91)
                 RW(35) = S(91)
                 RW(29) = S(90)
                 RW(34) = S(92)
           C ......................................................................
                 VYCG = GDV(7)
           C HORIZONTAL  CONSTANT FOR ALL 5 CONDITIONS
             38  DO 40 N=1,5
                 RH(N) = S(41)
                 RH(N+5) = S(42)
                 RH(N+10) = S(43)
                 RH(N+15) = GDH(11)
                 RH(N+25) = S(44)
                 RH(N+30) = S(45) + S(41)*S(43)**2
             40  CONTINUE
           C .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-..-
           C  VERTICAL - IF ON FUS. PUT IN RV, IF ON WING MUST ADD IN
                 IF(GDV(30)) 60,60,80
             60  DO 65 N=1,5
                 RV(N) = S(46)
                 RV(N+5) = S(47)
                 RV(N+10) = VYCG
                 RV(N+15) = S(48)
```

```
CARD NO  ****                    CONTENTS                    ****
     630          RV(N+25) = S(49)
     640          RV(N+30) = S(50)
     641        65 CONTINUE
     642           GO TO 180
     643        C VERTICAL ON WING
     644          80 DO 85 N=1,5
     645             WT = RW(N) + S(48)
     646             XCG = (RW(N)* RW(N+5) + S(48)*S(47)) / WT
     647             YCG = (RW(N)*RW(N+10) + S(48)*VYCG)/WT
     648             ZCG = (RW(N)* RW(N+15) + S(48)*S(48)) / WT
     649             RW(N+25) = RW(N+25) + S(49) + RW(N)*((RW(N+5)-XCG)**2 + (RW(N+15)
     650            1           -ZCG)**2) + S(48)*((S(47)-XCG)**2 + (S(48)-ZCG)**2)
     651             RW(N+30) = RW(N+30) + S(50) + RW(N)*((RW(N+5) - XCG)**2)
     652            1         + S(48)*((S(47) - XCG)**2 + VYCG**2)
     653             RW(N) = WT
     654             RW(N+5) = XCG
     655             RW(N+10) = YCG
     656             RW(N+15) = ZCG
     657          85 CONTINUE
     658        C    - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     659        C ..........  ..........
     660        C IF AIR INDUCTION SYSTEM ON FUSELAGE (GO)(7)+/- ) ON WING (+0)
     661        C TEST FOR NACELLES.  DATS(1) = NUMBER OF NACELLES
     662          100 IF(DATS(1)) 150,150,102
     663        C       NACELLE TYPE
     664          102 IF(GO)(7)) 104,120,104
     665          104 DO 110 N=1,5
     666             RA(N) = S(51)
     667             RA(N+5) = S(53)
     668             RA(N+10) = S(54)
     669             RA(N+15) = DATS(.5)
     670             RA(N+25) = S(55)
     671             RA(N+30) = S(56) + S(51)*S(54)**2
     672          110 CONTINUE
     673             GO TO 150
     674        C ..........  ......
     675        C                    WING MOUNTED, TEST NUMBER
     676          120 IF(DATS(1) - 0(3)) 122,122,124
     677        C       TWO NACELLES
     678          122 WT = S(51)
     679             XCG = S(53)
     680             YCG = S(54)
     681             ZCG = DATS(15)
     682             S(116) = S(55)
     683             S(117) = S(56) + S(51)*S(54)**2
     684             GO TO 130
     685        C       FOUR NACELLES
     686          124 WT = S(51) + S(57)
     687             XCG = (S(52) + S(58))/WT
     688             YCG = (S(51)*S(54) + S(57)*S(60))/WT
     689             ZCG = (S(51)*DATS(15) + S(57)*DATS(18))/WT
     690             S(118) = (S(53) - XCG)**2
     691             S(119) = (S(58) - XCG)**2
     692             S(120) = (DATS(15) - ZCG)**2
     693             S(121) = (DATS(18) - ZCG)**2
     694             S(116) = S(55) + S(61) + S(51)*(S(118) + S(120)) +
     695            1 S(57)*(S(119) + S(121))
     696             S(117) = S(56) + S(62) + S(51)*(S(118) + S(54)**2) +
     697            1 S(57)*(S(119) + S(60)**2)
     698          130 DO 140 N=1,5
     699             S(118) = RW(N) * WT
     700             S(119) = (RW(N)*RW(N+5) + WT*XCG)/S(118)
     701             S(120) = (RW(N)*RW(N+10) + WT*YCG)/S(118)
     702             S(121) = (RW(N)*RW(N+15) + WT*ZCG)/S(118)
     703             S(124) = (RW(N+5) - S(119))**2
     704             S(125) = (XCG - S(119))**2
     705             S(126) = (RW(N+15) - S(121))**2
     706             S(127) = (ZCG - S(121))**2
     707             RW(N+25) = RW(N+25) + S(116) + RW(N)*(S(124) + S(126)) +
     708            1 WT*(S(125) + S(127))
     709             RW(N+30) = RW(N+30) + S(117) + RW(N)*S(124) + WT*S(125)
```

```
      RM(N) = S(118)
      RM(N+5) = S(119)
      RM(N+10) = S(120)
      RM(N+15) = S(121)
  140 CONTINUE
C  SPACE FOR OTHER
  150  CONTINUE
C
      DO 160 N=1,400
  160 FUSDMI(N) = 0.0
C
      DO 170 N=1,42
      N = 151 + (N-1)*5
      FUSDMI(N    ) = S(N  )
      FUSDMI(N+ 90) = S(N+1)
      FUSDMI(N+180) = S(N+2)
      FUSDMI(N+270) = S(N+3)
  170 FUSDMI(N+360) = S(N+4)
C
      DO 180 N=51,70
      I = N - 50
      FUSDMI(N    ) = WFC2(I)
      FUSDMI(N+ 90) = WFC2(I)
      FUSDMI(N+180) = WFC1(I)
      FUSDMI(N+270) = WFC3(I)
  180 FUSDMI(N+360) = WFC1(I)
C
      DO 190 N=71,90
      I = N - 70
      FUSDMI(N    ) = WFUS(I)
      FUSDMI(N+ 90) = WFUS(I)
      FUSDMI(N+180) = WFUS(I)
      FUSDMI(N+270) = WFUS(I)
  190 FUSDMI(N+360) = WFUS(I)
C
      DO 200 N=1,10
      FUSDMI(N+450) = ALT(N)
      FUSDMI(N+460) = VL(N)
  200 FUSDMI(N+470) = QL(N)
C
C CHECK PRINT
C
      IF(IP(45))5001,5001,5002
 5001 CONTINUE
      WRITE (6,300) RT,RM,RH,RV,RA,RO
  300 FORMAT(1H1,28HCHECK PRINT FOR AVDINR,6X,21H** AVDINR - IP(45) **,
     1                                    //3HRT, 9E10.6/6(3X,9E10.6/)
     1 /3HRM, 9E10.6 / 6( 3X,9E10.6/ ) /3HRH, 9E10.6 /6(3X,9E10.6 /)
     2 /3HRV, 9E10.6 / 6( 3X,9E10.6/ ) /3HRA, 9E10.6 /6(3X,9E10.6 /)
     3 /3HRO, 9E10.6 / 6( 3X,9E10.6/ ) )
 5002 CONTINUE
C
C       **********************************************************
C
      CALL WRITMS(1,FUSDMI(1),400,34)
C
C       **********************************************************
C
      RETURN
      END
C
C ***************************************************************************************
C                          SUBROUTINE AVDATA
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE AVDATA
C         AUGUST 1972.  SEPERATE MING AND CONTENTS FROM AVDATA
      COMMON TCOM(4320)
C
      DIMENSION S(700),SD(700),DV(2320),S(1400),ND(200)
      DIMENSION DVAT(1000)
```

```
CARD NO     ****                       CONTENTS                       ****

      DIMENSION        UZ(12),WWT1(12),WWT2(12),WWT3(12),
     1 XB11(12),XB21(12),YB11(12),YB21(12),XB12(12),XB22(12),
     2 YB12(12),YB22(12),XB13(12),XB23(12),YB13(12),YB23(12),
     3 YY12(12),YY22(12),YY21(12),YY23(12)
C
      EQUIVALENCE (D(1),TCOM(1)),(DD(1),TCOM(70)),(DV(1),TCOM(140)),
     1 (S(1),TCOM(372)),(ND(1),TCOM(412))
      EQUIVALENCE (DWWT(1),DV(112))
      EQUIVALENCE                    (UZ(1),DWWT(461)),
     1 (WWT1(1),DWWT(497)),(WWT2(1),DWWT(521)),(WWT3(1),DWWT(545)),
     2 (XB11(1),DWWT(569)),(XB21(1),DWWT(581)),(YB11(1),DWWT(593)),
     3 (YB21(1),DWWT(605)),(XB12(1),DWWT(617)),(XB22(1),DWWT(629)),
     4 (YB12(1),DWWT(641)),(YB22(1),DWWT(653)),(XB13(1),DWWT(665)),
     5 (XB23(1),DWWT(677)),(YB13(1),DWWT(689)),(YB23(1),DWWT(701)),
     6 (YY12(1),DWWT(713)),(YY22(1),DWWT(725)),(YY21(1),DWWT(737)),
     7 (YY23(1),DWWT(749))
C
C  SUM WING AND CONTENTS
      DO 105 I=1,12
      S(1) = S(1) + WWT1(I)
      S(2) = S(2) + WWT1(I)*XB11(I)
      S(3) = S(3) + WWT1(I)*XB21(I)
      S(4) = S(4) + WWT1(I)*YB11(I)
      S(5) = S(5) + WWT1(I)*YB21(I)
      S(6) = S(6) + WWT2(I)
      S(7) = S(7) + WWT2(I)*XB12(I)
      S(8) = S(8) + WWT2(I)*XB22(I)
      S(9) = S(9) + WWT2(I)*YB12(I)
      S(10) = S(10) + WWT2(I)*YB22(I)
      S(11) = S(11) + WWT3(I)
      S(12) = S(12) + WWT3(I)*XB13(I)
      S(13) = S(13) + WWT3(I)*XB23(I)
      S(14) = S(14) + WWT3(I)*YB13(I)
      S(15) = S(15) + WWT3(I)*YB23(I)
      S(16) = S(16) + YY12(I)
      S(17) = S(17) + YY22(I)
      S(18) = S(18) + WWT2(I)*UZ(I)
      S(89) = S(89) + YY21(I)
      S(90) = S(90) + YY23(I)
      S(91) = S(91) + WWT1(I)*UZ(I)
      S(92) = S(92) + WWT3(I)*UZ(I)
  105 CONTINUE
      S(19) = S(18)
      S(20) = S(2)/S(1)
      S(21) = S(3)/S(1)
      S(22) = S(4)/S(1)
      S(23) = S(5)/S(1)
      S(24) = S(7)/S(6)
      S(25) = S(8)/S(6)
      S(26) = S(9)/S(6)
      S(27) = S(10)/S(6)
      S(28) = S(12)/S(11)
      S(29) = S(13)/S(11)
      S(30) = S(14)/S(11)
      S(31) = S(15)/S(11)
      DO 110 I=1,12
      S(16) = S(16) + WWT2(I)*(S(24) - XB12(I))**2
      S(17) = S(17) + WWT2(I)*(S(25) - XB22(I))**2
      S(18) = S(18) + WWT2(I)*((S(24) - XB12(I))**2 + YB12(I)**2)
      S(19) = S(19) + WWT2(I)*((S(25) - XB22(I))**2 + YB22(I)**2)
      S(89) = S(89) + WWT1(I)*(S(21) - XB21(I))**2
      S(90) = S(90) + WWT3(I)*(S(29) - XB23(I))**2
      S(91) = S(91) + WWT1(I)*((S(21) - XB21(I))**2 + YB21(I)**2)
      S(92) = S(92) + WWT3(I)*((S(29) - XB23(I))**2 + YB23(I)**2)
  110 CONTINUE
      RETURN
      END
C
C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                          SUBROUTINE CONDST
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
```

CARD NO **** CONTENTS ****

```
      C
            SUBROUTINE CONDST
      C       WRITTEN 7 JULY 1972
      C       TO DISTRIBUTE FUSELAGE CONTENTS
            COMMON TCON(4320)
            DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
            DIMENSION EQU(200)
            DIMENSION GDI(20),GDMT(160),GDB(80),DATS(40),DATD(70)
            DIMENSION DVH(50),DVH(30),DVV(30),DVMT(500)
            EQUIVALENCE (D(1),TCON(1)),(GD(1),TCON(701)),(DV(1),TCON(1401)),
           1 (S(1),TCON(3721)),(ND(1),TCON(4121))
            EQUIVALENCE (EQU(1),D(81))
            EQUIVALENCE (GDI(1),GD(1)),(GDMT(1),GD(81)),(GDB(1),GD(381)),
           1 (DATS(1),GD(461)),(DATD(1),GD(501))
            EQUIVALENCE (DVH(1),DV(321)),(DVH(1),DV(371)),(DVV(1),DV(461)),
           1           (DVMT(1),DV(1121))
            EQUIVALENCE       -     (J,ND(102)),(K,ND(103)),(L,ND(104))
            EQUIVALENCE (NCD,ND(117))
          10 FORMAT(2B40*** WARNING FROM CONDST *** /
           1 28X,33HPREVIOUS FORE AND AFT LIMITS WERE, 2F10.2/
           228X,F12.2,17H LBS AT FUS. STA.,F8.2,10X,21HDISTRIBUTED BY DSTNOR)
      C       STORE OPERATIONAL WEIGHT EMPTY DISTRIBUTION IN DVMT(351-380)
      C       MAIN GEAR
            IF(DVMT(102)) 110,110,101
          101 S(1) = DVMT(102)
            S(2) = DVMT(142)
            CALL DSTNOR
            DO 103 I=J,K
            DVMT(I+350) = DVMT(I+350) + S(I+20)
          103 CONTINUE
      C       NOSE GEAR
          110 S(1) = DVMT(103)
            S(2) = DVMT(143)
            CALL DSTNOR
            DO 115 I=J,K
            DVMT(I+350) = DVMT(I+350) + S(I+20)
          115 CONTINUE
      C       SURFACE CONTROLS
            IF(DVMT(104)) 127,127,117
          117 S(1) = DVMT(104)
            S(2) = DVMT(144)
            CALL DSTNOR
            DO 120 I=J,K
            DVMT(I+350) = DVMT(I+350) + S(I+20)
          120 CONTINUE
      C       DISTRIBUTE SURFACE CONTROLS USING A TRAPEZOIDAL DISTRIBUTION
            S(1) = DVMT(105)
            S(2) = DVMT(145)
            S(3) = DVMT(144)
            S(4) = AMAX1(DVH(43),DVH(24),DVV(24))
            S(5) = (S(4)-S(3))/D(3)
            IF( (S(2)-S(3)) - S(5) ) 123,121,121
          121 IF( (S(4)-S(2)) - S(5) ) 123,122,122
          122 CALL DSTTRP
            GO TO 125
          123 CALL DSTNOR
            WRITE(6, 10) S(3),S(4),S(1),S(2)
            WRITE(6,124)
          124 FORMAT(1H+,18X,18HSURFACE CONTROLS)
          125 DO 126 I=J,K
            DVMT(I+350) = DVMT(I+350) + S(I+20)
          126 CONTINUE
      C       OTHER STRUCTURE
          127 IF(DVMT(106)) 130,130,128
          128 S(1) = DVMT(106)
            S(2) = DVMT(146)
            CALL DSTNOR
            DO 129 I=J,K
            DVMT(I+350) = DVMT(I+350) + S(I+20)
          129 CONTINUE
      C       FUEL SYSTEM
```

```
CARD NO ****                                CONTENTS                                ****
  123       130  IF(DWAT(107)) 139,139,131
  124       C                        WHEN DWAT(107)=0.0  THERE IS NO FUEL IN FUSELAGE
  125       131  S(1) = DWAT(107)
  126            S(2) = DWAT(147)
  127            S(4) = AMAX1(DDWT(110),DDWT(112),DDWT(114),DDWT(116),DDWT(118))
  128            S(3) = S(4)
  129            DO 132 I=1,9,2
  130            IF(DDWT(I+100).EQ.0.0)GO TO 132
  131            IF(S(3).GT.DDWT(I+100))S(3)=DDWT(I+100)
  132       132  CONTINUE
  133            S(5) = (S(4)-S(3))/D(3)
  134            IF( (S(2)-S(3)) - S(5) ) 135,133,133
  135       133  IF( (S(4)-S(2)) - S(5) ) 135,134,134
  136       134  CALL DSTTRP
  137            GO TO 137
  138       135  CALL DSTNOR
  139            WRITE(6, 10) S(3),S(4),S(1),S(2)
  140            WRITE(6,136)
  141       136  FORMAT(1H+,24X,11HFUEL SYSTEM)
  142       137  DO 138  I=J,K
  143            DWAT(I+350) = DWAT(I+350) + S(I+20)
  144       138  CONTINUE
  145       C    ENGINE CONTROLS
  146       139  IF(DWAT(108)) 149,149,140
  147       140  S(1) = DWAT(108)
  148            S(2) = DWAT(148)
  149            S(3) = DWAT(144)
  150            S(4) = DWAT(60)
  151            IF(DATS(1)) 146,146,142
  152       142  IF(QD(17)) 144,144,146
  153       144  S(4) = DWAT(43)
  154       146  S(5) = (S(4)-S(3))/D(3)
  155            IF( (S(2)-S(3)) - S(5) ) 1463,1461,1461
  156       1461 IF( (S(4)-S(2)) - S(5) ) 1463,1462,1462
  157       1462 CALL DSTTRP
  158            GO TO 1465
  159       1463 CALL DSTNOR
  160            WRITE(6, 10) S(3),S(4),S(1),S(2)
  161            WRITE(6,1464)
  162       1464 FORMAT(1H+,20X,15HENGINE CONTROLS)
  163       1465 DO 148 I=J,K
  164            DWAT(I+350) = DWAT(I+350) + S(I+20)
  165       148  CONTINUE
  166       C      AUXILIARY POWER UNIT
  167       149  IF(DWAT(109)) 160,160,156
  168       150  S(1) = DWAT(109)
  169            S(2) = DWAT(149)
  170            CALL DSTNOR
  171            DO 152 I=J,K
  172            DWAT(I+350) = DWAT(I+350) + S(I+20)
  173       152  CONTINUE
  174       C      INSTRUMENTS - BREAK INTO DISTRIBUTED AND CONCENTRATED WEIGHTS
  175       160  IF(DWAT(110)) 166,166,161
  176       161  S(1) = EQU(139)*DWAT(21)
  177            S(2) = DWAT(144)
  178            CALL DSTNOR
  179            DO 162 I=J,K
  180            DWAT(I+350) = DWAT(I+350) + S(I+20)
  181       162  CONTINUE
  182            S(3) = DWAT(110) - S(1)
  183            S(4) = (DWAT(110)*DWAT(150) - S(1)*S(2))/S(3)
  184            S(1) = S(3)
  185            S(3) = S(2)
  186            S(2) = S(4)
  187            S(4) = AMAX1(DWAT(43),DWAT(24),DWV(24))
  188            S(5) = (S(4)-S(3))/D(3)
  189            IF( (S(2)-S(3)) - S(5) ) 1623,1621,1621
  190       1621 IF( (S(4)-S(2)) - S(5) ) 1623,1622,1622
  191       1622 CALL DSTTRP
  192            GO TO 163
  193       1623 CALL DSTNOR
```

```
CARD NO  ****                                  CONTENTS                                  ****
  004          WRITE(6, 10) S(3),S(4),S(1),S(2)
  005          WRITE(6,1624)
  006       1624 FORMAT(1H+,2HX,11HINSTRUMENTS)
  007        163 DO 164 I=J,K
               DVAT(I+350) = DVAT(I+350) + S(I+20)
  009        164 CONTINUE
 1000     C        HYDRAULICS
 1001        165 IF(DVAT(111)) 171,171,167
 1002        167 S(1) = DVAT(111)
               S(2) = DVAT(151)
               S(3) = DVAT(144)
 1005          S(4) = AMAX1(DVH(43),DVH(24),DVV(24))
 1006          S(5) = (S(4)-S(3))/D(3)
 1007          IF( (S(2)-S(3)) - S(5) ) 1683,1681,1681
 1008       1681 IF( (S(4)-S(2)) - S(5) ) 1683,1682,1682
 1009       1682 CALL DSTTRP
 1010          GO TO 169
 1011       1683 CALL DSTNOR
               WRITE(6, 10) S(3),S(4),S(1),S(2)
 1013          WRITE(6,1684)
 1014       1684 FORMAT(1H+,25X,10HHYDRAULICS)
 1015        169 DO 170 I=J,K
               DVAT(I+350) = DVAT(I+350) + S(I+20)
 1017        170 CONTINUE
 1018     C        ELECTRICAL
 1019        171 IF(DVAT(112)) 175,175,172
 1020        172 S(1) = DVAT(112)
               S(2) = DVAT(152)
               S(3) = DVAT(144)
 1023          S(4) = AMAX1( DVH(43),DVH(24),DVV(24))
 1024          S(5) = (S(4)-S(3))/D(3)
 1025          IF( (S(2)-S(3)) - S(5) ) 1733,1731,1731
 1026       1731 IF( (S(4)-S(2)) - S(5) ) 1733,1732,1732
 1027       1732 CALL DSTTRP
 1028          GO TO 1735
 1029       1733 CALL DSTNOR
 1030          WRITE(6, 10) S(3),S(4),S(1),S(2)
 1031          WRITE(6,1734)
 1032       1734 FORMAT(1H+,25X,10HELECTRICAL)
 1033       1735 DO 174 I=J,K
 1034          DVAT(I+350) = DVAT(I+350) + S(I+20)
 1035        174 CONTINUE
 1036     C        ELECTRONICS
 1037        175 IF(DVAT(113)) 200,200,176
 1038        176 IF(GDVT(153)) 180,180,190
 1039     C        TWO COMPARTMENTS
 1040        180 IF(GDVT(151)+GDVT(152)) 181,181,182
 1041        181 S(1)=DVAT(113)
               S(2)=DVAT(153)
               GO TO 185
 1044        182 S(1) = DVAT(113)*(DVAT(153) - GDVT(151))/(GDVT(152) - GDVT(151))
 1045          S(2) = GDVT(152)
 1046          CALL DSTNOR
 1047          DO 184 I=J,K
 1048          DVAT(I+350) = DVAT(I+350) + S(I+20)
 1049        184 CONTINUE
 1050          S(1) = DVAT(113)*(GDVT(152) - DVAT(153))/(GDVT(152) - GDVT(151))
 1051          S(2) = GDVT(151)
 1052        185 CALL DSTNOR
 1053          DO 186 I=J,K
 1054          DVAT(I+350) = DVAT(I+350) + S(I+20)
 1055        186 CONTINUE
 1056          GO TO 200
 1057     C        THREE COMPARTMENTS
 1058        190 S(1) = DVAT(113)*EQU(42)
 1059          S(2) = GDVT(152)
 1060          CALL DSTNOR
 1061          DO 192 I=J,K
 1062          DVAT(I+350) = DVAT(I+350) + S(I+20)
 1063        192 CONTINUE
 1064          S(3) = DVAT(113)*(D(1) - EQU(42))
```

```
CARD NO  ****                    CONTENTS                    ****
    1066          $(6) = (DVAT(113)*DVAT(153) - DVAT(113)*EQU(142)*GDAT(152))/$(5)
    1066          $(1) = $(5)*($(6) - GDAT($(1)))/(GDAT(153) - GDAT($(1)))
    1067          $(2) = GDAT(153)
    1068          CALL DSTNOR
    1069          DO 194 I=J,K
    1070          DVAT(I+350) = DVAT(I+356) + $(I+20)
    1071      194 CONTINUE
    1072          $(1) = $(5)*(GDAT(153) - $(6))/(GDAT(153) - GDAT($(1)))
    1073          $(2) = GDAT($(1))
    1074          CALL DSTNOR
    1075          DO 195 I=J,K
    1076          DVAT(I+350) = DVAT(I+350) + $(I+20)
    1077      195 CONTINUE
    1078      C       ARMAMENT
    1079        200 IF(DVAT(114)) 300,300,202
    1080        202 $(1) = DVAT(114)
    1081            $(2) = DVAT(154)
    1082            IF(DVAT(129)) 220,220,204
    1083        204 IF(EQU(138)*DVAT(114) - EQU(143)) 205,205,206
    1084        205 $(1) = EQU(138)*DVAT(114)
    1085            GO TO 207
    1086        206 $(1) = EQU(143)
    1087        207 $(2) = DVAT(93)
    1088            CALL DSTNOR
    1089            DO 209 I=J,K
    1090            DVAT(I+350) = DVAT(I+350) + $(I+20)
    1091        209 CONTINUE
    1092            $(5) = DVAT(114) - $(1)
    1093            $(2) = (DVAT(114)*DVAT(154) -   $(1) * DVAT(93))/$(5)
    1094            $(1) = $(5)
    1095        220 $(3) = DVAT(144)
    1096            $(4) = D(2)*$(2) - $(3)
    1097            $(5) = ($(4)-$(3))/D(3)
    1098            IF( ($(2)-$(3)) - $(5) ) 2213,2211,2211
    1099       2211 IF( ($(4)-$(2)) - $(5) ) 2213,2212,2212
    1100       2212 CALL DSTTRP
    1101            GO TO 2215
    1102       2213 CALL DSTNOR
    1103            WRITE(6, 10) $(3),$(4),$(1),$(2)
    1104            WRITE(6,2214)
    1105       2214 FORMAT(1H0,27X,8HARMAMENT)
    1106       2215 DO 222 I=J,K
    1107            DVAT(I+350) = DVAT(I+350) + $(I+20)
    1108        222 CONTINUE
    1109      C       FURNISHINGS
    1110        300 IF(DVAT(115)) 311,311,301
    1111        301 $(1) = DVAT(115)
    1112            $(2) = DVAT(155)
    1113            $(3) = DVAT(144)
    1114            $(4) = D(2)*$(2) - $(3)
    1115            IF(GDI(1) - D(3)*D(10)) 303,302,302
    1116      C       TRANSPORT
    1117        302 $(4) = GDAT(102)
    1118        303 $(5) = ($(4)-$(3))/D(3)
    1119            IF( ($(2)-$(3)) - $(5) ) 306,304,304
    1120        304 IF( ($(4)-$(2)) - $(5) ) 306,305,305
    1121        305 CALL DSTTRP
    1122            GO TO 309
    1123        306 CALL DSTNOR
    1124            WRITE(6, 10) $(3),$(4),$(1),$(2)
    1125            WRITE(6,307)
    1126        307 FORMAT(1H0,24X,11HFURNISHINGS)
    1127        309 DO 310 I=J,K
    1128            DVAT(I+350) = DVAT(I+350) + $(I+20)
    1129        310 CONTINUE
    1130      C       AIR CONDITIONING
    1131        311 IF(DVAT(116)) 315,315,312
    1132        312 $(1) = DVAT(116)
    1133            $(2) = DVAT(156)
    1134            $(3) = DVAT(144)
    1135            $(4) = AMAX1(DVAT(43),DVAT(24),DVAT(24),GDAT(153))
```

```
CARD NO ****                    CONTENTS                    ****
   1136          $(9) = ($(4)-$(3))/D(3)
   1137          IF( ($(2)-$(3)) - $(5) ) 3123,3121,3121
   1138    3121  IF( ($(4)-$(2)) - $(9) ) 3123,3122,3122
   1139    3122  CALL DSTRP
   1140          GO TO 313
   1141    3123  CALL DSTNOR
   1142          WRITE(6, 10) $(3),$(4),$(1),$(2)
   1143          WRITE(6,312)
   1144     312  FORMAT(1H+ 19X,10HAIR CONDITIONING)
   1145     313  DO 314 I=J,K
   1146          DWT(I+360) = DWT(I+360) + $(I+20)
   1147     314  CONTINUE
   1148     315  CONTINUE
   1149          DO 419 II=1,10
   1150    C             PHOTOGRAPHIC
   1151    C             AUXILIARY GEAR
   1152    C             OTHER EQUIPMENT
   1153    C             CREW
   1154    C             TRAPPED FUEL
   1155    C             LIQUID NITROGEN
   1156    C             MISC
   1157    C             GUNS
   1158    C             PYLONS
   1159    C             EXTERNAL TANKS
   1160          IF(DWT(II+118)) 419,419,402
   1161     402  $(1) = DWT(II+118)
   1162          $(2) = DWT(II+158)
   1163          IF(II  EQ  5) GO TO 404
   1164          CALL DSTNOR
   1165          GO TO 417
   1166     404  $(4) = AMAX1(GDWT(118),GDWT(112),GDWT(114),GDWT(116),GDWT(181))
   1167          $(3) = $(4)
   1168          DO 405 I=1,9,2
   1169          IF(GDWT(I+108) EQ 0(2))GO TO 405
   1170          IF($(3) GT GDWT(I+108))$(3)=GDWT(I+108)
   1171     405  CONTINUE
   1172          $(5) = ($(4)-$(3))/D(3)
   1173          IF( ($(2)-$(3)) - $(5) ) 408,406,408
   1174     406  IF( ($(4)-$(2)) - $(5) ) 408,407,407
   1175     407  CALL DSTRP
   1176          GO TO 417
   1177     408  CALL DSTNOR
   1178          WRITE(6, 10) $(3),$(4),$(1),$(2)
   1179          WRITE(6,409)
   1180     409  FORMAT(1H+,23X,12HTRAPPED FUEL)
   1181     417  DO 418 I=J,K
   1182          DWT(I+360) = DWT(I+360) + $(I+20)
   1183     418  CONTINUE
   1184     419  CONTINUE
   1185    C        TEST FOR NACELLE OR BURIED ENGINE
   1186          IF(DATS(1)) 420,420,500
   1187    C             ENGINE SECTION
   1188     420  $(1) = DWT(261)
   1189          $(2) = DWT(201)
   1190          CALL DSTNOR
   1191          DO 422 I=J,K
   1192          DWT(I+360) = DWT(I+360) + $(I+20)
   1193     422  CONTINUE
   1194    C             ENGINE AND EXHAUST SYSTEM
   1195          $(1) = DWT(262) + DWT(266)
   1196          $(2) = (DWT(262)*DWT(262) + DWT(266)*DWT(265))/$(1)
   1197          CALL DSTNOR
   1198          DO 424 I=J,K
   1199          DWT(I+360) = DWT(I+360) + $(I+20)
   1200     424  CONTINUE
   1201          IF(DWT(263)) 427,427,425
   1202    C             ACCESSORIES AND GEAR BOXES
   1203     425  $(1) = DWT(263)
   1204          $(2) = DWT(203)
   1205          CALL DSTNOR
   1206          DO 426 I=J,K
```

```
 CARD NO       ****                          CONTENTS                           ****

  1207            DWAT(I+360) = DWAT(I+360) + S(I+20)
  1208        426 CONTINUE
  1209        C             A I S  STRUCTURE
  1210        427 S(1) = DWAT(204)
  1211            S(2) = DWAT(204)
  1212            S(3) = DATS(13)
  1213            S(4) = DATS(13) + DATD(NCD+10)
  1214            S(5) = (S(4)-S(3))/D(3)
  1215            IF( (S(2)-S(3)) - S(5) ) 4203,4201,4201
  1216       4201 IF( (S(4)-S(2)) - S(5) ) 4203,4202,4202
  1217       4202 CALL DSTTRP
  1218            GO TO 429
  1219       4203 CALL DSTNOR
  1220            WRITE(6, 10) S(3),S(4),S(1),S(2)
  1221            WRITE(6,4204)
  1222       4204 FORMAT(1H+,20X,15HA I S STRUCTURE)
  1223        429 DO 430 I=J,K
  1224            DWAT(I+360) = DWAT(I+360) + S(I+20)
  1225        430 CONTINUE
  1226        C             A I S  ACTUATORS AND MECHANISMS
  1227            IF(DWAT(265)) 435,435,431
  1228        431 S(1) = DWAT(265)
  1229            S(2) = DWAT(285)
  1230            CALL DSTNOR
  1231            DO 434 I=J,K
  1232            DWAT(I+360) = DWAT(I+360) + S(I+20)
  1233        434 CONTINUE
  1234        435 DO 450 II=1,9
   235        C             COOLING AND DRAINS
  1236        C             LUBRICATION SYSTEMS
  1237        C             STARTING SYSTEM
  1238        C             A P U
  1239        C             INSTRUMENTS
  1240        C             HYDRAULICS
  1241        C             ELECTRICAL
  1242        C             AIR CONDITIONING
  1243        C             OIL
  1244            IF(DWAT(II+266)) 450,450,440
  1245        440 S(1) = DWAT(II+266)
  1246            S(2) = DWAT(II+286)
  1247            CALL DSTNOR
  1248            DO 444 I=J,K
  1249            DWAT(I+360) = DWAT(I+360) + S(I+20)
  1250        444 CONTINUE
  1251        450 CONTINUE
  1252        900 CONTINUE
  1253            RETURN
  1254            END
  1255        C
  1256        C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
  1257        C                         SUBROUTINE DBLCNT
  1258        C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  1259        C
  1260            SUBROUTINE DBLCNT
  1261        C
  1262        C       THIS ROUTINE SETS UP VEHICLE DATA IN BC ARRAY FOR USE
  1263        C       IN AIRLOADS MODULE ROUTINE BLCNTL
  1264        C
  1265            COMMON TCOM(4320)
  1266        C
  1267            DIMENSION D(700),DD(700),DV(2320),S(400),ND(200)
  1268        C
  1269            DIMENSION BC(200)
  1270        C
  1271            DIMENSION DD1(20),DDD(30),DATM(40),DCM(50)
  1272            DIMENSION DWI(50),DWH(30),DWV(30),DWAT(1000),ALT(10),VH(10),
  1273           * VL(10)
  1274        C
  1275            EQUIVALENCE (D(1),TCOM(1)),(DD(1),TCOM(701)),(DV(1),TCOM(1401))
  1276           * , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
  1277        C
```

CARD NO **** CONTENTS ****

```
      EQUIVALENCE (GD1(1),GD(1)),(GDD(1),GD(21)),(DATM(1),GD(51))
     * , (GDM(1),GD(75))
C
      EQUIVALENCE (DWH(1),DV(32)),(DWH(1),DV(37)),(DVV(1),DV(40))
     * , (DWAT(1),DV(112)),(ALT(1),DV(1)),(VH(1),DV(6))
     * , (VL(1),DV(7))
C
      EQUIVALENCE (BC(1),BV(212))
C
C     SETUP BC ARRAY DATA
      BC(1) = DWAT(94)
      BC(2) = DWAT(96)
      BC(3) = DWAT(96)
      BC(4) = DWAT(95)
      BC(5) = DWAT(93)
      BC(6) = DWAT(92)
      BC(7) = DWAT(98)/GD(17)/D(26)
      BC(8) = DWAT(97)/GD(17)/D(26)
      BC(9) = DWAT(96)/GD(17)/D(26)
      BC(10) = DWAT(99)/GD(17)/D(26)
      BC(11) = DWAT(91)
      BC(12) = DWAT(93)
C
      DO 410 I=11,16
      BC(I+2) = GDD(I)
  410 CONTINUE
C
C     DO SPEED PROFILE
      J = 5
      K = 7
      IF(GD1(19)) 414,414,412
  412 J = GD1(19)+D(2) - D(1)
  414 IF(GD1(20)) 416,416,416
  416 K = GD1(20)+D(2) - D(1)
  418 BC(19) = ALT(I)
      BC(20) = ALT(J)
      BC(21) = ALT(K)
      BC(22) = VH(I)
      BC(23) = VH(J)
      BC(24) = VH(K)
      BC(166) = VL(I)
      BC(167) = VL(J)
      BC(168) = VL(K)
      DO 420 I=1,6
      BC(I+24) = DATM(I+32)
  420 CONTINUE
      BC(31) = GDD(17)
      BC(32) = GDD(18)
C
C     TRANSFER CARRY-OVER LIFT REDUCTION FACTORS
      BC(163) = GD1(18)
      BC(164) = GD1(17)
      BC(165) = GD1(18)
C
C     TRANSFER WING GEOMETRY DATA
      DO 430 I=1,21
      BC(I+35) = DWH(I)
  430 CONTINUE
C
C     DO FORWARD WING POSITION
      DO 440 I=22,41
      BC(I+47) = DWH(I)
  440 CONTINUE
      BC(108) = GDM(37)
      BC(101) = GDM(38)
      BC(102) = GDM(39)
      BC(103) = GDM(40)
C
C     TRANSFER HORIZONTAL TAIL DATA
      DO 450 I=1,80
      BC(I+103) = DWH(I)
```

```
CARD NO  ****                          CONTENTS                          ****

 1349      480 CONTINUE
 1350      C
 1351      C       TRANSFER VERTICAL TAIL DATA
 1352            DO 490 I=1,21
 1353            BC(I+138) = BVV(I)
 1354      490 CONTINUE
 1355            BC(159) = BVV(26)
 1356      C
 1357            RETURN
 1358            END
 1359      C
 1360      C ***************************************************************************
 1361      C                          SUBROUTINE BCONTL
 1362      C ***************************************************************************
 1363      C
 1364            SUBROUTINE BCONTL
 1365      C
 1366      C       THIS ROUTINE SETS UP GEOMETRY AND INERTIA DATA IN MD ARRAY
 1367      C       FOR USE IN WING AND EMPENNAGE MODULE ROUTINE CONTL
 1368      C
 1369            COMMON TCON(4320)
 1370            COMMON /MISC/ MISC(100)
 1371            COMMON /IPRINT/ IP(100)
 1372      C
 1373            DIMENSION B(700),GD(700),BV(2320),S(400),MD(200)
 1374      C
 1375            DIMENSION GD(120)
 1376            DIMENSION GDD(30),GDWT(160),GDH(50),GDH(40),GDV(40),DATS(40)
 1377      C
 1378            DIMENSION BWH(50),BVH(30),BVV(30),BWT(1000)
 1379            DIMENSION DSP(10)
 1380      C
 1381            DIMENSION MD(200)
 1382      C
 1383            EQUIVALENCE (B(1),TCON(1)),(GD(1),TCON(701)),(BV(1),TCON(1401))
 1384           * , (S(1),TCON(3721)),(MD(1),TCON(4121))
 1385      C
 1386            EQUIVALENCE (GD(1),GD(1))
 1387            EQUIVALENCE (GDD(1),GD(21)),(GDWT(1),GD(91)),(GDH(1),GD(251))
 1388           * , (GDH(1),GD(301)),(GDV(1),GD(341)),(DATS(1),GD(401))
 1389      C
 1390            EQUIVALENCE (BWH(1),BV(321)),(BVH(1),BV(371)),(BVV(1),BV(401))
 1391           * , (BWT(1),BV(1121))
 1392            EQUIVALENCE (DSP(1),BV(31))
 1393      C
 1394            EQUIVALENCE (MD(1),S(1))
 1395      C
 1396      C
 1397      C*********
 1398      C       CLEAR TRANSFER REGION
 1399      C
 1400            DO 100 I=1,200
 1401            MD(I) = B(2*)
 1402       100 CONTINUE
 1403      C
 1404      C*********
 1405      C
 1406      C
 1407      C********
 1408      C
 1409      C          ***SUM TOTAL FUEL AND DELTA FUEL TO DESIGN***
 1410            DO 1000 I=1,7
 1411            MD(183) = MD(183) + GDWT(I+83)*GDWT(I+123)
 1412            MD(184) = MD(184) + GDWT(I+83)*GDWT(I+133)
 1413       1000 CONTINUE
 1414            MD(184) = MD(183) - MD(184)
 1415      C
 1416      C          ***TEMP CALC FOR DETLA USEFUL LOAD TO DESIGN--LB/AV***
 1417            MD(188) = (GDWT(121) - GDWT(131)*GDWT(81) + (GDWT(122) - GDWT(132
 1418           1))*GDWT(82) + (GDWT(123) - GDWT(133))*GDWT(83)
 1419      C
```

```
C*********
C
C       TRANSFER DATA FOR USE IN SURFACE GEOMETRY INPUT DATA
      MD(2) = DVAR(994)
      MD(3) = DVAR(995)
      MD(4) = GDD(11)
C
C**********
C
      MD(5) = GDD(13)
C
C**********
C
      MD(6) = DEP(5)
      MD(7) = GDM(16)
      MD(8) = GDM(5)
      MD(9) = DVAR(101)/D(2)/MD(3)
      IF(GDD(42)) 1003,1003,1002
C
C       VARIABLE SWEEP WING - TRANSFER PIVOT DATA AND SWEEP ANGLES
 1002 MD(10) = GDM(31)
      MD(11) = GDM(32) - DVAR(47) - GDM(31)*TAN(DVAR(48)*D(18))
      MD(12) = GDM(34)
      MD(13) = GDM(33)
 1003 MD(14) = DVAR(47)
      MD(15) = GDM(1)
      MD(16) = GDM(2)
      MD(17) = GDM(4)
      MD(18) = GDM(3)
      MD(19) = GDM(7)*D(2)
      IF(GDMT(84)) 1010,1010,1005
 1005 MD(20) = (DVAR(104) + DVAR(105))/(GDMT(84) + GDMT(85))
 1010 MD(21) = DVAR(103)/D(2)
      MD(22) = (DVAR(105) + DVAR(106))/D(2)
      MD(23) = D(1)
      MD(24) = D(2)
      DO 1020 I=1,11
      MD(I+24) = DVAR(I+988)
 1020 CONTINUE
      MD(35) = GDMT(84)/D(2)
      MD(37) = DVAR(44)/D(2)
      MD(38) = GDMT(94)
      MD(39) = GDMT(105)
      MD(40) = GDMT(106)
      MD(41) = GDMT(85)/D(2)
      MD(42) = DVAR(45)/D(2)
      MD(43) = GDMT(95)
      MD(44) = GDMT(107)
      MD(45) = GDMT(108)
C
C*********
C
C       ***ADDITIONAL WING DATA***
C
      MD(100) = GDM(10)
      MD(101) = GDM(11)
      MD(102) = GDM(12)
      MD(103) = GDM(13)
C
C         ***FUEL DENSITY--LB/IN**3***
      MD(104) = GDM(30)
C          ***MAX POSITIVE LOAD FACTOR***
      MD(105) =-GDD(11)
C
      IF (MD(105) - GDD(12)) 1021,1022,1022
 1021 MD(105) = GDD(12)
 1022 IF (MD(105) - GDD(41)) 1023,1024,1024
 1023 MD(105) = GDD(14)
 1024 CONTINUE
C
C**********
```

CARD NO **** CONTENTS ****

```
      C
            IF(QD(7)) 1030,1030,1040
 1030 IF(DVAR(97)) 1040,1040,1032
 1032 MD(46) = DVAR(97)/D(2)
      MD(47) = DATS(14)
      MD(48) = DVAR(975)
      MD(49) = DATS(15) - QDM(11)
      MD(50) = D(1)
      MD(51) = DVAR(977)/D(2)
C        NO DATA FOR 10X AT PRESENT
      MD(53) = DVAR(978)/D(2)
      IF(DVAR(979)) 1040,1040,1035
 1035 MD(54) = DVAR(979)/D(2)
      MD(55) = DATS(17)
      MD(56) = DVAR(980)
      MD(57) = DATS(18) - QDM(11)
      MD(58) = D(1)
      MD(59) = DVAR(982)/D(2)
C        NO DATA FOR 10X AT PRESENT
      MD(61) = DVAR(983)/D(2)
 1040 IF(DVAR(182)) 1050,1050,1045
 1045 MD(62) = DVAR(182)/D(2)
      MD(63) = QDD(29)
      MD(84) = DVAR(202)
C
C        TEST ON PAYLOAD
 1050 IF(GDMT(82)) 1070,1070,1052
 1052 IF(GDMT(104)) 1054,1054,1060
C        INBOARD PAYLOAD ONLY
 1054 MD(70) = GDMT(82)/D(2)
      MD(71) = GDMT(103)
      MD(72) = GDMT(92)
      IF(GDMT(37)) 1070,1070,1056
 1056 MD(86) = GDMT(37)/D(2)
      MD(87) = GDMT(103)
      MD(88) = GDMT(77)
      GO TO 1070
C        INBOARD AND OUTBOARD PAYLOAD
 1060 MD(70) = GDMT(82)/D(4)
      MD(71) = GDMT(103)
      S(201) = ((GDMT(104) - GDMT(103))*TAN(DVAR(46)*D(18)))/D(2)
      MD(72) = GDMT(92) - S(201)
      MD(78) = GDMT(82)/D(4)
      MD(79) = GDMT(104)
      MD(80) = GDMT(92) + S(201)
      IF(GDMT(37)) 1070,1070,1062
 1062 MD(86) = GDMT(37)/D(4)
      MD(87) = GDMT(103)
      MD(88) = GDMT(77) - S(201)
      MD(94) = GDMT(37)/D(4)
      MD(95) = GDMT(104)
      MD(96) = GDMT(77) + S(201)
C
C        HORIZONTAL TAIL DATA
 1070 MD(111) = DVAR(22)
      MD(112) = D(24)
      MD(113) = DVAR(2211)/D(2)/MD(3)
      MD(114) = DVAR(4)
      MD(115) = DVAR(8)
      MD(116) = DVAR(7)
      MD(117) = DVAR(2)
      MD(118) = DVAR(6)
      MD(119) = DVAR(10) + 2.0
      MD(120) = DVAR(222)/D(2)
      MD(122) = D(2)
      DO 1075 I=1,11
      MD(I+122) = DVAR(I+10)
 1075 CONTINUE
C
C*********
C
```

```
CARD NO  ****                          CONTENTS                          ****
 1952    C        ***ADDITIONAL HORI DATA***
 1953          MD(163) = GDH(10)
 1954          MD(164) = GDH(11)
 1955          MD(165) = GDH(12)
 1956          MD(166) = GDH(13)
 1957    C
 1958    C*********
 1959    C
 1970    C        VERTICAL TAIL DATA
 1971          MD(134) = GDV(10)
 1972          MD(135) = GDV(5)
 1973          MD(136) = DWT(241)/DVV(23)/WD(3)
 1974          MD(137) = DVV(3) + DVV(9)*TAN(DVV(1)*D(16))
 1975          MD(138) = GDV(1)*D(2)
 1976          MD(139) = GDV(2)*D(2)
 1977          MD(140) = GDV(4)
 1978          MD(141) = GDV(3)
 1979          MD(142) = D(24)
 1980          MD(143) = DWT(242)/DVV(23)
 1981          MD(145) = D(2)
 1982          DO 1005 I=1,11
 1983          MD(I+145) = DVV(I+8) - DVV(9)
 1984     1005 CONTINUE
 1985    C
 1986    C*********
 1987    C
 1988    C        ***ADDITIONAL VERT DATA***
 1989    C         **NO OF VERT PNLS***
 1990          MD(156) = 1.0
 1991          IF (GDV(7)) 1077,1077,1076
 1992     1076 MD(156) = 2.0
 1993    C         ***TYPE OF TAILS--0=CONV., 1=T-TAIL***
 1994     1077 MD(157) = 0.0
 1995          IF (XMISC(53)) 1079,1079,1078
 1996     1078 MD(157) = 1.0
 1997     1079 MD(167) = GDV(10)
 1998          MD(168) = GDV(11)
 1999          MD(169) = GDV(12)
 2000          MD(170) = GDV(13)
 2001    C
 2002    C*********
 2003    C
 2004          IF(IP(47)) 9001,9001,9002
 2005     9001 CONTINUE
 2006          WRITE(6,9000) (I,MD(I),MD(I+1),MD(I+2),MD(I+3),MD(I+4),I=1,200,5)
 2007     9000 FORMAT(1H0 MD(I) FROM DCCNTL,70X,21H** DCCNTL - IP(47) **/
 2008         1  (110, 5F10.4))
 2009     9002 CONTINUE
 2010    C
 2011    C        ********************************************************
 2012    C
 2013          CALL WRITMS(1,MD(1),200,21)
 2014    C
 2015    C        ************************************************************
 2016    C
 2017          RETURN
 2018          END
 2019    C
 2020    C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
 2021    C                              SUBROUTINE DFATHD
 2022    C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
 2023    C
 2024          SUBROUTINE DFATHD
 2025    C
 2026    C       THIS ROUTINE SETS UP INERTIAL LOADS DATA IN BC ARRAY FOR
 2027    C       USE IN AIRLOADS MODULE FATIGUE EVALUATION ROUTINE FATHD
 2028    C
 2029          COMMON TCOM(4320)
 2030          COMMON /MISC/ XMISC(100)
 2031    C
 2032          DIMENSION D(700),GD(700),GV(2320),G(400),MD(200)
```

```
CARD NO      ****                          CONTENTS                          ****
  1633          *, DVWT(1000)
  1634          *, BD(20), DATS(40), DVW(50), BC(200), WWT1(12), WWT2(12)
  1635          *, WWT3(12), XB11(12), XB21(12), YB11(12), YB21(12), XB12(12)
  1636          *, XB22(12), YB12(12), YB22(12), XB13(12), XB23(12), YB13(12)
  1637          *, YB23(12), BOD(30)
  1638      C
  1639           DIMENSION DW(12),XDW1(12),XDW2(12),YDW1(12),YDW2(12)
  1640      C
  1641           EQUIVALENCE (TCOM(1),D(1)), (TCOM(701),OD(1)), (TCOM(1401),DV(1))
  1642          *, (TCOM(3721),S(1)), (TCOM(4121),ND(1))
  1643          *, (DV(321),DW(1)), (DV(1121),DWT(1)), (DV(2121),BC(1))
  1644          *, (BD(1),OD(1)), (OD(21),ODD(1)), (OD(481),DATS(1))
  1645          *, (YEA1,DWT(394)), (XEA1,DWT(395))
  1646          *, (YEA2,DWT(396)), (XEA2,DWT(397))
  1647          *, (YSF ,DW(1))
  1648           EQUIVALENCE          (WWT1(1),DWT(497)), (WWT2(1),DWT(521))
  1649          *, (WWT3(1),DWT(545)), (XB11(1),DWT(569)), (XB21(1),DWT(581))
  1650          *, (YB11(1),DWT(593)), (YB21(1),DWT(605)), (XB12(1),DWT(617))
  1651          *, (XB22(1),DWT(629)), (YB12(1),DWT(641)), (YB22(1),DWT(653))
  1652          *, (XB13(1),DWT(665)), (XB23(1),DWT(677)), (YB13(1),DWT(689))
  1653          *, (YB23(1),DWT(701))
  1654      C
  1655           EQUIVALENCE (DW(1),DWT(872))
  1656          *, (XDW1(1),DWT(884)),(XDW2(1),DWT(908))
  1657          *, (YDW1(1),DWT(920)),(YDW2(1),DWT(932))
  1658      C
  1659      C
  1660      C  IF HAVE WING MOUNTED NACELLS SET-UP SO THEY CAN BE ADDED IN TO MOM.
  1661           DO 70 N=1,25
  1662           S(N+100) = D(24)
  1663          70 CONTINUE
  1664           IF(OD(17).NE.D(24)) GO TO 80
  1665      C  SO KNOW HAVE INBOARD PAIR AT LEAST - SUM WTS AND MOMENTS OF NACELLE
  1666      C                                       AND CONTENTS
  1667           S(116) = DWT(974)
  1668           S(117) = DWT(975)
  1669           S(118) = DATS(14)
  1670      C  TEST IF HAVE MORE THAN 2, IF SO DO OUTBOARD PAIR
  1671           IF(DATS(1) - D(2)) 80,80,76
  1672          76 S(119) = DWT(979)
  1673           S(120) = DWT(980)
  1674           S(121) = DATS(17)
  1675          80 CONTINUE
  1676      C
  1677      C
  1678      C  ALWAYS HAVE 3 WEIGHTS AND 2 CUTS,   FIXED WING AND AFT SWEEP IN SAME
  1679      C     LOCATIONS    IF SWEPT, ALSO DO FWD  SWEEP.
  1680      C
  1681      C  SIDE OF FUSELAGE (FIXED OR AFT)
  1682           DO 135 N = 2,12
  1683           S(1) = YB11(N)-YSF
  1684           IF(S(1)) 115,115,110
  1685         110 S(10N) = S(10N) + WWT1(N)*S(1)
  1686         115 S(1) = YB12(N)-YSF
  1687           IF(S(1)) 125,125,120
  1688         120 S(105) = S(105) + WWT2(N)*S(1)
  1689         125 S(1) = YB13(N)-YSF
  1690           IF(S(1)) 132,132,130
  1691         130 S(106) = S(106) + WWT3(N)*S(1)
  1692         132 S(1) = YDW1(N) - YSF
  1693           IF(S(1)) 135,135,134
  1694         134 S(122) = S(122) + DW(N)*S(1)
  1695      C
  1696         135 CONTINUE
  1697      C HAVE TO CHECK FOR INBOARD AND/OR OUTBOARD NACELLES
  1698           IF(OD(17).NE.D(24)) GO TO 150
  1699           S(2) = D(24)
  1700           S(1) = S(118)-YSF
  1701           IF(S(1)) 141,141,140
  1702         140 S(2) = S(118)*S(1)
  1703         141 S(1) = S(121)-YSF
```

```
CARD NO  ****                          CONTENTS                          ****
1704            IF(S(1)) 140,140,,+6
1705       140  S(2) = S(2) + S(119)*S(1)
1706       145  S(104) = S(104) + S(2)
1707            S(105) = S(105) + S(2)
1708            S(106) = S(106) + S(2)
1709            S(122) = S(122) + S(2)
1710       C
1711       C  MIND OUTBOARD STATION  (FIXED OR AFT POSITION)
1712       C     SINE, COSINE AND TANGENT OF SWEEP OF THE ELASTIC AXIS IN AFT POSIT
1713       150  S(1) = DVH(3) *D(16)
1714            S(2) =SIN(S(1))
1715            S(3) =COS(S(1))
1716            S(4) =S(2)/S(3)
1717            DO 175 N = 2,12
1718            S(1) = YB11(N) - YEAI
1719            IF(S(1)) 155,155,153
1720       153  S(107) = S(107) + WHT1(N)*((XB11(N)-XEAI- S(1)*S(4))*S(2) +
1721           *                              S(1)/S(3) )
1722       155  S(1) = YB12(N) - YEAI
1723            IF(S(1)) 165,165,160
1724       160  S(108) = S(108) + WHT2(N)*( (XB12(N)-XEAI-S(1)*S(4))*S(2) +
1725           *                              S(1)/S(3) )
1726       165  S(1) = YB13(N) - YEAI
1727            IF(S(1)) 172,172,170
1728       170  S(109) = S(109) + WHT3(N)*( (XB13(N)-XEAI-S(1)*S(4))*S(2) +
1729           *                              S(1)/S(3) )
1730       172 S(1) = YDM1(N) - YEAI
1731            IF(S(1)) 175,175,174
1732       174 S(123) = S(123) + DWH(N)*( (XDM1(N)-XEAI-S(1)*S(4))*S(2) +
1733           *                             S(1)/S(3) )
1734       175  CONTINUE
1735       C
1736            IF(ID1(7).NE.D(24)) GO TO 200
1737            S(5)= D(24)
1738            S(1) = S(118)-YEAI
1739            IF(S(1))185,185,180
1740       180  S(5) =((S(117)-XEAI-S(1)*S(4))*S(2) + S(1)/S(3))*S(116)
1741       185  S(1) = S(121)-YEAI
1742            IF(S(1)) 195,195,190
1743       190  S(5) = S(5) + ((S(120)-XEAI-S(1)*S(4))*S(2) + S(1)/S(3))*S(119)
1744       195  S(107) = S(107)+S(5)
1745            S(108) = S(108)+S(5)
1746            S(109) = S(109)+S(5)
1747            S(123) = S(123)+S(5)
1748       C
1749       C  TEST IF CALCULATE FOR FORWARD SWEEP POSITION
1750       200  IF(ID1(2)) 300,300,202
1751       C                          FWD. SWEEP   SIDE OF FUSELAGE.
1752       202  DO 235 N=2,12
1753            S(1) = YB21(N) - YSF
1754            IF(S(1)) 215,215,210
1755       210  S(110) = S(110) + WHT1(N)*S(1)
1756       215  S(1) = YB22(N) - YSF
1757            IF(S(1)) 225,225,220
1758       220  S(111) = S(111) + WHT2(N)*S(1)
1759       225  S(1) = YB23(N) - YSF
1760            IF(S(1)) 232,232,230
1761       230  S(112) = S(112) + WHT3(N)*S(1)
1762       232 S(1) = YDM2(N) - YSF
1763            IF(S(1)) 235,235,234
1764       234 S(124) = S(124) + DWH(N)*S(1)
1765       235  CONTINUE
1766       C
1767            IF(ID1(7).NE.D(24)) GO TO 240
1768       C AT PRESENT NACELLE LOCATION IS CONSIDERED TO BE FIXED EVEN IF SWEPT
1769            S(110) = S(110) + S(2)
1770            S(111) = S(111) + S(2)
1771            S(112) = S(112) + S(2)
1772            S(124) = S(124) + S(2)
1773       C
1774       C             MIND OUTBOARD STATION (FORWARD SWEEP)
```

```
C  SINE, COSINE AND TANGENT OF SWEEP OF ELASTIC AXIS
 240  S(1) = DVA(23)*D(18)
      S(2) = SIN(S(1))
      S(3) = COS(S(1))
      S(4) = S(2)/S(3)
      DO 275 N=2,12
      S(1) = YB21(N)-YEA2
      IF(S(1)) 255,255,250
 250  S(113) = S(113) + WNT1(N)*( (XB21(N)-XEA2-S(1)*S(4))*S(2) +
     *                              S(1)/S(3) )
 255  S(1) = YB22(N)-YEA2
      IF(S(1)) 265,265,260
 260  S(114) = S(114) + WNT2(N)*( (XB22(N)-XEA2-S(1)*S(4))*S(2) +
     *                              S(1)/S(3) )
 265  S(1) = YB23(N)-YEA2
      IF(S(1)) 272,272,270
 270  S(115) = S(115) + WNT3(N)*( (XB23(N)-XEA2-S(1)*S(4))*S(2) +
     *                              S(1)/S(3) )
  272 S(1) = YDW2(N) - YEA2
      IF(S(1)) 275,275,274
  274 S(125) = S(125) + DWW(N)*( (XDW2(N)-XEA2-S(1)*S(4))*S(2) +
     *                             S(1)/S(3) )
 275  CONTINUE
C
      IF(DD(17).NE.D(24)) GO TO 300
      S(5) = D(24)
      S(1) = S(118)-YEA2
      IF(S(1))285,285,280
 280  S(5) = ((S(117)-XEA2-S(1)*S(4))*S(2) + S(1)/S(3)) *S(116)
 285  S(1) = S(121)-YEA2
      IF(S(1)) 295,295,290
 290  S(5) = S(5) + ((S(120)-XEA2-S(1)*S(4))*S(2) + S(1)/S(3)) * S(119)
 295  S(113) = S(113) + S(5)
      S(114) = S(114) + S(5)
      S(115) = S(115) + S(5)
      S(125) = S(125)+S(5)
C .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
C CHANGE SIGNS OF MOMENTS... DOWN BENDING IS SUPPOSED TO BE NEGATIVE.
C              AND ARE TO BE FOR ONE SIDE INSTEAD OF BOTH AS ARE NOW.
C
 300  DO 3001 N=1,12
      S(N+103) = -S(N+103)/D(2)
 3001 CONTINUE
      DO 3002 N=1,4
      S(N+121) = - S(N+121)/D(2)
 3002 CONTINUE
C
C .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
C
C MOMENTS THAT ARE CALCULATED FOR BASIC FLIGHT DESIGN WT   SAVED IN ---
C -----  FOR USE IN CALC. MAX STATIC BENDING MOM. FACTORS    ----------
      XMISC(43) = S(105)
      XMISC(44) = S(108)
      IF(DD(12)) 340,340,342
  340 DO 341 N=1,8
      XMISC(N+44) = S(N+103)
  341 CONTINUE
      GO TO 344
  342 DO 343 N=1,8
      XMISC(N+44) = S(N+100)
  343 CONTINUE
  344 CONTINUE
C     BC(100)   SERVICE LIFE
      BC(100) = DDD(7)
      XMISC(34) = DDD(7)
C      BC(170)  NO. OF LANDINGS IN LIFE
      BC(170) = DDD(8)
      BC(171) = YEA1
C      BC(187)  FATIGUE TAKE-OFF WT.
C        TEST FOR FATIGUE TAKE-OFF WEIGHT DIFFERENT FROM WDW
      IF(DDD(8)) 352,352,354
```

```
CARD NO     ****                       CONTENTS                      ****

              302 BC(107) = BC(1)
                  GO TO 306
              304 BC(107) = GDD(9)
            C      BC(108)  FATIGUE LANDING  WT.
            C        TEST FOR FATIGUE LANDING WEIGHT DIFFERENT FROM LDW
              306 IF(GDD(10)) 308,308,310
              308 BC(108) = BC(11)
                  GO TO 312
              310 BC(108) = GDD(10)
              312 BC(172) = BC(1)/BC(107)
                  BC(173) = BC(11)/BC(107)
            C        CALCULATE MINIMUM FLYING WEIGHT
                  S(1) = D(24)
                  DO 305 N=1,40
                  S(1) = S(1) + DWAT(N)
              305 CONTINUE
                  BC(174) = S(1)/BC(107)
                  BC(175) = S(104)
                  BC(176) = S(108)
                  BC(177) = S(122)
                  BC(178) = S(107)
                  BC(179) = S(109)
                  BC(180) = S(123)
                  BC(181) = S(110)
                  BC(182) = S(112)
                  BC(183) = S(124)
                  BC(184) = S(113)
                  BC(185) = S(115)
                  BC(186) = S(125)
                  RETURN
                  END
            C
            C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
            C                          SUBROUTINE GLNDGR
            C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
            C
                  SUBROUTINE GLNDGR
            C
            C       THIS ROUTINE SETS UP DESIGN DATA FOR USE  IN LANDING GEAR
            C       MODULE ROUTINE LANDGR
            C
                  COMMON TCOM(4320)
            C
                  DIMENSION D(700),OD(700),DV(2320),S(400),ND(200)
            C
                  DIMENSION GDD(30),DWAT(1000)
            C
                  EQUIVALENCE (D(1),TCOM(1)),(OD(1),TCOM(701)),(DV(1),TCOM(1401))
                 * , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
            C
                  EQUIVALENCE (GDD(1),OD(211)),(DWAT(1),DV(1121))
            C
            C      TRANSFER DATA TO LANDING GEAR IF XMISC(24) = TOGW = - OR 0
            C      NOTE, FIRST 118 CELLS IN TCOM NOW WILL HAVE LANDING GEAR DATA
            C
                  CALL READMS(1,D(1),118,25)
            C
                  D(46) = DWAT(944)
                  D(47) = DWAT(951)
                  D(48) = 0.0
                  D(49) = DWAT(946)
                  D(50) = DWAT(953)
                  D(51) = DWAT(947) - GDD(28)
                  D(52) = GDD(26)
                  D(53) = GDD(27)
                  D(54) = GDD(28)*2.0
                  D(72) = GDD(23)
                  D(73) = GDD(21)
                  D(81) = GDD(24)
                  D(82) = GDD(22)
                  D(80) = GDD(19)
```

```
CARD NO     ****                    CONTENTS                    ****

      D(90) = GDD(20)
C      LEVL CONVERTED TO FT/SEC
      D(92) = GDD(18)*1.2*1.6878
      D(91) = D(92)*(D(46)/D(47))**.5
C
C       ************************************************************
C
      CALL WRITMS(1,D(1),116,25)
C
C       ************************************************************
C
      RETURN
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                           SUBROUTINE DMAILD
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE DMAILD
C
C        THIS ROUTINE SETS UP INERTIA DATA IN MLD ARRAY FOR USE IN
C        AIRLOADS MODULE SURFACE DESIGN LOAD CALCULATION ROUTINE MAILDS
C
      COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
C
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
C
      DIMENSION GDI(20),GDD(30),DATS(40)
      DIMENSION DVM(50),DVH(30),DVV(30),DVWT(1000)
C
      DIMENSION YB10(12),XB10(12),YB20(12),XB20(12),WWT(12)
      DIMENSION WWT1(12),WWT2(12),WWT3(12),XB11(12),XB21(12),YB11(12)
     * , YB21(12),XB12(12),XB22(12),YB12(12),YB22(12),XB13(12)
     * , XB23(12),YB13(12),YB23(12)
C
      DIMENSION HWT(12),YBH(12),XBH(12),VWT(12),ZBV(12),XBV(12)
      DIMENSION YEAH(13),XEAH(13),ZEAV(13),XEAV(13)
     * , CBH(12),XLH(12),CBV(12),XLV(12)
C
      DIMENSION MLD(300)
C
      DIMENSION VE0(11),BME0(11),T20(11),VW1(11),BMW1(11),TW1(11)
     *, VW2(11),BMW2(11),TW2(11),VE1(11),BME1(11),T21(11)
     *, VI2(11),BMI2(11),TI2(11),VE2(11),BME2(11),T22(11)
     *, VE3(11),BME3(11),T23(11),VH(11),BMH(11),TH(11)
     *, VV(11),BMV(11),TV(11)
C
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401))
     * , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
C
      EQUIVALENCE (GDI(1),GD(1)),(GDD(1),GD(21)),(DATS(1),GD(46))
      EQUIVALENCE (DVM(1),DV(321)),(DVH(1),DV(371)),(DVV(1),DV(401))
     * , (DVWT(1),DV(1121))
C
      EQUIVALENCE (YB10(1),DVWT(401)),(XB20(1),DVWT(413))
     *, (YB20(1),DVWT(425)),(WWT(1),DVWT(473)),(XB10(1),DVWT(485))
C
      EQUIVALENCE            (WWT1(1),DVWT(497)), (WWT2(1),DVWT(521))
     *, (WWT3(1),DVWT(545)), (XB11(1),DVWT(569)), (XB21(1),DVWT(581))
     *, (YB11(1),DVWT(593)), (YB21(1),DVWT(605)), (XB12(1),DVWT(617))
     *, (XB22(1),DVWT(629)), (YB12(1),DVWT(641)), (YB22(1),DVWT(653))
     *, (XB13(1),DVWT(665)), (XB23(1),DVWT(677)), (YB13(1),DVWT(689))
     *, (YB23(1),DVWT(701))
C
      EQUIVALENCE (HWT(1),DVWT(872)),(YBH(1),DVWT(884))
     * , (XBH(1),DVWT(896)),(VWT(1),DVWT(908)),(ZBV(1),DVWT(920))
     * , (XBV(1),DVWT(932))
      EQUIVALENCE (YEAH(1),S(49)),(XEAH(1),S(62)),(ZEAV(1),S(75))
     * , (XEAV(1),S(88))
C
```

```
      EQUIVALENCE (MLD(1),S(10))
C
      EQUIVALENCE (VB0(1),MLD(4)),(BMB(1),MLD(15)),(TB0(1),MLD(26))
     *, (VAH(1),MLD(37)),(BMH(1),MLD(48)),(TMH(1),MLD(59))
     *, (VH2(1),MLD(70)),(BH2(1),MLD(81)),(TH2(1),MLD(92))
     *, (VB1(1),MLD(103)),(BM1(1),MLD(114)),(TB1(1),MLD(125))
     *, (VH1(1),MLD(136)),(BH12(1),MLD(147)),(T12(1),MLD(158))
     *, (VB2(1),MLD(169)),(BM2(1),MLD(180)),(TB2(1),MLD(191))
     *, (VB3(1),MLD(202)),(BM3(1),MLD(213)),(TB3(1),MLD(224))
     *, (VAV(1),MLD(235)),(BMV(1),MLD(246)),(TMV(1),MLD(257))
     *, (VV(1),MLD(268)),(BMV(1),MLD(279)),(TV(1),MLD(290))
C
      EQUIVALENCE (CBH(1),MLD(1)),(XLH(1),MLD(13))
     *, (CBV(1),MLD(25)),(XLV(1),MLD(37))
C
      DATA MF/1MF /, MZ/1MZ/
C
C       SETUP HORIZONTAL TAIL DATA
      S(7) = DVH(8)*D(8)
      YEAH(1) = B(24)
      XEAH(1) = DVH(4) + DVH(5)*DVH(22)
C
      DO 400 I=1,11
      YEAH(I+1) = DVH(I+18)
      XEAH(I+1) = XEAH(I) + YEAH(I+1)*TAN(DVH(3)*D(16))
      YBH(I) = (YEAH(I+1) + YEAH(I))/D(2)
      CBH(I) = DVH(5) - DVH(5)*D(1) - DVH(6)*YBH(I)/S(7)
      XLH(I) = DVH(4) + YBH(I)*TAN(DVH(2)*D(16))
  400 CONTINUE
C
      YEAH(13) = S(7)
      XEAH(13) = XEAH(1) + YEAH(13)*TAN(DVH(3)*D(16))
      YBH(12) = (YEAH(13) + YEAH(12))/D(2)
      CBH(12) = DVH(5) - DVH(5)*D(1) - DVH(6)*YBH(12)/S(7)
      XLH(12) = DVH(4) + YBH(12)*TAN(DVH(2)*D(16))
C
C       DISTRIBUTE HORIZONTAL TAIL AND CONTENTS BASED ON PARABOLIC DIST.
      S(1) = DWT(221) + DWT(222)
      S(2) = DWT(221)*DWT(231) + DWT(222)*DWT(232)
      S(5) = S(1)/(S(7)**D(20))
      S(8) = S(1)
      S(3) = B(24)
      S(4) = B(24)
C
      DO 420 I=1,12
      S(6) = S(8)
      IF(S(7) - YEAH(I+1)) 412,412,414
  412 S(8) = B(24)
      GO TO 416
  414 S(8) = S(5)*(S(7) - YEAH(I+1))**D(20)
  416 HWT(I) = S(8) - S(6)
      S(3) = S(3) + HWT(I)*XLH(I)
      S(4) = S(4) + HWT(I)*CBH(I)
  420 CONTINUE
C
      S(6) = (S(2) - S(3))/S(4)
C
      DO 430 I=1,12
      XBH(I) = XLH(I) + S(6)*CBH(I)
  430 CONTINUE
C
C       SETUP VERTICAL TAIL DATA
      S(7) = DVV(8)*D(12) - DVV(9)
      ZEAV(1) = B(24)
      XEAV(1) = DVV(3) + DVV(4)*DVV(21) + DVV(9)*TAN(DVV(2)*D(16))
C
      DO 450 I=1,11
      ZEAV(I+1) = DVV(I+9) - DVV(9)
      XEAV(I+1) = XEAV(1) + ZEAV(I+1)*TAN(DVV(2)*D(16))
      ZBV(I) = (ZEAV(I+1) + ZEAV(I))/D(2)
      CBV(I) = DVV(4) - DVV(4)*D(1) - DVV(5)*(ZBV(I) + DVV(9))/
```

```
CARD NO  ****                          CONTENTS                          ****
   2080          1 (S(7) + DVV(9))
   2080           XLV(I) = DVV(3) + (ZBV(I) + DVV(9))*TAN(DVV(I))*D(15))
   2081       450 CONTINUE
   2082       C
   2083           ZEAV(3) = S(7)
   2084           XEAV(3) = XEAV(1) + ZEAV(3)*TAN(DVV(2)*D(16))
   2085           ZBV(2) = (ZEAV(3) + ZEAV(2))/D(2)
   2086           CBV(2) = DVV(4) - DVV(5)*(D(1) - DVV(5))*(ZBV(2) + DVV(9))/
   2087          1 (S(7) + DVV(9))
   2088           XLV(2) = DVV(3) + (ZBV(2) + DVV(9))*TAN(DVV(I))*D(15))
   2089       C
   2090       C      DISTRIBUTE VERTICAL TAIL AND CONTENTS BASED ON PARABOLIC DIST
   2091           S(1) = DVAT(2,1) + DVAT(2,2)
   2092           S(2) = DVAT(2,1)*DVAT(2,1) + DVAT(2,2)*DVAT(2,2)
   2093       C
   2094       C      TEST FOR TWO VERTICALS
   2095           IF(D(1) - DVV(23)) 452,454,454
   2096       452 S(1) = S(1)/D(2)
   2097           S(2) = S(2)/D(2)
   2098       454 S(5) = S(1)/(S(7)*D(20))
   2099           S(9) = S(1)
   2100           S(3) = D(24)
   2101           S(4) = D(24)
   2102       C
   2103           DO 470 I=1,12
   2104           S(8) = S(9)
   2105           IF(S(7) - ZEAV(I+1)) 462,462,464
   2106       462 S(9) = D(24)
   2107           GO TO 466
   2108       464 S(9) = S(5)*(S(7) - ZEAV(I+1))**D(20)
   2109       466 VAT(I) = S(8) - S(9)
   2110           S(3) = S(3) + VAT(I)*XLV(I)
   2111           S(4) = S(4) + VAT(I)*CBV(I)
   2112       470 CONTINUE
   2113       C
   2114           S(6) = (S(2) - S(3))/S(4)
   2115       C
   2116           DO 480 I=1,12
   2117           XBV(I) = XLV(I) + S(6)*CBV(I)
   2118       480 CONTINUE
   2119       C
   2120       C      CLEAR LOADS DATA REGION
   2121       C
   2122           DO 500 I=1,300
   2123           XLD(I) = D(24)
   2124       500 CONTINUE
   2125       C
   2126       C      CALCULATE DEAD WEIGHT LOADS - HORIZONTAL TAIL IB NET
   2127           S(1) = DVH(3)*D(18)
   2128           S(2) = SIN(S(1))
   2129           S(3) = COS(S(1))
   2130           S(4) = S(2)/S(3)
   2131       C
   2132           DO 520 I=1,11
   2133           K = I + 1
   2134           DO 510 J=K,12
   2135           S(9) = YBH(J) - YEAH(I+1)
   2136           S(8) = XBH(J) - XEAH(I+1)
   2137           YBH(I) = YBH(I) - HAT(J)/D(2)
   2138           BMH(I) = BMH(I) - HAT(J)*((S(8) - S(5)*S(4))*S(2) + S(9)/S(3))/
   2139          1 D(2)
   2140           TH(I) = TH(I) + HAT(J)*(S(8) - S(5)*S(4))*S(3)/D(2)
   2141       510 CONTINUE
   2142       520 CONTINUE
   2143       C
   2144       C      CALCULATE DEAD WEIGHT LOADS - VERTICAL TAIL IB NET
   2145           S(1) = DVV(2)*D(16)
   2146           S(2) = SIN(S(1))
   2147           S(3) = COS(S(1))
   2148           S(4) = S(2)/S(3)
   2149       C
```

```
2130            DO 970 I=1,11
2131            K = I + 1
2132            DO 980 J=K,12
2133            S(5) = ZBV(J) - ZEAV(I+1)
2134            S(6) = XBV(J) - XEAV(I+1)
2135            VA(I) = VV(I) - VAT(J)
2136            BVV(I) = BVV(I) - VAT(J)*((S(6) - S(5)*S(4))*S(2) + S(5)/S(3))
2137            TV(I) = TV(I) + VAT(J)*(S(6) - S(5)*S(4))*S(3)
2138          980 CONTINUE
2139          970 CONTINUE
2140        C
2141            MLD(1) = DMAT(9,4)
2142            MLD(2) = GDD(11)
2143            MLD(3) = GDD(13)
2144        C
2145        C       CALCULATE LOADS DUE TO WING WEIGHT
2146            S(1) = DWW(3)*D(18)
2147            S(2) = SIN(S(1))
2148            S(3) = COS(S(1))
2149            S(4) = S(2)/S(3)
2150            IF(GDD(12)) 605,605,610
2151          605 S(11) = S(1)
2152            S(12) = S(2)
2153            S(13) = S(3)
2154            S(14) = S(4)
2155            GO TO 620
2156          610 S(11) = DWW(23)*D(18)
2157            S(12) = SIN(S(11))
2158            S(13) = COS(S(11))
2159            S(14) = S(12)/S(13)
2160        C
2161        C
2162          620 DO 950 I=1,11
2163            S(8) = DWW(4) + DWW(5)*DWW(21) + DWW(I+8)*S(4)
2164            S(7) = DWW(I+8)
2165            IF(GDD(12)) 625,625,630
2166        C
2167          625 S(17) = S(7)
2168            S(18) = S(8)
2169            GO TO 632
2170          630 S(17) = DWW(I+20)
2171            S(18) = DWW(24) + DWW(25)*DWW(14) + DWW(I+28)*S(14)
2172          632 K = 1
2173          634 IF(YB18(K) - S(7)) 635,640,640
2174          635 K = K + 1
2175            IF(K - 12) 634,634,700
2176        C
2177          640 DO 645 J=K,12
2178            S(5) = YB18(J) - S(7)
2179            S(6) = XB18(J) - S(8)
2180            S(15) = YB20(J) - S(17)
2181            S(16) = XB20(J) - S(18)
2182            VA(I) = VA(I) - WAT(J)/D(2)
2183            BVV(I) = BVV(I) - WAT(J)*((S(6) - S(5)*S(4))*S(2) +
2184           1 S(5)/S(3))/D(2)
2185            TV(I) = TV(I) + WAT(J)*(S(6) - S(5)*S(4))*S(3)/D(2)
2186            VA2(I) = VA2(I) - WAT(J)/D(2)
2187            BVA2(I) = BVA2(I) - WAT(J)*((S(16) - S(15)*S(14))*S(12) +
2188           1 S(15)/S(13))/D(2)
2189            TV2(I) = TV2(I) + WAT(J)*(S(16) - S(15)*S(14))*S(13)/D(2)
2190          645 CONTINUE
2191        C
2192          650 CONTINUE
2193        C
2194        C
2195        C       CALCULATE LOADS DUE TO WING AND CONTENT INERTIA
2196          700 DO 800 I=1,11
2197            S(7) = DWW(I+8)
2198            S(8) = DWW(4) + DWW(5)*DWW(21) + DWW(I+8)*S(4)
2199            IF(GDD(21)) 705,705,710
2200          705 S(17) = S(7)
```

```
CARD NO ****                                CONTENTS                                ****
 2001          S(10) = S(8)
 2002          GO TO 712
 2003        710 S(17) = DVAR(I+20)
 2004          S(18) = DVAR(24) + DVAR(25)*DVAR(1) + DVAR(I+20)*S(14)
 2005        712 K = 1
 2006        714 IF(YB12(K) - S(7)) 716,718,718
 2007        716 K = K + 1
 2008          IF(K - 12) 714,714,730
 2009        C
 2010        718 DO 720 J=K,12
 2011          S(9) = YB12(J) - S(7)
 2012          S(8) = XB12(J) - S(8)
 2013          V12(I) = V12(I) - WMT2(J)/D(2)
 2014          BM12(I) = BM12(I) - WMT2(J)*((S(8) - S(5)*S(14))*S(2) +
 2015         1 S(9)/S(13))/D(2)
 2016          T12(I) = T12(I) + WMT2(J)*(S(8) - S(9)*S(14))*S(3)/D(2)
 2017          S(9) = YB21(J) - S(7)
 2018          S(8) = XB21(J) - S(8)
 2019          V21(I) = V21(I) - WMT1(J)/D(2)
 2020          BM21(I) = BM21(I) - WMT1(J)*((S(8) - S(5)*S(14))*S(2) +
 2021         1 S(9)/S(13))/D(2)
 2022          T21(I) = T21(I) + WMT1(J)*(S(8) - S(9)*S(14))*S(3)/D(2)
 2023          S(9) = YB22(J) - S(7)
 2024          S(8) = XB22(J) - S(8)
 2025          V22(I) = V22(I) - WMT2(J)/D(2)
 2026          BM22(I) = BM22(I) - WMT2(J)*((S(8) - S(5)*S(14))*S(2) +
 2027         1 S(9)/S(13))/D(2)
 2028          T22(I) = T22(I) + WMT2(J)*(S(8) - S(9)*S(14))*S(3)/D(2)
 2029          S(9) = YB23(J) - S(7)
 2030          S(8) = XB23(J) - S(8)
 2031          V23(I) = V23(I) - WMT3(J)/D(2)
 2032          BM23(I) = BM23(I) - WMT3(J)*((S(8) - S(5)*S(14))*S(2) +
 2033         1 S(9)/S(13))/D(2)
 2034          T23(I) = T23(I) + WMT3(J)*(S(8) - S(9)*S(14))*S(3)/D(2)
 2035        720 CONTINUE
 2036        C
 2037        C       TEST ON NACELLE
 2038        730 IF(QD(17) .NE. D(24)) GO TO 800
 2039            II = 1
 2040            S(5) = DATS(19) - S(7)
 2041            IF(S(5)) 770,770,752
 2042        752 S(8) = DVAR(975) - S(8)
 2043            S(15) = DATS(14) - S(7)
 2044            S(16) = DVAR(975) - S(10)
 2045            S(9) = DVAR(974)*((S(8) - S(5)*S(14))*S(2) + S(5)/S(13))/D(2)
 2046            S(10) = DVAR(974)*(S(8) - S(5)*S(14))*S(3)/D(2)
 2047            S(19) = DVAR(974)*((S(16) - S(15)*S(14))*S(12) + S(15)/S(13))/D(2)
 2048            S(20) = DVAR(974)*(S(16) - S(15)*S(14))*S(13)/D(2)
 2049            S(21) = DVAR(974)/D(2)
 2050        760 V12(I) = V12(I) - S(21)
 2051            BM12(I) = BM12(I) - S(9)
 2052            T12(I) = T12(I) + S(10)
 2053            V21(I) = V21(I) - S(21)
 2054            BM21(I) = BM21(I) - S(19)
 2055            T21(I) = T21(I) + S(20)
 2056            V22(I) = V22(I) - S(21)
 2057            BM22(I) = BM22(I) - S(19)
 2058            T22(I) = T22(I) + S(20)
 2059            V23(I) = V23(I) - S(21)
 2060            BM23(I) = BM23(I) - S(19)
 2061            T23(I) = T23(I) + S(20)
 2062            IF(II - 1) 770,770,800
 2063        770 II = 2
 2064            S(5) = DATS(17) - S(7)
 2065            IF(S(5)) 800,800,772
 2066        772 S(8) = DVAR(980) - S(8)
 2067            S(15) = DATS(17) - S(7)
 2068            S(16) = DVAR(980) - S(10)
 2069            S(9) = DVAR(979)*((S(8) - S(5)*S(14))*S(2) + S(5)/S(13))/D(2)
 2070            S(10) = DVAR(979)*(S(8) - S(5)*S(14))*S(3)/D(2)
 2071            S(19) = DVAR(979)*((S(16) - S(15)*S(14))*S(12) + S(15)/S(13))/D(2)
```

```
CARD NO  ****                          CONTENTS                          ****
0272          S(20) = DWAT(979)*(S(18) - S(15)*S(14))*S(13)/D(2)
0273          S(21) = DWAT(979)/D(2)
0274          GO TO 750
0275        800 CONTINUE
0276      C
0277      C       GO TAXI LOADS - 20
0278      C
0279          DO 810 I=1,11
0280          VEG(I) = VE1(I)*D(2)
0281          BMEG(I) = BME1(I)*D(2)
0282          TZG(I) = TZ1(I)*D(2)
0283        810 CONTINUE
0284      C
0285          IF(GDI(3)) 900,900,820
0286      C
0287      C       LANDING GEAR ON WING
0288        820 S(21) = DWAT(944)*(DWAT(946) - GDD(27))/(GDD(26) - GDD(27))
0289      C
0290          DO 830 I=1,11
0291          IF(GDI(21)) 812,812,814
0292        812 S(17) = DWA(I+9)
0293          S(18) = DWA(4) + DWA(5)*DWA(21) + DWA(I+9)*S(14)
0294          GO TO 816
0295        814 S(17) = DWA(I+29)
0296          S(18) = DWA(24) + DWA(25)*DWA(41) + DWA(I+29)*S(14)
0297        816 S(15) = GDD(29) - S(17)
0298          IF(S(15)) 900,900,818
0299        818 S(16) = GDD(26) - S(
0300          S(19) = S(21)*(S(18) - S(15)*S(14))*S(12) + S(15)/S(13))
0301          S(20) = S(21)*(S(18) - S(15)*S(14))*S(13)
0302          VEG(I) = VEG(I) + S(21)
0303          BMEG(I) = BMEG(I) + S(19)
0304          TZG(I) = TZG(I) - S(20)
0305        830 CONTINUE
0306      C
0307        900 CONTINUE
0308      C
0309          IF(IP(47)) 9501,9501,9502
0310       9501 CONTINUE
0311      C     FOR WING FORWARD STATIONS TEST BC(78)   IF 0 USE AFT STATIONS
0312          IF(GDI(2)) 910,910,915
0313        910 DO 911 I=1,11
0314        911 S(I) = DWA(I+8)
0315          GO TO 920
0316        915 DO 916 I=1,11
0317        916 S(I) = DWA(I+29)
0318      C
0319        920 WRITE(6,921)
0320        921 FORMAT(1H1,40X,24HSHEAR, MOMENT AND TORQUE,24X,
0321           1  21H** DWALD - IP(47) **,              //  19X,15HWING ONLY AT
0322           *18, 33X, 23HWING AND CONTENTS AT 10//  20X, 12HAFT POSITION, 33X,
0323           * 27HAFT POSITION GROSS WEIGHT 2 )
0324            WRITE(6,923)
0325        923 FORMAT(4H0  I, 5X,5HSHEAR,6X,6HMOMENT,6X,6HTORQUE,6X,10HBUTT PLANE
0326           *11X,5HSHEAR,6X,6HMOMENT,6X,6HTORQUE, 9X,1HI /)
0327            WRITE(6,925) (I,VW(I),BMW(I),TW(I),DWA(I+9),VT2(I),BMT2(I),
0328           *TT2(I), I, I=1,11 )
0329        925 FORMAT(1I4,3F12.0,6X,1F8.1,6X,3F12.0,5X,113 )
0330      C
0331            WRITE(6,931)
0332        931 FORMAT(//1H0, 19X,15HWING ONLY AT 10,33X,23HWING AND CONTENTS AT 10
0333           *//19X,18HFORWARD POSITION, 28X, 3HFORWARD POSITION AT GROSS WEIGH
0334           *T 1 )
0335            WRITE(6,923)
0336            WRITE(6,929) (I,VW(I),BMW(I),TW(I),S(I),VE1(I),BME1(I),
0337           * TZ1(I), I, I=1,11 )
0338      C
0339            WRITE(6,938)
0340        938 FORMAT(1H1,68X,21H** DWALD - IP(47) **/
0341           1           19X, 23HWING AND CONTENTS AT 10, 28X, 23HWING AND CONTE
0342           *NTS AT 10 // 8X, 3HFORWARD POSITION AT GROSS WEIGHT 2, 18X,
```

```
CARD NO  ****                          CONTENTS                          ****
2343          * 3HAFTWARD POSITION AT GROSS WEIGHT 3 )
2344           WRITE(6,923)
2345           WRITE(6,875) (I,V82(I),BV82(I),T82(I),S(I),V23(I),BV23(I),
2346          * T23(I),I, I=1,11 )
2347           WRITE(6,937)
2348      937 FORMAT(/1H0,40X, 25HJET WING LOADS AT 20 TAXI// 35X, 3HFORWARD PO
2349         *SITION AT GROSS WEIGHT I // 20X, 1HI,5X,5HSHEAR,6X,6HMOMENT, 6X,
2350          * 6HTORQUE, 6X,10HBUTT PLANE / )
2351           WRITE(6,938) (I,V20(I),BV20(I),T20(I),S(I),I=1,11)
2352      938 FORMAT(35X, I1, 3F12.0, 6X, F8.1 )
2353      C
2354           WRITE(6,940)
2355      940 FORMAT(1H1, 38X, 28HHORIZONTAL TAIL AND CONTENTS ,
2356         1  35X,21H** DMAXLD - IP(47) **///)
2357           WRITE(6,941) HY,HY
2358      941 FORMAT(7X,19HCOORD. OF E. A., 9X,7HSECTION, 9X, 11HCOORDINATES /
2359          * 11X, 1A1, 8X, 1HX, 9X, 5HSHEAR, 8X, 6HMOMENT, 8X, 6HTORQUE, 14X,
2360          * 6HWEIGHT, 9X, 1A1, 4H BAR, 9X, 5HX BAR /)
2361           WRITE(6,943) YEAH(1), XEAH(1)
2362      943 FORMAT( 4X,2F10.2)
2363           WRITE(6,945) HWT(1),YBH(1),XBH(1),YEAH(1+1),XEAH(1+1),VH(1),
2364          * BMH(1),TH(1), I=1,11 )
2365      945 FORMAT(70X,1F14.2,2F10.2/ 4X, 2F10.2, 3F14.0 )
2366           WRITE(6,945) HWT(12),YBH(12),XBH(12),YEAH(13),XEAH(13)
2367      C
2368           WRITE(6,951)
2369      951 FORMAT(1H1,39X, 26HVERTICAL TAIL AND CONTENTS ,
2370         1  23X,21H** DMAXLD - IP(47) **///)
2371           WRITE(6,941) HZ,HZ
2372           WRITE(6,943) ZEAV(1), XEAV(1)
2373           WRITE(6,945) (VWT(I),ZBV(I),XBV(I),ZEAV(I+1),XEAV(I+1),VV(I),
2374          * BMV(I),TV(I), I=1,11 )
2375           WRITE(6,945) VWT(12),ZBV(12),XBV(12),ZEAV(13),XEAV(13)
2376      9002 CONTINUE
2377      C
2378      C       ***************************************************************
2379      C
2380           CALL MRITMS(1,MLD(1),300,10)
2381      C
2382      C       ***************************************************************
2383      C
2384           RETURN
2385           END
2386      C
2387      C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
2388      C                              SUBROUTINE DSDMPR
2389      C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
2390      C
2391           SUBROUTINE DSDMPR
2392      C
2393            COMMON TCOM(4320)
2394           COMMON /MISC/ XMISC(100)
2395           COMMON /IPRINT/ IP(100)
2396           DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
2397           DIMENSION EQU(200)
2398           DIMENSION DATS(40)
2399           DIMENSION TITLE(16)
2400           DIMENSION ALT(10),VH(10),VL(10),TEMH(10),TEML(10),PTH(10),
2401          1 PTL(10),PBL(10),RIH(10),RIL(10),R2H(10),R2L(10),R3H(10),
2402          * R3L(10),PWTH(10),PHEH(10),PWTL(10),PWEL(10),PST(10),RATL(10)
2403           DIMENSION GH(10),GL(10),PO(10)
2404           DIMENSION DSP(10)
2405           EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
2406          1 (S(1),TCOM(372)),(ND(1),TCOM(412))
2407           EQUIVALENCE (EQU(1),D(81))
2408           EQUIVALENCE (DATS(1),GD(481))
2409           EQUIVALENCE (EQTP,DATS(2))
2410           EQUIVALENCE (TITLE(1),XMISC(85))
2411           EQUIVALENCE (GH(1),DV(81)),(GL(1),DV(91)),
2412          1 (PO(1),DV(21))
2413           EQUIVALENCE (ALT(1),DV(1)),(VH(1),DV(11)),(VL(1),DV(21)),
```

```
CARD NO  ****                          CONTENTS                          ****
2N19          1 (TEXH(I),DV(14)),(TEPL(I),DV(15)),(PTH(I),DV(16)),
2N15          2 (PTL(I),DV(17)),(PBL(I),DV(18)),(RIH(I),DV(20)),
2N16          3 (RIL(I),DV(21)),(RBH(I),DV(22)),(RBL(I),DV(23)),
2N17          4 (R3H(I),DV(24)),(R3L(I),DV(25)),(PWTH(I),DV(26)),
2N18          5 (PWEH(I),DV(27)),(PWTL(I),DV(28)),(PWEL(I),DV(29)),
2N19          6 (PST(I),DV(30)),(RATL(I),DV(13))
2N20          EQUIVALENCE (DSP(1),DV(3))
2N21          EQUIVALENCE (ND(101),I),(ND(102),J)
2N22          EQUIVALENCE (ND(112),IVG)
2N23     C
2N24          DO 200 I=1,9
2N25          RIH(I) = EQU(32) - EQU(33)*VH(I)
2N26          RIL(I) = EQU(32) - EQU(33)*VL(I)
2N27     C      SETUP CONSTANTS FOR ENGINE FACE HAMMERSHOCK PRESSURE RATIO
2N28          S(1) = TEXH(I)/EQU(34)
2N29          S(2) = EQU(34)/TEXH(I)
2N30          S(3) = TEPL(I)/EQU(34)
2N31          S(4) = EQU(34)/TEPL(I)
2N32     C      GO TO PROPER CURVE FIT
2N33          IF(EGTP) 10,10,20
2N34     C      TURBOJET
2N35       10 RBH(I) = EQU(35) - EQU(36)*S(1) + EQU(37)*S(2) - EQU(38)*S(2)**2
2N36          RBL(I) = EQU(35) - EQU(36)*S(3) + EQU(37)*S(4) - EQU(38)*S(4)**2
2N37          GO TO 50
2N38     C      FANJET
2N39       20 IF(EGTP - EQU(39)) 22,22,30
2N40     C      FANJET BY PASS RATIO LESS THAN OR EQUAL TO 1.5
2N41       22 RBH(I) = -EQU(40) + EQU(41)*S(1) + EQU(42)*S(2) - EQU(43)*S(2)**2
2N42          RBL(I) = -EQU(40) + EQU(41)*S(3) + EQU(42)*S(4) - EQU(43)*S(4)**2
2N43     C      CHECK FOR APPLICABILITY OF CURVE
2N44          IF(TEPL(I) - EQU(44)) 50,50,24
2N45       24 WRITE(6,60) EGTP,TEPL(I),EQU(44)
2N46       60 FORMAT(1H1,20X,23H*** WARNING MESSAGE ***/10X,
2N47         1HBRAM TEMPERATURE EXCEEDED FOR FANJET BPR =,F5.1/10X,
2N48         2HBRAM TEMP =,F8.2,7HLIMIT =,F8.2)
2N49          GO TO 50
2N50       30 IF(EGTP - EQU(45)) 32,32,40
2N51     C      FANJET BPR G.T. 1.5 BUT L.T. OR.E.T. 2.5
2N52       32 RBH(I) = -EQU(46) + EQU(47)*S(1) + EQU(48)*S(2) - EQU(49)*S(2)**2
2N53          RBL(I) = -EQU(46) + EQU(47)*S(3) + EQU(48)*S(4) - EQU(49)*S(4)**2
2N54          IF(TEPL(I) - EQU(50)) 50,50,34
2N55       34 WRITE(6,60) EGTP,TEPL(I),EQU(50)
2N56          GO TO 50
2N57     C      FANJET BY PASS RATIO GREATER THAN 2.5     BOTH FACE AND THROAT
2N58       40 RBH(I) = EQU(51) - EQU(52)*S(1) + EQU(53)*S(2) - EQU(54)*S(2)**2
2N59          RBL(I) = EQU(51) - EQU(52)*S(3) + EQU(53)*S(4) - EQU(54)*S(4)**2
2N60          R3H(I) = RBH(I)
2N61          R3L(I) = RBL(I)
2N62          IF(TEPL(I) - EQU(55)) 200,200,42
2N63       42 WRITE(6,60) EGTP,TEPL(I),EQU(55)
2N64          GO TO 200
2N65     C      GO THROAT HAMMERSHOCK PRESSURE RATIOS
2N66       50 IF(IVG - 2) 52,52,60
2N67     C      FIXED GEOMETRY INLET                              IVG=1 OR 2
2N68       52 IF(VL(I) - EQU(61)) 60,60,58
2N69       58 WRITE(6,61) EGTP,IVG,VL(I),EQU(61)
2N70       61 FORMAT(1H1,20X,23H*** WARNING MESSAGE ***/10X,
2N71         1HBMSPEED EXCEEDED FOR ENGINE INLET COMBINATION/20X,
2N72         2HBMSPR =,F5.1,3X,12HINLET TYPE =,I3,3X,7HSPEED =,F5.2,3X,
2N73         3HBLIMIT SPEED =,F5.2)
2N74       60  CONTINUE
2N75     C                FOR BY PASS RATIO  LESS THAN OR EQUAL TO 2.5
2N76     C             RATIO ENGIN FACE FOR THROAT              (4-21-72)
2N77          RBH(I) = (EQU(62) - EQU(63)*VH(I) - EQU(64)*VH(I)**2) * RBH(I)
2N78          RBL(I) = (EQU(62) - EQU(63)*VL(I) - EQU(64)*VL(I)**2) * RBL(I)
2N79     C
2N80     C          FANJET BYPASS RATIO GREATER THAN 2.5  THROAT PRESSURE RATIO
2N81     C             SAME AS ENGINE FACE - - - -      SET-UP THERE.
2N82     C
2N83       200 CONTINUE
2N84     C
```

CARD NO **** CONTENTS ****

```
2488            DO  210 I=1,9
2485                PHTH(I) = PTH(I)*R3H(I)
2487                PHEH(I) = PTH(I)*R2H(I)
2488                PHTL(I) = PTL(I)*R3L(I)
2489                PHEL(I) = PTL(I)*R2L(I)
2490            PST(I) = PTL(I)/RATL(I)*RIL(I)
2491          210 CONTINUE
2492        C       TEST FOR MAXIMUM VALUE OF GL
2493            DSP(5) = GL(1)
2494            J=1
2495            DO 300 I=2,9
2496            IF(DSP(5) - GL(I)) 301,300,300
2497          301 DSP(5) = GL(I)
2498            J = I
2499          300 CONTINUE
2500            DSP(1) = ALT(J)
2501            DSP(2) = VH(J)
2502            DSP(3) = VL(J)
2503            DSP(4) = GH(J)
2504        C       TEST FOR MAXIMUM DUCT PRESSURES
2505            DSP(6) = PHTH(1)
2506            DO 304 I=2,9
2507            IF(DSP(6) - PHTH(I)) 302,304,304
2508          302 DSP(6) = PHTH(I)
2509          304 CONTINUE
2510            DSP(7) = PHTL(1)
2511            DO 308 I=2,9
2512            IF(DSP(7) - PHTL(I)) 306,308,308
2513          306 DSP(7) = PHTL(I)
2514          308 CONTINUE
2515        C       TEST FOR MAXIMUM STATIC PRESSURE
2516            S(5) = PST(1) - PO(1)/D(17)
2517            DSP(8) = S(5)
2518            DO 310 I=2,9
2519            S(9) = PST(I) - PO(I)/D(17)
2520            IF(DSP(8) - S(5)) 309,310,310
2521          309 DSP(8) = S(5)
2522          310 CONTINUE
2523        C       TEST FOR HAMMERSHOCK AT VH
2524            S(5) = PHEH(1) - PO(1)/D(17)
2525            DSP(9) = S(5)
2526            DO 312 I=2,9
2527            S(5) = PHEH(I) - PO(I)/D(17)
2528            IF(DSP(9) - S(5)) 311,312,312
2529          311 DSP(9) = S(5)
2530          312 CONTINUE
2531        C       TEST FOR HAMMERSHOCK AT VL
2532            S(5) = PHEL(1) - PO(1)/D(17)
2533            DSP(10) = S(5)
2534            DO 316 I=2,9
2535            S(5) = PHEL(I) - PO(I)/D(17)
2536            IF(DSP(10) - S(5)) 315,316,316
2537          315 DSP(10) = S(5)
2538          316 CONTINUE
2539        C
2540            IF(IP(43))5011,5011,5012
2541         5011 CONTINUE
2542              WRITE(6,217)(TITLE(N),N=1,18)
2543          217 FORMAT(1H1,88X,21H** DSGNPR - IP(43) **//10X,8A10/10X,8A10)
2544              WRITE(6,218)        EGTP, FVG
2545          218 FORMAT(1H0, 38X, 30HSPEED PROFILE DESIGN CONSTANTS
2546             *     / 1H0, 88X, 14HBYPASS RATIO =, 1F8.2, 20X, 9HFVG =, 1I2 /
2547             2 1H0, 44X, 7HTEMP(R), 8X, 8HSTATIC(R), 12X, 19HHAMMERSHOCK (R) /
2548             3 16X, 5HALT, 13X, 2HVH, 8X, 11HDEG RANKINE, 8X, 11HPRES. RATIO,
2549             4 8X, 4HFACE, 11X, 8HTHROAT )
2550              WRITE(6,221)(ALT(I), VH(I),TEMH(I),RIH(I),R2H(I),R3H(I),I=1,9)
2551          221 FORMAT( 8X, 1F13.1, 1F14.2, 1F17.3, 3F18.4 )
2552              WRITE(6,222)
2553          222 FORMAT(1H0,44X,7HTEMP(L),8X,8HSTATIC(L),12X,19HHAMMERSHOCK (L) /
2554             1 16X, 5HALT, 13X, 2HVL, 8X, 11HDEG RANKINE, 8X, 11HPRES. RATIO,
2555             2 8X, 4HFACE, 11X, 8HTHROAT )
```

CARD NO **** CONTENTS ****

```
      WRITE(6,821)((ALT(I),VL(I),TEPL(I),RIL(I),RBL(I),RSL(I)),I=1,9)
      WRITE(6,823)
  823 FORMAT(1HO, 28X, 7HPRES(H), 9X, 7HPRES(H), 9X, 7HPRES(L), 9X,
     1 7HPRES(L), 10X, 5HSTATIC / 27X, 11HTHROAT-PSIA, 5X,
     2 11HENGINE-PSIA, 5X, 11HTHROAT-PSIA, 5X, 11HENGINE-PSIA, 6X,
     3 11HPRES THROAT )
      WRITE(6,824)((ALT(I),PHTH(I),PHEH(I),PHTL(I),PHEL(I),PST(I)),
     1 I=1,9)
  824 FORMAT( 8X, 1F13.1, 1F15.3, 4F16.3)
 9812 CONTINUE
  999 RETURN
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                          SUBROUTINE DSTNOR
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE DSTNOR
C     DISTRIBUTE POINT WEIGHT TO 2 OR 3 FUS. STA. ACCORDING TO SPACING
C             JULY 1972
C             J IS 1ST  AND K IS LAST, WEIGHT IN S(J+20)TO S(K+20)
      COMMON TCOM(4320)
      DIMENSION D(700),        DV(2320),S(400),ND(200)
     *, XBAR(20), DELX(20), DVB(440)
      EQUIVALENCE (TCOM(1),D(1)), (TCOM(1401),DV(1))
     *, (TCOM(3721),S(1)), (TCOM(4121),ND(1))
     *,(DV(431),DVB(1))
     *,(DVB(141),XBAR(1)), (DVB(161),DELX(1))
     *,(ND(182),J), (ND(183),K), (ND(115),NC)
      WT = S(1)
      CG = S(2)
C         TEST  IF FORWARD OF 1ST STA. PRINT ERROR AND DISTR. TO 1 + 2
      IF(S(2)-XBAR(1)) 10,20,20
   10 WRITE(6,15) S(1),S(2),XBAR(1)
   15 FORMAT(1H1,23H***** IN DSTNOR, WEIGHT,1E13.6,15H LB. LOCATED AT,
     * 1E13.6,18H IS FWD OF 1ST STA.,1E13.6, 6H ***** )
      J=1
      K=2
      GO TO 99
C         TEST IF AFTER LAST STA.
   20 IF(S(2) - XBAR(NC+1)) 50,50,25
   25 K = NC+1
      J=NC
      WRITE(6,30) S(1), S(2), XBAR(K)
   30 FORMAT(1H1,23H***** IN DSTNOR, WEIGHT,1E13.6,15H LB. LOCATED AT,
     * 1E13.6,20H IS AFT OF LAST STA.,1E13.6, 6H ***** )
      GO TO 99
C GK  IN LIMITS SO SEARCH FOR HIGHER STA.
   50 N=2
   51 IF(S(2) -XBAR(N)) 60,55,55
   55 N=N+1
      GO TO 51
   60 K=N
      J=N-1
      IF(DELX(K)-D(4)) 62,62,65
   62 IF(K - (NC+1)) 63,99,99
   63 N = K
      K=K+1
      GO TO 70
   65 IF(DELX(J)-D(4)) 67,67,99
   67 IF(J - 1) 99,99,68
   68 N = J
      J=J-1
   70 S(N+20) = S(1)*DELX(N)/(DELX(J)+DELX(N)+DELX(K))
C         NEW CG FOR REMAINING LOAD AFTER SMALL SECTION TAKEN OUT
      CG  = (S(1)*S(2)-S(N+20)*XBAR(N))/(S(1)-S(N+20))
      WT  = S(1) - S(N+20)
   80 S(K+20) =  WT *( CG -XBAR(J))/(XBAR(K)-XBAR(J))
      S(J+20) =  WT  - S(K+20)
  999 CONTINUE
      RETURN
```

```
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                              SUBROUTINE DSTTRI
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE DSTTRI
C       WRITTEN 5 JULY 1972
C       TO DISTRIBUTE WEIGHT ACCORDING TO A TRIANGULAR DISTRIBUTION
C       COUNTERS I = APEX CUT, K = LASTCUT, L = FIRST CUT
C       S(201) = FRONT END OF TRIANGLE
C       S(202) = FORWARD PART OF BASE
C       S(203) = TOTAL LENGTH OF BASE
C       S(214) = C.G. OF WEIGHT TO BE DISTRIBUTED
C       S(215) = INPUT WEIGHT
      COMMON TCOM(4320)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
      DIMENSION XO(20),XBAR(20),DELX(20)
      DIMENSION GDB(80),DVB(440)
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
      EQUIVALENCE (GDB(1),GD(381)),(DVB(1),DV(431))
      EQUIVALENCE (XO(1),GDB(56)),(XBAR(1),DVB(141)),(DELX(1),DVB(161))
      EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104))
      DO 4 J=1,20
      S(J+220) = D(24)
    4 CONTINUE
      S(204) = D(24)
      S(205) = D(24)
      S(206) = D(24)
      S(207) = D(24)
      DO 20 J=L,I
      S(J+240) = (XBAR(J) - S(201))/S(202)*DELX(J)
      S(204) = S(204) + S(J+240)
      S(205) = S(205) + S(J+240)*XBAR(J)
   20 CONTINUE
      I = I + 1
C           AFT PART OF BASE
      S(216) = S(203)-S(202)
C           AFT END OF BASE
      S(217) = S(201) + S(203)
      DO 30 J=I,K
      S(J+240) = (S(217)- XBAR(J))/S(216)*DELX(J)
      S(204) = S(204) + S(J+240)
      S(205) = S(205) + S(J+240)*XBAR(J)
   30 CONTINUE
      S(208) = S(205)/S(204)
      IF(S(208) - S(214)) 40,70,50
   40 DO 41 J=L,K
      S(J+260) = (XBAR(J) - S(201))/S(203)*DELX(J)
      S(206) = S(206) + S(J+260)
      S(207) = S(207) + S(J+260)*XBAR(J)
   41 CONTINUE
      GO TO 71
   50 DO 51 J=L,K
      S(J+260) = (S(217)- XBAR(J))/S(203)*DELX(J)
      S(206) = S(206) + S(J+260)
      S(207) = S(207) + S(J+260)*XBAR(J)
   51 CONTINUE
      GO TO 71
   70 S(213) = D(24)
      S(211) = S(215)
      GO TO 80
   71 S(209) = S(207)/S(206)
      S(210) = S(215)*(S(214) - S(208))/(S(209) - S(208))
      S(211) = S(215) - S(210)
      S(213) = S(210)/S(206)
   80 S(212) = S(211)/S(204)
      DO 81 J=L,K
      S(J+220) = S(J+240)*S(212) + S(J+260)*S(213)
   81 CONTINUE
```

CARD NO **** CONTENTS ****

```
      RETURN
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                          SUBROUTINE DSTTRP
C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE DSTTRP
C          TRAPEZOIDAL DISTRIBUTION OF WT.          JULY 1972
C.-.-.-.-.-.NOTE, INCREASING POINT VALUE --- XO(I),XBAR(I+1),XO(I+1) ETC
C           AND DELX(I+1) IS DIST BTN XJ(I) AND XO(I+1)
C     GIVEN-  S(1) = WEIGHT
C             S(2) = C.G.
C             S(3) = FORWARD EXTENT OF DISTRIBUTION
C             S(4) = AFT EXTENT OF DISTRIBUTION
C     OUTPUT  J INDEX FOR FWD XBAR AT WHICH LOAD IS PLACED.
C             K INDEX FOR AFT XBAR AT WHICH LOAD IS PLACED.
C             S(J+20) THRU S(K+20) WEIGHTS FOR ABOVE XBAR LOCATIONS.
C
       COMMON TCOM(4320)
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
     *, GDB(80), DVB(440), XO(20), XBAR(20), DELX(20)
      EQUIVALENCE (TCOM(1),D(1)), (TCOM(701),GD(1)), (TCOM(1401),DV(1))
     *, (TCOM(3721),S(1)), (TCOM(4121),ND(1))
     *, (GD(381),GDB(1)), (GDB(56),XO(1))
     *, (DV(441),DVB(1)), (DVB(141),XBAR(1)), (DVB(161),DELX(1))
     *, (ND(102),J), (ND(103),K), (ND(115),NC)
      NCL = NC+1
      IF(S(3) -XO(1)) 10,10,15
C     FORWARD OF  XO(1) SET EQUAL TO FIRST
   10 J=1
      GO TO 60
   15 N=2
   20 IF(S(3) -XO(N)) 25,30,40
   25 IF((XO(N)-S(3) )-(S(3) -XO(N-1))) 30,30,35
   30 J = N
      GO TO 60
   35 J = N - 1
      GO TO 60
   40 N=N+1
      IF(N.LE.NCL) GO TO 20
      J=NCL
      K=NCL
C ERROR  PRINT
   50 WRITE(6,55) S(1), S(2), S(3), S(4), J, K
   55 FORMAT(///1H0,45H***** IN DSTTRP, WEIGHT CANNOT BE DISTRIBUTED /
     * 18H, SWT=,1F14.2,2X, 5HCG=,1F9.2, 2X, 4HFWD=, 1F9.2, 4HAFT=,
     * 1F9.2,18HRETURN J,K =, 113, 1H,,113 )
      GO TO 102
C     AFT OF LAST STATION SET TO LAST
   60 IF(S(4)-XO(NCL)) 70,65,65
   65 K=NCL
      GO TO 100
   70 N=NC
   72 IF(S(4)-XO(N)) 80,85,75
   75 IF((XO(N+1)-S(4))-(S(4)-XO(N))) 80,80,85
   80 K=N+1
      GO TO 100
   85 K=N
      GO TO 100
   90 N=N-1
      IF(N.GE.1) GO TO 72
      K=1
      J=1
      GO TO 90
  100 IF((K-J) .LT. 1) GO TO 50
C
      IF(K - J - 1) 102,102,104
  102 CALL DSTHOR
      GO TO 999
  104 DO 106 N=5,8
```

```
 CARD NO     ****                        CONTENTS                        ****

  2769        105  S(N) = D(24)
  2770             S(9) = XD(K)-XD(J-1)
  2771             S(10)= S(9)/D(2)
  2772             S(11)= XD(J-1)+S(10)
  2773             DO 110 N=J,K
  2774             S(N+60) = (S(11) - XBAR(N))/S(10)*DELX(N)
  2775             S(5) = S(5) + S(N+60)
  2776             S(6) = S(6) + S(N+60)*XBAR(N)
  2777             S(N+80) = DELX(N)/S(9)
  2778             S(7) = S(7) + S(N+80)
  2779             S(8) = S(8) + S(N+80)*XBAR(N)
  2780        110  CONTINUE
  2781             S(12) = S(8)/S(7)
  2782             S(13) = (S(1)*S(12)-S(1)*S(2))/(S(5)*S(12)-S(6))
  2783             S(14) = (S(1)-S(13)*S(5))/S(7)
  2784             DO 140 N=J,K
  2785             S(N+20) = S(N+60)*S(13) + S(N+80)*S(14)
  2786        140  CONTINUE
  2787        999  RETURN
  2788             END
  2789       C
  2790       C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
  2791       C                          SUBROUTINE DUCGEO
  2792       C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  2793       C
  2794             SUBROUTINE DUCGEO
  2795       C     WRITTEN 23 MARCH 1972
  2796       C     TO DEVELOP DUCT GEOMETRY
  2797       C
  2798              COMMON TCOM(4320)
  2799       C
  2800             DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
  2801             DIMENSION EQU(200)
  2802             DIMENSION DATD(70)
  2803             DIMENSION DVD(100)
  2804             DIMENSION WOD(10),ROD(10),DDD(10),BUD(10),BLD(10),BSD(10),
  2805            1 DLXD(10),SFD(10),XBD(10)
  2806       C
  2807             EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
  2808            1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
  2809             EQUIVALENCE (EQU(1),D(81))
  2810             EQUIVALENCE (DATD(1),GD(501))
  2811             EQUIVALENCE (DVD(1),DV(871))
  2812             EQUIVALENCE (WOD(1),DVD(1)),(ROD(1),DVD(11)),(DDD(1),DVD(21)),
  2813            1 (BUD(1),DVD(31)),(BLD(1),DVD(41)),(BSD(1),DVD(51)),
  2814            2 (DLXD(1),DVD(61)),(SFD(1),DVD(71)),(XBD(1),DVD(81))
  2815             EQUIVALENCE (ND(101),I),(ND(102),J)
  2816             EQUIVALENCE (ND(114),IDD),(ND(117),NCD),(ND(118),KCD)
  2817       C
  2818       C        SETUP INDICATORS AND COUNTERS
  2819             IF(1 - KCD) 10,20,20
  2820       C       SHAPE CODE IS 2 DEVELOP PERIMETER
  2821          10 DO 15 I=1,NCD
  2822             IF(DATD(I+60)) 15,15,12
  2823          12 DATD(I+60) = DATD(I+60)*(DATD(I+40) + DATD(I+50))*D(15)/D(2)
  2824          15 CONTINUE
  2825       C       FIT SHAPES
  2826          20 DO 200 I=1,NCD
  2827             IF(DATD(I+60)) 30,30,100
  2828       C        PERIMETER IS ZERO CHECK ON GUIDE
  2829          30 IF(DATD(I+40)) 34,34,32
  2830       C        VERTICAL LIP (INBD WEDGE)
  2831          32 IDD = 1
  2832             DATD(I+60) = DATD(I+40)
  2833             GO TO 200
  2834          34 IF(DATD(I+50)) 36,36,38
  2835          36 WRITE(6,80)
  2836         80  FORMAT(39H0WARNING FROM DUCGEO IN DATA MANAGEMENT /39X,23HDUCT LIP
  2837           1 GEOMETRY ERROR )
  2838       C        HORIZONTAL LIP (UPPER L.E.)
  2839          38 IDD = 2
```

```
CARD NO  ****                          CONTENTS                          ****

 2040          DATD(I+60) = DATD(I+50)
 2041          GO TO 200
 2042      100 S(1) = D(1)
 2043          S(2) = (D(2)*(DATD(I+40) + DATD(I+50)) - DATD(I+60))/
 2044         1(D(8) - D(2)*D(15))
 2045          IF(S(2)) 101,101,102
 2046      101 S(1) = DATD(I+60)/(D(2)*DATD(I+40) + D(2)*DATD(I+50))
 2047          S(2) = D(24)
 2048          IF(S(1) .GT. EQU(145) .AND. S(1) .LT. EQU(146)) GO TO 110
 2049      C
 2050      C       ERROR MESSAGE
 2051          WRITE(6,95) J,S(1)
 2052       95 FORMAT(38HBENCHMARKING FROM DUCDEO IN DATA MANAGEMENT / 8H SECTION,
 2053         1 I3, 41H IS RECT. OR ROUNDED RECT., CORRECTION IS, 1F7.4 )
 2054          GO TO 110
 2055      C
 2056      102 S(4) = AMAX1(DATD(I+40),DATD(I+50))
 2057          S(5) = AMIN1(DATD(I+40),DATD(I+50))
 2058          IF(S(5) - D(2)*S(2)) 109,110,110
 2059      109 S(2) = S(5)/D(2)
 2060          S(1) = DATD(I+60)/(D(2)*(D(15)*S(2) + S(4) - D(2)*S(2)))
 2061          IF(S(1) .GT. EQU(145) .AND. S(1) .LT. EQU(146)) GO TO 110
 2062      C
 2063      C       ERROR MESSAGE
 2064          WRITE(6,95) I,S(1)
 2065      110 S(6) = (DATD(I+50) - D(2)*S(2))*S(1)/D(2)
 2066          S(3) = (DATD(I+40) - D(2)*S(2))*S(1)/D(2)
 2067          IF(S(6)) 111,112,112
 2068      111 S(6) = D(24)
 2069      112 IF(S(3)) 114,115,115
 2070      114 S(3) = D(24)
 2071      115 ROD(I) = S(6)
 2072          RDD(I) = S(2)*S(1)
 2073          DDD(I) = S(3)
 2074          BUD(I) = D(2)*RDD(I) + D(15)/D(2)*ROD(I)
 2075          BLD(I) = BUD(I)
 2076          BSD(I) = D(2)*DDD(I) + D(15)/D(2)*ROD(I)
 2077      200 CONTINUE
 2078          J = 2
 2079          IF(IDD) 250,250,205
 2080      C       CALCULATE LEADING EDGE SURFACE
 2081      205 J = 3
 2082          DLXD(1) = DATD(12) - DATD(1)
 2083          XSD(1) = DATD(1) + D(2)/D(3)*DLXD(1)
 2084          IF(IDD-1) 208,208,220
 2085      208 IF(DATD(22)) 210,210,208                                    00120900
 2086      C       SECOND CUT IS OFFSET THEREFORE THERE ARE TWO INLETS PER NACELLE00120905
 2087      209 IF(DATD(21)) 207,207,209                                    00120907
 2088      C       FOR VERTICAL LIP CALCULATE LIP, TWO TRIANGLES PLUS VERTICAL RE00120910
 2089      207 SFD(1) = DLXD(1)*((DATD(6) + BSD(2))/D(2) + BUD(2) + BLD(2))  00120920
 2090          GO TO 250                                                   00120930
 2091      C       SPLIT INLET AS PER FUSELAGE MOUNTED                     00120932
 2092      208 SFD(1) = DLXD(1)*(DATD(6)) + BSD(2) + BUD(2) + BLD(2))      00120934
 2093          GO TO 250                                                   00120936
 2094      C       THERE IS A SINGLE INLET PER NACELLE                     00120938
 2095      210 SFD(1) = DLXD(1)/D(2)*(DATD(6)) + BUD(2) + BSD(2) + BLD(2))
 2096          GO TO 250
 2097      C       HORIZONTAL LIP                                          00120950
 2098      220 IF(DATD(22)) 224,224,222                                    00120960
 2099      C       TWO INLETS PER NACELLE                                  00120962
 2100      222 SFD(1) = DLXD(1)*(DATD(6)) + BUD(2) + BSD(2)*D(3)/D(2))     00120964
 2101          GO TO 250                                                   00120966
 2102      C       ONE INLET PER NACELLE                                   00120967
 2103      224 SFD(1) = DLXD(1)/D(2)*(DATD(6)) + BUD(2) + D(2)*BSD(2))     00120968
 2104      C       CALCULATE SUBSEQUENT SECTIONS OR NOSE IF NO L.E.
 2105      250 DO 300 I=J,NCD
 2106          DLXD(I-1) = DATD(I+10) - DATD(I+9)
 2107          XSD(I-1) = DATD(I+9) + DLXD(I-1)/D(2)
 2108          S(1) = D(2)
 2109          IF(DATD(I+29)) 260,260,252                                  00121010
 2110      252 IF(DATD(I+19)) 265,265,260
```

```
CARD NO  ****                          CONTENTS                          ****
  2011       295 S(I) = D(I)
  2012       270 SFD(I-1) = DLXD(I-1)/D(2)*S(I)*(DATD(I+60) + DATD(I+59))
  2013           GO TO 300
  2014     C       SHARP TRANSITION TWO DUCTS BECOMING ONE
  2015       280 SFD(I-1) = DLXD(I-1)*DATD(I+60)
  2016           GO TO 300                                          80121072
  2017       282 IF(DATD(I+19)) 284,284,270                         80121074
  2018     C       SHARP TRANSITION ONE DUCT BECOMING TWO            80121075
  2019       284 SFD(I-1) = DLXD(I-1)*DATD(I+60)*D(2)               80121076
  2020       300 CONTINUE
  2021     C       *** EXIT ***
  2022           RETURN
  2023           END
  2024     C
  2025     C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
  2026     C                        SUBROUTINE D4MOD
  2027     C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  2028     C
  2029           SUBROUTINE D4MOD
  2030     C
  2031     C       THIS ROUTINE SETS UP SPEED ALTITUDE AND H-TAIL INERTIA
  2032     C       DATA IN SPAL ARRAY FOR USE IN FLUTTER AND TEMPERATURE MODULE
  2033     C
  2034           COMMON TCOM(4320)
  2035     C
  2036           DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
  2037     C
  2038           DIMENSION SPAL(50)
  2039     C
  2040           DIMENSION GD1(20),GDH(40),DATM(40),ALT(10),VL(10)
  2041     C
  2042           DIMENSION DVAT(1000)
  2043     C
  2044           EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401))
  2045          * , (S(1),TCOM(3721)),(ND(1),TCOM(4121))
  2046     C
  2047           EQUIVALENCE (SPAL(1),S(1))
  2048     C
  2049           EQUIVALENCE (GD1(1),GD(1)),(GDH(1),GD(30)),(DATM(1),GD(51))
  2050          * , (ALT(1),DV(1)),(VL(1),DV(7))
  2051     C
  2052           EQUIVALENCE (DVAT(1),DV(121))
  2053     C
  2054           CALL READMS(1),SPAL(1),50,38)
  2055     C
  2056     C       HORIZONTAL WEIGHT AND INERTIA FOR VERTICAL TAIL FLUTTER
  2057     C       CALCULATIONS
  2058           SPAL(1) = DVAT(968)/D(2)
  2059           SPAL(2) = DVAT(970)
  2060           SPAL(3) = DVAT(969)
  2061           SPAL(4) = GDH(1)
  2062           SPAL(5) = DVAT(972)
  2063           SPAL(6) = D(24)
  2064           SPAL(7) = DVAT(973)
  2065           SPAL(12) = GDH(10)
  2066           SPAL(16) = D(34)
  2067     C
  2068     C       SPEED ALTITUDE PROFILE FOR WINDS FIXED OR AFT
  2069           DO 501 I=1,8
  2070           SPAL(I+8) = ALT(I)
  2071           SPAL(I+25) = VL(I)
  2072       501 CONTINUE
  2073     C
  2074     C       TEST FOR VARIABLE SWEEP WING
  2075           IF(GD1(2)) 520,520,505
  2076     C
  2077     C       VARIABLE SWEEP WING, FORWARD POSITION
  2078       505 DO 507 I=1,8
  2079           SPAL(I+34) = DATM(I+32)
  2080       507 CONTINUE
  2081     C
```

```
CARD NO  ****                          CONTENTS                          ****
0002     C       ************************************************************
0003     C       WRITE RECORD 30 FOR FLUTTER MODULE, COMPRESSIBLE
0004     C       DYNAMIC PRESSURE
0005       820 CALL WRITMS(1,QPAL(1),90,30)
0006     C
0007     C       ************************************************************
0008     C
0009           RETURN
0010           END
0011     C
0012     C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
0013     C                          SUBROUTINE FTOTAL
0014     C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
0015     C
0016           SUBROUTINE FTOTAL
0017     C       WRITTEN 24 JULY 1972
0018     C       TO SETUP FUSELAGE USEFUL LOAD DISTRIBUTION AND CONTENTS FOR
0019     C       THREE GROSS WEIGHTS.
0020            COMMON TCON(4320)
0021           DIMENSION D(700),OD(700),DV(2320),S(400),ND(200)
0022           DIMENSION GDWT(160)
0023           DIMENSION DWT(1000),WFC1(20),WFC2(20),WFC3(20)
0024           EQUIVALENCE (D(1),TCON(1)),(OD(1),TCON(701)),(DV(1),TCON(1401)),
0025          1 (S(1),TCON(3721)),(ND(1),TCON(4121))
0026           EQUIVALENCE (GDWT(1),GD(4
0027           EQUIVALENCE (DWT(1),DV(
0028          1 (WFC2(1),DWT(801)),(WFC3(1),DWT(821))
0029           EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104)),
0030          1 (N,ND(106)),(II,ND(107)),(JJ,ND(108)),(LL,ND(110))
0031           EQUIVALENCE (NC,ND(115))
0032     C       SETUP COUNTERS   II = GROSS WEIGHT PASS COUNTER
0033           II = 1
0034       100 DO 101 I=1,20
0035           S(I+20) = D(I+)
0036           S(I+40) = D(I+)
0037       101 CONTINUE
0038     C       TEST ON FUSELAGE PAYLOAD
0039           JJ = II*10 + 111
0040           S(1) = GDWT(81)*GDWT(JJ)
0041           IF(S(1)) 150,150,103
0042       103 S(2) = GDWT(81)
0043           S(3) = GDWT(101)
0044           S(4) = GDWT(102)
0045           S(5) = (S(4)-S(3))/D(3)
0046           IF( (S(2)-S(3)) - S(5) ) 110,104,104
0047       104 IF( (S(4)-S(2)) - S(5) ) 110,105,105
0048       105 CALL DSTTRP
0049           GO TO 120
0050       110 CALL DSTNOR
0051           WRITE(6,115) S(3),S(4),S(1),S(2)
0052       115 FORMAT(2H0*** WARNING FROM FTOTAL *** /
0053          1 20X,33HPREVIOUS FORE AND AFT LIMITS WERE, 3F10.2/
0054          2 23X,F12.2,17H LBS AT FUS. STA.,F8.2,10X,21HDISTRIBUTED BY DSTNOR)
0055           WRITE(6,116)
0056       116 FORMAT(1H+,19X,18HFUSELAGE PAYLOAD )
0057       120 DO 125 I=J,K
0058           S(I+40) = S(I+40) + S(I+20)
0059       125 CONTINUE
0060     C       TEST ON AMMUNITION
0061       150 S(1) = GDWT(83)*GDWT(JJ+2)
0062           IF(S(1)) 200,200,152
0063       152 S(2) = GDWT(83)
0064           CALL DSTNOR
0065           DO 155 I=J,K
0066           S(I+40) = S(I+40) + S(I+20)
0067       155 CONTINUE
0068     C       TEST ON FUSELAGE FUEL, FIVE TANKS
0069       200 DO 300 L=1,5
0070           N = JJ + L + 5
0071           S(1) = GDWT(L+85)*GDWT(N)
0072           IF(S(1)) 300,300,202
```

CARD NO **** CONTENTS ****

```
      202 S(2) = QDWT(L+95)
          LL = L+2
          S(3) = QDWT(LL+107)
          S(4) = QDWT(LL+108)
          S(5) = (S(4)-S(3))/D(3)
          IF( (S(2)-S(3)) - S(5) ) 210,204,204
      204 IF( (S(4)-S(2)) - S(5) ) 210,205,205
      205 CALL DSTRP
          GO TO 220
      210 CALL DSTNOR
          WRITE(6,115) S(3),S(4),S(5),S(2)
          WRITE(6,215)
      215 FORMAT(1H+,22X,13HFUSELAGE FUEL )
      220 DO 225 I=J,K
          S(I+40) = S(I+80) + S(I+20)
      225 CONTINUE
      300 CONTINUE
          L = NC + 1
          GO TO (310,320,330),II
      310 II = 2
          DO 312 I=1,L
          WFC1(I) = DWAT(I+360) + S(I+40)
      312 CONTINUE
          GO TO 100
      320 II = 3
          DO 322 I=1,L
          WFC2(I) = DWAT(I+360) + S(I+40)
      322 CONTINUE
          GO TO 100
      330 DO 332 I=1,L
          WFC3(I) = DWAT(I+360) + S(I+40)
      332 CONTINUE
          RETURN
          END
C
C ***********************************************************************
C                        SUBROUTINE FUSDST
C ***********************************************************************
C
      SUBROUTINE FUSDST
C       WRITTEN 9 JULY 1972
C       TO DISTRIBUTE FUSELAGE WEIGHT FOR LOADS AND INERTIA CALCULATION
      COMMON TC(2)*520)
      DIMENSION D(700),QD(700),DV(2370),S(400),ND(200)
      DIMENSION DWAT(500),WFUS(20)
      DIMENSION QDB(80),DVB(40)
      DIMENSION X1(10),XD(20),XBAR(20),SF(20),TOT(20)
      EQUIVALENCE (D(1),TCOM(1)),(QD(1),TCOM(70)),(DV(1),TCOM(140)),
     1 (S(1),TCOM(372)),(ND(1),TCOM(412))
      EQUIVALENCE (DWAT(1),DV(1121)), (WFUS(1),DWAT(3A1))
      EQUIVALENCE (QDB(1),QD(181)),(DVB(1),DV(43))
      EQUIVALENCE (X1(1),QDB(61)),(XD(1),QDB(56)),(XBAR(1),DVB(19)),
     1 (SF(1),DVB(101)),(TOT(1),DVB(36))
      EQUIVALENCE (I,ND(10)),(J,ND(102)),(K,ND(103)),(L,ND(104))
      EQUIVALENCE (NF,ND(115))
C       DETERMINE FACTOR
      S(1) = DWAT(101)/TOT(1)/D(2)
      J = NC + 1
      S(2) = D(24)
      S(3) = D(24)
      DO 100 I=1,J
      WFUS(I) = S(I)*SF(I)
      S(2) = S(2) + WFUS(I)*XBAR(I)
      S(3) = S(3) + WFUS(I)
  100 CONTINUE
      S(4) = DWAT(101) - S(3)
      S(5) = (DWAT(101)*DWAT(104) - S(2))/S(4)
C       DISTRIBUTE HALF OF FUSELAGE WEIGHT ACCORDING TO TRIANGULAR DIST.
      S(201) = X1(1)
      S(202) = S(5) - S(201)
      S(203) = X1(10) - X1(1)
```

```
CARD NO  ****                          CONTENTS                          ****
    3124          S(214) = S(5)
    3125          S(215) = S(4)
    3126          I = 1
    3127          IF(MD(NC) - S(214)) 104,104,106
    3128      104 I = NC
    3129          WRITE(6,68) DVAT(101),DVAT(141)
    3130       68 FORMAT(21HD** FUSDST WARNING **    /10X,20HCHECK NT AND CG DATA,
    3131         * /10X,33HFUSELAGE NT DIST IS NOT REALISTIC,/10X,4HNT =,F10.2,
    3132         * 3X,4HCG =,F8.2)
    3133          GO TO 108
    3134      106 IF(MD(I) - S(214)) 107,108,108
    3135      107 I = I + 1
    3136          GO TO 105
    3137      108 K = NC + 1
    3138          L = 1
    3139          CALL DSTRI
    3140          DO 110 I=1,K
    3141          NFUS(I) = NFUS(I) + S(I+220)
    3142      110 CONTINUE
    3143          RETURN
    3144          END
    3145      C
    3146      C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
    3147      C                          SUBROUTINE FUSGEO
    3148      C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
    3149      C
    3150          SUBROUTINE FUSGEO
    3151      C   INITIAL ROUTINE FOR EXTERNAL SHELL GEOMETRY                  70020020
    3152      C   FAMILY OF ROUNDED RECTANGLES                               70020030
    3153      C                                                              70020050
    3154           COMMON TCOM(4320)
    3155      C                                                              70020060
    3156          DIMENSION D(700),QD(700),DV(2320),S(480),ND(200)
    3157          DIMENSION EQU(200)
    3158          DIMENSION QDB(80),DVB(440)
    3159          DIMENSION X(10),Z(10),DI(10),MI(10),PI(10),XO(20)
    3160          DIMENSION ZC(20),RCU(20),RCL(20),RCS(20),BU(20),BL(20),
    3161         1 BS(20),XBAR(20),DELX(20),SF(20),VOL(20),DDD(20),WD(20),RO(20),
    3162         2 PER(20)
    3163          DIMENSION S1(20),S2(20),S3(20),TOT(20)
    3164          DIMENSION UIX(20),UIY(20),UIZ(20)
    3165      C                                                              70020070
    3166          EQUIVALENCE (D(1),TCOM(1)),(QD(1),TCOM(701)),(DV(1),TCOM(1401)),
    3167         1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
    3168          EQUIVALENCE (EQU(1),D(81))
    3169          EQUIVALENCE (QDB(1),QD(381)),(DVB(1),DV(4321))
    3170          EQUIVALENCE (X(1),QDB(6)),(Z(1),QDB(16)),(DI(1),QDB(26)),
    3171         1 (MI(1),QDB(36)),(PI(1),QDB(46)),(XO(1),QDB(56))
    3172          EQUIVALENCE (ZC(1),DVB(1)),(RCU(1),DVB(21)),(RCL(1),DVB(41)),
    3173         1 (RCS(1),DVB(61)),(BU(1),DVB(81)),(BL(1),DVB(101)),
    3174         2 (BS(1),DVB(121)),(XBAR(1),DVB(141)),(DELX(1),DVB(161)),
    3175         3 (SF(1),DVB(181)),(VOL(1),DVB(201)),(DDD(1),DVB(221)),
    3176         4 (WD(1),DVB(241)),(RO(1),DVB(261)),(PER(1), DVB(281))
    3177          EQUIVALENCE (S1(1),DVB(301)),(S2(1),DVB(321)),(S3(1),DVB(341)),
    3178         1 (TOT(1),DVB(361))
    3179          EQUIVALENCE (UIX(1),DVB(381)),(UIY(1),DVB(401)),(UIZ(1),DVB(421))
    3180          EQUIVALENCE (TOT(1),STOT),(TOT(2),VOLT)
    3181          EQUIVALENCE (ND(115),NC), (ND(116),KC)
    3182      C   EQUIVALENCE SCRATCH COUNTERS                               70020210
    3183          EQUIVALENCE (ND(101),I),(ND(102),J)
    3184      C                                                              70020350
    3185      C   SO = FUSELAGE WATER PLANE FOR EVALUATION                   70020370
    3186      C   MD = FLAT HORIZONTAL          DDD = FLAT VERTICAL          70020390
    3187      C   DESCRIPTION OF OUTPUT ARRAYS                               70020390
    3188      C   XBAR = CENTROID OF SEGMENT    DELX = SEGMENT LENGHT        70020410
    3189      C   BU = UPPER PANEL CIRCUM       BL = LOWER PANEL CIRCUM      70020430
    3190      C   BS = SIDE PANEL CIRCUM        RCU = RADIUS OF CURVATURE UPPER 70020450
    3191      C   RCL = RADIUS OF CURVATURE LOWER RCS = RADIUS OF CURVATURE SIDE 70020470
    3192      C   PER = PERIMETER               RO = CORNER RADIUS           70020490
    3193      C   SF = SURFACE AREA SEGMENT     VOL = VOLUME OF SEGMENT      70020510
    3194      C                                                              70020530
```

```
CARD NO.  ****                          CONTENTS                          ****
 3186       C     DESCRIPTION OF INPUT ARRAYS                                 70020470
 3188       C     ZI = FUSELAGE WATER PLANE FOR GEOMETRY                      70020480
 3187       C     XI = FUS STATION FOR GEOMETRY    DI = FUS DEPTH             70020490
 3188       C     WI = FUS WIDTH                   PI = PERIMETER - NC = 1    70020500
 3189       C     PI = PERIMETER CORRECTION - NC+2 XO = FUS STATION FOR EVALUATION 70020510
 3200       C     I = INPUT GEOMETRY COUNTER                                  70020520
 3201       C     J = OUTPUT GEOMETRY COUNTER                                 70020530
 3202       C                                                                 70020540
 3203       C     SCRATCH ARRAYS                                              70020550
 3204       C     S1 = DEPTH                       S2 = WIDTH                 70020560
 3205       C     S3 = AREA CROSS-SECTION                                     70020570
 3206             IF(I-NC)10,15,15                                            70020580
 3207       C                                                                 70020590
 3208       C     CONVERT PERIMETER CORRECTION TO PERIMETER AND SUBSTITUTE     70020600
 3209          10 DO 12 I=1,10                                                70020610
 3210             PI(I) = PI(I)*(DI(I) + WI(I))*D(15)/D(2)                    70020620
 3211          12 CONTINUE                                                    70020630
 3212          15 I = 2                                                       70020640
 3213             J = 1                                                       70020650
 3214             STOT = D(24)                                                70020660
 3215             VOLT = D(24)                                                70020670
 3216       C                                                                 70020680
 3217       C     STRAIGHT LINE INTERPOLATION                                 70020690
 3218             TOT(19) = DI(1)                                             70020692
 3219             TOT(20) = WI(1)                                             70020694
 3220          20 IF(XO(J)-XI(I))21,21,22                                     70020700
 3221          21 S(1) = (XO(J) - XI(I-1))/(XI(I)-XI(I-1))                    70020710
 3222             S1(J) = DI(I-1) + S(1)*(DI(I)-DI(I-1))                      70020720
 3223             S2(J) = WI(I-1) + S(1)*(WI(I)-WI(I-1))                      70020730
 3224             PER(J) = PI(I-1) + S(1)*(PI(I)-PI(I-1))                     70020740
 3225             ZO(J) = ZI(I-1) + S(1)*(ZI(I)-ZI(I-1))                      70020750
 3226             IF(TOT(19) - S1(J)) 31,32,32                                70020753
 3227          31 TOT(19) = S1(J)                                             70020754
 3228          32 IF(TOT(20) - S2(J)) 33,34,34                                70020755
 3229          33 TOT(20) = S2(J)                                             70020756
 3230          34 CONTINUE                                                    70020757
 3231             J = J+1                                                     70020760
 3232             IF(J-NC)20,20,100                                           70020770
 3233          22 I = I+1                                                     70020780
 3234             GO TO 20                                                    70020790
 3235       C                                                                 70020800
 3236       C     SHAPE FIT                                                   
 3237         100 DO 200 J=1,NC                                               
 3238             S(1) = D(1)                                                 
 3239             S(2) = (D(2)*S1(J) + D(2)*S2(J) - PER(J))/(D(8) - D(2)*D(15)) 70020840
 3240             IF(S(2))101,101,102                                         70020850
 3241         101 S(1) = PER(J)/(D(2)*S1(J) + D(2)*S2(J))                     70020860
 3242             S(2) = D(24)                                                70020870
 3243             IF(S(1) .GT. EQU(145) .AND. S(1) .LT. EQU(146)) GO TO 110   
 3244       C                                                                 70020880
 3245       C     ERROR MESSAGE                                               70020890
 3246             WRITE(6,61)                                                 70020902
 3247          61 FORMAT( 38HBOMBARDMENT FROM FUSGEO IN DATA MANAGEMENT )     70020903
 3248             WRITE(6,65) J,S(1)                                          70020910
 3249          65 FORMAT( 8H SECTION, I13,32H IS RECTANGULAR, CORR. FACTOR IS,F8.3)70020920
 3250             GO TO 110                                                   70020940
 3251         102 S(4) = AMAX1(S1(J),S2(J))                                   70020950
 3252             S(5) = AMIN1(S1(J),S2(J))                                   70020960
 3253             IF(S(5) - D(2)*S(2)) 103,110,110                            70020970
 3254         103 S(2) = S(5)/D(2)                                            70020980
 3255             S(1) = PER(J)/(D(2)*(D(15)*S(2) + S(4) - D(2)*S(2)))        70020990
 3256             IF(S(1) .GT. EQU(145) .AND. S(1) .LT. EQU(146)) GO TO 110   
 3257       C                                                                 70021000
 3258       C     ERROR MESSAGE                                               70021010
 3259             WRITE(6,81)                                                 70021020
 3260             WRITE(6,65) J,S(1)                                          70021030
 3261          65 FORMAT( 8H SECTION,I13,34H IS ROUNDED RECT., CORR. FACTOR IS,F8.3)70021040
 3262         110 S(6) = (S2(J) - D(2)*S(2))*S(1)/D(2)                        70021050
 3263             S(3) = (S1(J) - D(2)*S(2))*S(1)/D(2)                        70021060
 3264             IF(S(6))111,112,112                                         70021070
 3265         111 S(6) = D(24)                                                70021080
```

```
CARD NO      ****                          CONTENTS                          ****
  2666        112 IF(S(3))114,115,115                                        70021080
  2667        114 S(3) = D(24)                                               70021100
  2668        115 RO(J) = S(6)                                               70021110
  2669            RO(J) = S(2)*S(1)                                          70021120
  2670            OOO(J) = S(3)                                              70021130
  2671            SI(J) = D(15)*RO(J)**2 + D(4)*OOO(J)*(RO(J)+RO(J)) +       70021140
  2672           1D(4)*RO(J)*RO(J)                                           70021150
  2673            BU(J) = D(2)*RO(J) + D(15)/D(2)*RO(J)                      70021160
  2674            BL(J) = BU(J)                                              70021170
  2675            BS(J) = D(2)*OOO(J) + D(15)/D(2)*RO(J)                     70021180
  2676            IF(D(2)-RO(J))116,116,200                                  70021190
  2677        116 S(1) = RO(J)*(D(1) - D(25)/D(2))                           70021200
  2678            S(2) = RO(J)*D(25)/D(2) + OOO(J)                           70021210
  2679            S(3) = RO(J)*D(25)/D(2) + RO(J)                            70021220
  2680            RCU(J) = (S(1)**2 + S(3)**2) D(2)/S(1)                     70021230
  2681            RCL(J) = RCU(J)                                            70021240
  2682            RCS(J) = (S(1)**2 + S(2)**2)/D(2)/S(1)                     70021250
  2683      C     IF R12.8 ASSUME RADIUS OF CURVATURE IS INFINITY IE FLAT PANEL 70021260
  2684        200 CONTINUE                                                   70021270
  2685      C                                                                70021280
  2686      C     NOSE CONE GEOMETRY                                         70021290
  2687            S(1) = PI(1)/D(2)/D(15)                                    70021300
  2688            S(2) = PER(1)/D(2)/D(15)                                   70021310
  2689            DELX(1) = XO(1) - X1(1)                                    70021320
  2690            SF(1) = D(15)*(S(1)+S(2))*(DELX(1)**2 + (S(2)-S(1))**2)** 5 70021330
  2691            XBAR(1) = XO(1) - DELX(1)/D(3)*(D(2)*S(1) + S(2))/(S(1)+S(2)) 70021340
  2692            VOL(1) = D(15)*DELX(1)/D(3)*(S(2)**2 + S(2)*S(1) + S(1)**2) 70021350
  2693            DO 300 J=2,NC                                              70021360
  2694            XBAR(J) = (XO(J) + XO(J-1))/D(2)                           70021370
  2695            DELX(J) = XO(J) - XO(J-1)                                  70021380
  2696            IF(DELX(J)-D(2))301,301,302                                70021390
  2697      C                                                                70021400
  2698      C     SHARP DISCONTINUITY IN GEOMETRY                            70021410
  2699        301 SF(J) = PER(J)*DELX(J)                                     70021420
  2700            VOL(J) = DELX(J)*S3(J)                                     70021430
  2701            GO TO 300                                                  70021440
  2702      C                                                                70021450
  2703      C     NORMAL GEOMETRIC TRANSITION                                70021460
  2704        302 S(1) = PER(J-1)/D(2)/D(15)                                 70021470
  2705            S(2) = PER(J)/D(2)/D(15)                                   70021480
  2706            SF(J) = D(15)*(S(1)+S(2))*(DELX(J)**2 + (S(2)-S(1))**2)**.5 70021490
  2707            VOL(J) = DELX(J)/D(3)*(S3(J)+S3(J-1) + (S3(J)*S3(J-1))**.5) 70021500
  2708        300 CONTINUE                                                   70021510
  2709      C                                                                70021520
  2710      C     TAIL CONE GEOMETRY                                         70021530
  2711            J = NC                                                     70021540
  2712            S(1) = PER(J)/D(2)/D(15)                                   70021550
  2713            S(2) = PI(10)/D(2)/D(15)                                   70021560
  2714            XO(J+1) = X1(10)                                           70021570
  2715            ZO(J+1) = Z1(10)                                           
  2716            DELX(J+1) = X1(10) - XO(J)                                 70021580
  2717            XBAR(J+1) = XO(J) + DELX(J+1)/D(3)*(D(2)*S(2) + S(1))/(S(2)+S(1)) 70021590
  2718            SF(J+1) = D(15)*(S(1)+S(2))*(DELX(J+1)**2+(S(1)-S(2))**2)**.5 70021600
  2719            VOL(J+1) = D(15)*DELX(J+1)/D(3)*(S(1)**2 + S(1)*S(2) + S(2)**2) 70021610
  2720            I = NC+1                                                   70021620
  2721            DO 400 J=1,I                                               70021630
  2722            STOT = STOT + SF(J)                                        70021640
  2723            VOLT = VOLT + VOL(J)                                       70021650
  2724        400 CONTINUE                                                   70021660
  2725      C     DESCRIPTION OF OUTPUT ARRAYS                               70030370
  2726      C     UIX = UNIT ROLL INERTIA          UIY = UNIT PITCH INERTIA  70030380
  2727      C     UIZ = UNIT YAW INERTIA                                     70030390
  2728      C                                                                70030400
  2729      C     NOSE CONE                                                  70030410
  2730            S(1) = PI(1)/D(2)/D(15)                                    70030420
  2731            S(2) = PER(1)/D(2)/D(15)                                   70030430
  2732            S(3) = XO(1) - XBAR(1)                                     70030440
  2733            UIX(1) = D(3)/D(10)*(S(1)**4 + S(1)**3*S(2) + S(1)**2*S(2)**2 + 70030450
  2734           1 S(1)*S(2)**3 + S(2)**4)/(S(1)**2 + S(1)*S(2) + S(2)**2)   70030451
  2735            UIY(1) = UIX(1)/D(2) + D(3)*D(9)/D(8)/D(10)*S(3)**2        70030460
  2736            UIZ(1) = UIY(1)                                            70030470
```

```
 CARD NO    ****                        CONTENTS                         ****

  3337            DO 500 J=2,NC
  3338            IF(DELX(J) - D(2)) 501,501,502
  3339      C                                                            70030500
  3340      C     SHARP DISCONTINUITY IN GEOMETRY                        70030510
  3341        501 S(4) = D(4)*WD(J)*(DOO(J)+RO(J)) + D(4)*RO(J)*DOO(J) +
  3342           1D(15)*RO(J)**2                                         70030530
  3343            S(5) = D(4)/D(3)*WD(J)*(DOO(J)+RO(J))**3 + D(4)/D(3)*RO(J)*  70030540
  3344           1DOO(J)**3 + D(15)/D(4)*RO(J)**4 + D(15)*RO(J)**2*DOO(J)**2  70030550
  3345            S(6) = D(4)/D(3)*DOO(J)*(WD(J)+RO(J))**3 + D(4)/D(3)*RO(J)*  70030560
  3346           1WD(J)**3 + D(15)/D(4)*RO(J)**4 + D(15)*RO(J)**2*WD(J)**2    70030570
  3347        503 UIX(J) = (S(5) + S(6))/S(4)
  3348            UIY(J) = S(5)/S(4) + DELX(J)**2/D(12)                  70030590
  3349            UIZ(J) = S(6)/S(4) + DELX(J)**2/D(12)                  70030600
  3350            GO TO 500
  3351      C                                                            70030620
  3352      C     NORMAL GEOMETRY TRANSITION                             70030630
  3353      C     USE NORMAL SEGMENT SHAPE                               70030640
  3354        502 S(1) = (DOO(J) + DOO(J-1))/D(2)
  3355            S(2) = (WD(J) + WD(J-1))/D(2)                          70030660
  3356            S(3) = (RO(J) + RO(J-1))/D(2)                          70030670
  3357            S(4) = D(4)*S(2)*(S(1)+S(3)) + D(4)*S(3)*S(1) + D(15)*S(3)**2   70030680
  3358            S(5) = D(4)/D(3)*S(2)*(S(1)+S(3))**3 + D(4)/D(3)*S(3)*S(1)**3 +  70030690
  3359           1D(15)/D(4)*S(3)**4 + D(15)*S(3)**2*S(1)**2             70030700
  3360            S(6) = D(4)/D(3)*S(1)*(S(2)+S(3))**3 + D(4)/D(3)*S(3)*S(2)**3 +  70030710
  3361           1D(15)/D(4)*S(3)**4 + D(15)*S(3)**2*S(2)**2             70030720
  3362            GO TO 503
  3363        500 CONTINUE
  3364      C                                                            70030750
  3365      C     TAIL CONE                                              70030760
  3366            J = NC                                                 70030770
  3367            S(1) = PER(J)/D(2)/D(15)                               70030780
  3368            S(2) = PI(10)/D(2)/D(15)                               70030790
  3369            S(3) = XBAR(J+1) - XO(J)                               70030800
  3370            UIX(J+1) = D(3)/D(10)*(S(1)**4 + S(1)**3*S(2) + S(1)**2*S(2)**2 + 70030810
  3371           1 S(1)*S(2)**3 + S(2)**4)/(S(1)**2 + S(1)*S(2) + S(2)**2)  70030811
  3372            UIY(J+1' = UIX(J+1)/D(2) + D(3)*D(9)/D(8)/D(10)*S(3)**2   70030820
  3373            UIZ(J+1) = UIY(J+1)                                    70030830
  3374            RETURN                                                 70030840
  3375            END
  3376      C
  3377      C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
  3378      C                        SUBROUTINE NACGEO
  3379      C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  3380      C
  3381            SUBROUTINE NACGEO
  3382      C     WRITTEN 6 APRIL 1972
  3383      C     TO DEVELOP NACELLE GEOMETRY
  3384      C
  3385             COMMON TCOM(4320)
  3386      C
  3387            DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
  3388            DIMENSION DATN(70)
  3389            DIMENSION DVN(150)
  3390            DIMENSION EQU(200)
  3391            DIMENSION WDN(10),RON(10),DON(10),BUN(10),BLN(10),BSN(10),
  3392           1 DLXN(10),SFN(10),RCUN(10),RCLN(10),RCSN(10),XBN(10)
  3393            DIMENSION UIXN'10),UIYN(10),UIZN(10)
  3394            EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
  3395           1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
  3396            EQUIVALENCE (EQU(1),D(81))
  3397            EQUIVALENCE (DATN(1),GD(501))
  3398            EQUIVALENCE (DVN(1),DV(971))
  3399            EQUIVALENCE (WDN(1),DVN(1)),(RON(1),DVN(11)),(DON(1),DVN(21)),
  3400           1 (BUN(1),DVN(31)),(BLN(1),DVN(41)),(BSN(1),DVN(51)),
  3401           2 (DLXN(1),DVN(61)),(SFN(1),DVN(71)),(RCUN(1),DVN(81)),
  3402           3 (RCLN(1),DVN(91)),(RCSN(1),DVN(101)),(XBN(1),DVN(111))
  3403            EQUIVALENCE (UIXN(1),DVN(121)),(UIYN(1),DVN(131)),
  3404           1 (UIZN(1),DVN(141))
  3405            EQUIVALENCE (ND(101),I),(ND(102),J)
  3406            EQUIVALENCE (NCN,ND(119)),(KCN,ND(120)),(ION,ND(121))
  3407      C
```

```
CARD NO      ****                          CONTENTS                          ****
  3H08       C
  3H09       C       SETUP INDICATORS AND COUNTERS
  3H10             IF(I - NCN) 10,20,20
  3H11       C       SHAPE CODE IS 2 DEVELOP PERIMETER
  3H12          10 DO 15 I=1,NCN
  3H13             IF(DATN(I+60)) 15,15,12
  3H14          12 DATN(I+60) = DATN(I+60)*(DATN(I+40) + DATN(I+50))*D(15)/D(2)
  3H15          15 CONTINUE
  3H16       C       FIT SHAPES
  3H17          20 DO 200 I=1,NCN
  3H18             IF(DATN(I+60)) 30,30,100
  3H19       C       PERIMETER IS ZERO CHECK ON GUIDE
  3H20          30 IF(DATN(I+40)) 34,34,32
  3H21       C       VERTICAL LIP
  3H22          32 ION = 1
  3H23             DATN(I+60) = D(24)
  3H24             GO TO 200
  3H25          34 IF(DATN(I+50)) 38,38,36
  3H26          36 WRITE(6,60)
  3H27          60  FORMAT (*0HO WARNING FROM NACGEO IN DATA MANAGEMENT,10X,
  3H28             12H*NACELLE LIP GEOMETRY ERROR)
  3H29       C       HORIZONTAL LIP  (UPPER L E )
  3H30          38 ION = 2
  3H31             DATN(I+60) = DATN(I+50)
  3H32             GO TO 200
  3H33       C       FULL SECTION
  3H34         100 S(1) = D(1)
  3H35             S(2) = (D(2)*(DATN(I+40) + DATN(I+50)) - DATN(I+60))/
  3H36            1(D(8) - D(2)*D(5))
  3H37             IF(S(2)) 101,101,102
  3H38         101 S(1) = DATN(I+60)/(D(2)*DATN(I+40) + D(2)*DATN(I+50))
  3H39             S(2) = D(24)
  3H40             IF(S(1) .GT. EQU(145) .AND. S(1) .LT. EQU(146)) GO TO 110
  3H41       C
  3H42       C       ERROR MESSAGE
  3H43             GO TO 108
  3H44         102 S(4) = AMAX1(DATN(I+40),DATN(I+50))
  3H45             S(5) = AMIN1(DATN(I+40),DATN(I+50))
  3H46             IF(S(5) - D(2)*S(2)) 104,110,110
  3H47         104 S(2) = S(5)/D(2)
  3H48             S(1) = DATN(I+60)/(D(2)*(D(15)*S(2) + S(4) - D(2)*S(2)))
  3H49             IF(S(1) .GT. EQU(145) .AND. S(1) .LT. EQU(146)) GO TO 110
  3H50       C
  3H51       C       ERROR MESSAGE
  3H52         108 WRITE (6,85) I, S(1)
  3H53         85  FORMAT (*0HO WARNING FROM NACGEO IN DATA MANAGEMENT  /5H  SECTION,
  3H54            1 I13,47H IS RECTANGLE OR ROUNDED RECT., CORR. FACTOR IS, 1PE 3 )
  3H55       C
  3H56         110 S(6) = (DATN(I+50) - D(2)*S(2))*S(1)/D(2)
  3H57             S(3) = (DATN(I+40) - D(2)*S(2))*S(1)/D(2)
  3H58             IF(S(6)) 111,112,112
  3H59         111 S(6) = D(24)
  3H60         112 IF(S(3)) 114,115,115
  3H61         114 S(3) = D(24)
  3H62         115 RON(I) = S(6)
  3H63             RON(I) = S(2)*S(1)
  3H64             DON(I) = S(3)
  3H65             BUN(I) = D(2)*RON(I) + D(15)/D(2)*RON(I)
  3H66             BLN(I) = BUN(I)
  3H67             BSN(I) = D(2)*DON(I) + D(15)/D(2)*RON(I)
  3H68             IF(D(2) - RON(I)) 118,118,200
  3H69         118 S(1) = RON(I)*(D(1) - D(25)/D(2))
  3H70             S(2) = RON(I)*D(25)/D(2) + DON(I)
  3H71             S(3) = RON(I)*D(25)/D(2) + RON(I)
  3H72             RCUN(I) = (S(1)**2 + S(3)**2)/D(2)/S(1)
  3H73             RCLN(I) = RCUN(I)
  3H74             RCSN(I) = (S(1)**2 + S(2)**2)/D(2)/S(1)
  3H75       C       IF R(2.0  ASSUME RADIUS OF CURVATURE IS INFINITY IE FLAT PANEL
  3H76         200 CONTINUE
  3H77             J = 0
  3H78             IF(ION) 250,250,205
```

```
 CARD NO      ****                       CONTENTS                         ****

  3479        C      CALCULATE LEADING EDGE SURFACE
  3480          805 J = 3
  3481              DLXN(1) = DATN(12) - DATN(11)
  3482              XBN(1) = DATN(11) + D(2)/D(3)*DLXN(1)
  3483              IF(ION - 1) 210,210,220
  3484          210 SFN(1) = DLXN(1)/D(2)*(BUN(2) + BLN(2))
  3485              GO TO 250
  3486          220 SFN(1) = DLXN(1)/D(2)*(DATN(81) + BUN(2) + D(2)*BSN(2))
  3487        C      CALCULATE SUBSEQUENT SECTIONS OR NOSE IF NO L.E.
  3488          250 DO 300 I=J,NCN
  3489              DLXN(I-1) = DATN(I+10) - DATN(I+9)
  3490              XBN(I-1) = DATN(I+9) + DLXN(I-1)/D(2)
  3491              SFN(I-1) = DLXN(I-1)/D(2)*(DATN(I+60) + DATN(I+59))
  3492          300 CONTINUE
  3493        C      DESCRIPTION OF OUTPUT ARRAYS
  3494        C      UIXN = UNIT ROLL INERTIA        UIYN = UNIT PITCH INERTIA
  3495        C      UIZN = UNIT YAW INERTIA
  3496              J = 2
  3497              IF(ION) 720,720,702
  3498          702 J = 3
  3499        C      CALCULATE LIP
  3500              IF(ION - 1) 704,704,710
  3501        C      VERTICAL LIP
  3502          704 S(1) = DATN(42)**2/D(4)
  3503              S(2) = DATN(52)**2/D(2)/D(12)
  3504              S(3) = DLXN(1)**2/D(2)/D(9)
  3505              UIXN(1) = S(2) + S(1)
  3506              UIYN(1) = S(3) + S(1)
  3507              UIZN(1) = S(2) + S(3)
  3508              GO TO 720
  3509        C      HORIZONTAL LIP
  3510          710 S(1) = BSN(2)*DLXN(1)
  3511              S(2) = BUN(2)*DLXN(1)
  3512              S(4) = DATN(42)**2/D(2)/D(9)*S(1)
  3513              S(5) = DATN(52)**2/D(12)*S(2)
  3514              S(6) = DATN(52)**2/D(4)*S(1)
  3515              S(7) = DATN(42'**2/D(9)*S(2)
  3516              S(8) = DLXN(1)**2/D(2)/D(9)*S(1)
  3517              S(9) = DLXN(1)**2/D(9)*S(2)
  3518              UIXN(1) = (S(4) + S(5) + S(6) + S(7))/SFN(1)
  3519              UIYN(1) = (S(4) + S(7) + S(8) + S(9))/SFN(1)
  3520              UIZN(1) = (S(5) + S(6) + S(8) + S(9))/SFN(1)
  3521        C      NORMAL GEOMETRY
  3522          720 DO 740 I=J,NCN
  3523              S(1) = (DON(I) + DON(I-1))/D(2)
  3524              S(2) = (HON(I) + HON(I-1))/D(2)
  3525              S(3) = (RON(I) + RON(I-1))/D(2)
  3526              S(4) = D(4)*S(2)*(S(1) + S(3)) + D(4)*S(3)*S(1) + D(15)*S(3)**2
  3527              S(5) = D(4)/D(3)*S(2)*(S(1) + S(3))**3 + D(4)/D(3)*S(3)*S(1)**3 +
  3528             1 D(15)/D(4)*S(3)**4 + D(15)*S(3)**2*S(1)**2
  3529              S(6) = D(4)/D(3)*S(1)*(S(2) + S(3))**3 + D(4)/D(3)*S(3)*S(2)**3 +
  3530             1 D(15)/D(4)*S(3)**4 + D(15)*S(3)**2*S(2)**2
  3531              UIXN(I-1) = (S(5) + S(6))/S(4)
  3532              UIYN(I-1) = S(5)/S(4) + DLXN(I-1)**2/D(12)
  3533              UIZN(I-1) = S(6)/S(4) + DLXN(I-1)**2/D(12)
  3534          740 CONTINUE
  3535        C      *** EXIT ***
  3536              RETURN
  3537              END
  3538        C
  3539        C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
  3540        C                          SUBROUTINE NOSGEO
  3541        C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  3542        C
  3543              SUBROUTINE  NOSGEO
  3544        C                          JULY 1970
  3545               COMMON TCOM(4320)
  3546              DIMENSION D(700),DD(700),DV(2320),S(400),ND(200)
  3547             *, DDB(80), DVB(440), BC(200), DVW(50), DVH(30)
  3548             *, DELX(20), VOL(20), S3(20), XD(20), XI(10), EQU(200)
  3549              EQUIVALENCE (TCOM(1),D(1)), (TCOM(701),DD(1)), (TCOM(1401),DV(1))
```

```
CARD NO    ****                       CONTENTS                       ****
           *, (TCOM(3721),S(1)), (TCOM(4121),ND(1))
           *, (DV(3211),DVH(1)), (DV(3711),DVW(1)), (DV(4311),DVB(1))
           *, (DVB(161),DELX(1)), (DVB(201),VOL(1)), (DVB(341),S3(1))
           *, (GD(381),GDB(1)), (GDB(8),XI(1)), (GDB(56),XO(1))
           *, (ND(115),NC)
           *, (DV(2121),BC(1)), (EQU(1),D(81))
      C BC(33)    X SUB B    DIST. FROM X REFERENCE TO BODY NOSE   INCHES
      C BC(34)  SM.L SUB N   LENGTH OF NOSE   INCHES
      C BC(35)    V SUB N    VOLUME OF NOSE  CUBIC INCHES
      C BC(38)    R SUB N    RADIUS AT BASE OF NOSE   INCHES
      C  DEFINE AFT STATION OF NOSE SECTION BY SMALLEST OF FOLLOWING 3 POINTS
      C     (1)  APEX OF WING OR HORIZONTAL WHICH EVER IS SMALLEST.
      C     (2)  1ST. FUSELAGE POINT WHERE CROSS SEC. AREA IS CONST. OR DECRES
      C     (3)  STATION WHERE DELX IS LESS THAN OR EQU TO 2.0 (START DUCTS)
            BC(33) = XI(1)
            S(3) = XI(10)
            S(1) = AMIN1(DVH(47), DVH(4))
            N=1
         70 IF(S3(N)-S3(N+1)) 75,80,80
         75 N=N+1
            IF(N.LE.NC ) GO TO 70
         80 S(2) = XO(N)
            DO 90 N=1,NC
            IF(DELX(N).GT.D(2)) GO TO 90
         85 IF(S3(N)*EQU(144) - S3(N+1)) 86,86,90
         86 S(3) = XO(N)
            GO TO 95
         90 CONTINUE
         95 S(4) = AMIN1(S(1),S(2),S(3))
            BC(34) = S(4) - XI(1)
            N=0
            S(5)=D(24)
        100 N=N+1
            IF(XO(N)-S(4))105,110,115
        105 S(5) = S(5)+VOL(N)
            GO TO 100
        110 S(5) = S(5)+VOL(N)
            S(6) = S3(N)
            GO TO 120
        115 S(6) = S3(N-1) + (S3(N)-S3(N-1))*(S(4)-XO(N-1))/(XO(N)-XO(N-1))
            S(5) = S(5) + (S(4)-XO(N-1))*((S3(N-1)+S(6))/D(2))
        120 BC(35) = S(5)
            BC(38) = (S(6)/D(15))**D(19)
            RETURN
            END
      C
      C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      C                            SUBROUTINE PRTOWE
      C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
      C
            SUBROUTINE PRTOWE
      C     WRITTEN SEPT. 1972   OPERATIONAL WEIGHT EMPTY AND EXPEND. USEFUL
      C       LOAD PRINT SUBROUTINE
      C
            COMMON TCOM(4320)
      C
            DIMENSION D(700), GD(700), DV(2320), S(400), ND(200)
            DIMENSION GDWT(180)
            DIMENSION DVWT(1000)
            DIMENSION F(40), W(20), H(10), V(10), AI(20), AO(20)
            DIMENSION T(6,40)
      C
            EQUIVALENCE (D(1),TCOM(1)), (GD(1),TCOM(701)), (DV(1),TCOM(1401)),
           1 (S(1),TCOM(3721)), (ND(1),TCOM(4121))
            EQUIVALENCE (GDWT(1),GD(91))
            EQUIVALENCE (DVWT(1),DV(1121))
            EQUIVALENCE (F(1),DVWT(101)), (W(1),DVWT(181)), (H(1),DVWT(221)),
           1 (V(1),DVWT(241)), (AI(1),DVWT(281)),(AO(1),DVWT(301))
            EQUIVALENCE (T(1,1),S(1))
      C  PRINT BREAK DOWN FROM  T(I,J) WHERE I=COMPONANT, J=LINE NO.
      C
```

```
      DO 10 N=1,240
      S(N) = D(2N)
   10 CONTINUE
      T(2,1) = M(1)
      T(3,2) = M(1)
      T(4,3) = V(1)
      T(1,4) = F(1)
      T(1,5) = F(2)
      T(2,5) = M(2)
      T(1,6) = F(3)
      T(1,7) = F(4)+F(5)
      T(2,7) = M(3)
      T(3,7) = M(2)
      T(4,7) = V(2)
      T(5,8) = AI(1)
      T(6,8) = AO(1)
      T(1,9) = F(6)
      T(5,10)= AI(2)
      T(6,10)= AO(2)
      T(5,11)= AI(3)
      T(6,11)= AO(3)
      T(5,12)= AI(4)
      T(6,12)= AO(4)
      T(5,13)= AI(5)
      T(6,13)= AO(5)
      T(5,14)= AI(6)
      T(6,14)= AO(6)
      T(5,15)= AI(7)
      T(6,15)= AO(7)
      T(5,16)= AI(8)
      T(6,16)= AO(8)
      T(1,17)= F(7)
      T(2,17)= M(4)
      T(1,18)= F(8)
      T(5,19)= AI(9)
      T(6,19)= AO(9)
      T(1,20)= F(9)
      T(5,20)= AI(10)
      T(6,20)= AO(10)
      T(1,21)= F(10)
      T(2,21)= M(5)
      T(5,21)= AI(11)
      T(6,21)= AO(11)
      T(1,22)= F(11)
      T(5,22)= AI(12)
      T(6,22)= AO(12)
      T(1,23)= F(12)
      T(5,23)= AI(13)
      T(6,23)= AO(13)
      T(1,24)= F(13)
      T(1,25)= F(14)
      T(1,26)= F(15)
      T(1,27)= F(16)
      T(5,27)= AI(14)
      T(6,27)= AO(14)
      T(1,28)= F(17)
      T(1,29)= F(18)
      T(1,30)= F(19)
      T(1,31)= F(20)
      T(1,32)= F(21)
      T(2,32)= M(6)
      T(6,33)= AI(15)
      T(6,33)= AO(15)
      T(1,34)= F(22)
      T(1,35)= F(23)
      T(1,36)= F(24)
      T(2,37)= M(7)
      T(2,38)= M(8)
      T(1,39)= F(25)
      T(1,40)= F(26)
C
```

```
C
      WRITE(6,20)
   20 FORMAT(1H1,80X,21H** PRTONE - IP(46) **/
     1        50H* DATA MANAGEMENT *** OPERATIONAL WEIGHT EMPTY ***  /
     1 18X,  38HTOTAL AND MAJOR COMPONANT BREAK DOWN // 20X,
     2 10HTOTAL WT. 6X, 3HARM 4X, 8HFUSELAGE 6X, 4HWING 2X,
     3 10HHORIZONTAL 3X, 8HVERTICAL 11H IB NACELLE  11H OB NACELLE )
      WRITE(6,30) (DWT(N),DWT(N+50), (T(I,N),I=1,8), N=1,40)
   30 FORMAT(5H WING 12X,0F11.1 /11H HORIZONTAL 6X,0F11.1 /9H VERTICAL
     1 8X,0F11.1 /5H BODY 12X,0F11.1 /10H MAIN GEAR 7X,0F11.1 /
     2 10H NOSE GEAR 7X,0F11.1 /14H SURF. CONTROL 3X,0F11.1 /
     3 13H ENG. SECTION 4X,0F11.1 /17H OTHER STRUCTURE  0F11.1 /
     4 8H ENGINES 9X,0F11.1 /17H ACCESSORY G BOX  0F11.1 /
     5 14H AIS STRUCTURE 3X,0F11.1 /17H AIS ACT AND MEC ,0F11.1 /
     6  8H EXHAUST 9X,0F11.1 / 17H COOL. AND DRNG. ,0F11.1 /
     7 12H LUBE SYSTEM 5X,0F11.1 / 12H FUEL SYSTEM 5X,0F11.1 /
     8 17H ENGINE CONTROLS  , 0F11.1 /
     9 14H STARTING SYS. 3X,0F11.1 / 6H A P U 11X,0F11.1/12H INSTRUMENTS
     A 5X,0F11.1/11H HYDRAULICS6X,0F11.1/11H ELECTRICAL 6X,0F11.1 /
     B 11H ELECTRONIC 6X,0F11.1 / 9H ARMAMENT 8X,0F11.1 /12H FURNISHINGS
     C 5X,0F11.1 /15H AIR CONDITION. 2X,0F11.1 / 7H PHOTO. 10X,0F11.1 /
     D 10H AUX. GEAR 7X,0F11.1 / 17H OTHER EQUIPMENT  0F11.1 / 5H CREW
     E 12X,0F11.1 / 11H TRAP. FUEL 6X,0F11 1 /  4H OIL 13X,0F11.1 /
     F  4H LN2 13X,0F11.1 / 14H MISCELLANEOUS 3X,0F11.1 / 5H GUNS 12X,
     G 0F11.1 / 10H W. PYLONS 7X,0F11.1 /14H W. EXT. TANKS 3X,0F11.1 /
     H 10H F. PYLONS 7X,0F11.1 /14H F. EXT. TANKS 3X,0F11.1  )
C
C .-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
C
C EXPENDABLE USEFUL LOAD
      WRITE(6,100) (ODWT(N+80),ODWT(N+90),DWT(N+840),DWT(N+40),
     * DWT(N+850),N=1,5 )
  100 FORMAT(1H1,80X,21H** PRTONE - IP(48) **//
     1        44H  DATA MANAGEMENT --  EXPENDABLE USEFUL LOAD //34X,
     124H------- CAPACITY -------, 8X,4HTOGW,8X,4HFDGW,8X,4HLDGW / 38X,
     2 6HWEIGHT,9X, 3HARM  // 13X, 21HPASSENGERS OR PAYLOAD, 5F12.2 /
     3 13X, 12HWING PAYLOAD,9X, 5F12.2 / 13X, 10HAMMUNITION, 11X,
     4 5F12.2 / 13X, 17HWING FUEL TANK  1, 4X, 5F12.2 / 13X,
     5 17HWING FUEL TANK  2, 4X, 5F12.2  )
      WRITE (6,110) ( N, ODWT(N+85),ODWT(N+95),DWT(N+845), DWT(N+45),
     * DWT(N+855), N=1,5 )
  110  FORMAT ( 13X, 18HFUSELAGE FUEL TANK, I2, X, 5F12.2 )
C
      RETURN
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                             SUBROUTINE QUIKIE
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE QUIKIE
C       WRITTEN 26 JUNE 1972
C       TO DO FIRST PASS WEIGHT ESTIMATES AND C.G.
       COMMON TCOM(4320)
      COMMON /IPRINT/ IP(80)
      DIMENSION D(700),OD(700),DV(2320),S(400),ND(200)
      DIMENSION EQU(200)
      DIMENSION OD1(20),ODD(30),ODWT(180),ODW(50),ODH(40),ODV(40),
     1 ODB(80),DATS(40),DATD(70),DATR(20),DATN(70)
      DIMENSION DVW(50),DVH(30),DVV(30),DVB(440),DVO(100),DVN(150),
     1 DWT(1000)
      DIMENSION DSP(10)
      DIMENSION XI(10)
      DIMENSION TOT(20),XBD(10:,SFD(10),XBN(10),SFN(10)
C
      EQUIVALENCE (D(1),TCOM(1)),(OD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
C
      EQUIVALENCE (EQU(1),D(81))
      EQUIVALENCE (OD1(1),OD(1)),(ODD(1),OD(21)),(ODWT(1),OD(81)),
     1 (ODW(1),OD(251)),(ODH(1),OD(301)),(ODV(1),OD(341)),
```

```
CARD NO  ****                    CONTENTS                    ****
   3783      2 (GD8(1),GD(381)),(DATS(1),GD(461)),(DATD(1),GD(501)),
   3784      3 (DATR(1),GD(57)),(DATN(1),GD(59))
   3785       EQUIVALENCE (X)(1),GD8(8))
   3786       EQUIVALENCE (DVA(1),DV(32)),(DVH(1),DV(37)),(DVV(1),DV(40)),
   3787      1 (DVB(1),DV(43)),(DVD(1),DV(87)),(DVN(1),DV(97)),
   3788      2 (DVT(1),DV(112))
   3789       EQUIVALENCE (DSP(1),DV(3))
   3790       EQUIVALENCE (TOT(1),DVB(36)),(SFD(1),DVD(7)),
   3791      1 (XED(1),DVD(8)),(SFN(1),DVN(7)),(XEN(1),DVN(11))
   3792  C
   3793       EQUIVALENCE (I,ND(101)),(J,ND(102))
   3794       EQUIVALENCE (I1,ND(107))
   3795       EQUIVALENCE (ITP,ND(111)),(IVO,ND(112)),(IOD,ND(114)),
   3796      1 (NC,ND(115)),(NCD,ND(117)),(NCN,ND(119)),(IGN,ND(121))
   3797  C      CLEAR SCRATCH REGION
   3798       DO 10 I=1,150
   3799       S(I) = D(24)
   3800    10 CONTINUE
   3801  C
   3802  C      TRANSFER INPUT DATA TO STORAGE REGION DV FILE
   3803       DO 20 I=1,40
   3804       DVAT(I) = GDAT(I)
   3805       DVAT(I+50) = GDAT(I+40)
   3806    20 CONTINUE
   3807       DO 30 I=41,50
   3808       DVAT(I) = GDAT(I+40)*GDAT(I+90)
   3809       DVAT(I+50) = GDAT(I+50)
   3810  C      TRANSFER USEFUL LOAD TO DV REGION FOR TOGW AND LDGW
   3811       DVAT(I+800) = GDAT(I+40)*GDAT(I+80)
   3812       DVAT(I+810) = GDAT(I+40)*GDAT(I+100)
   3813    30 CONTINUE
   3814  C
   3815  C      TEST FOR AIR INDUCTION SYSTEM WEIGHT
   3816       IF(GDAT(12)) 50,50,95
   3817  C      DO FIRST PASS ESTIMATE ON AIS
   3818    50 GO TO (60,70,60,60,70,70),IVG
   3819  C      DO RAMPS
   3820    60 S(51) = DATR(4)*DATR(8) + DATR(5)*DATR(9) + DATR(6)*DATR(10) +
   3821      1 DATR(7)*DATR(11)
   3822       S(52) = DATR(17) + (DATR(4) + DATR(5) + DATR(6) + DATR(7))/D(2)
   3823       S(53) = D(31)*DSP(6)
   3824       S(54) = D(32)*DSP(7)
   3825       S(55) = AMAX1(S(53),S(54))
   3826       S(61) = S(51)/D(17)*(EQU(71) + (S(55)/EQU(72))**EQU(73))*
   3827      * DATS(5)*EQU(78)
   3828       S(71) = S(52)
   3829       GO TO 80
   3830  C      SPIKES INITIAL ESTIMATE
   3831    70 IF(IVG - 5) 72,74,76
   3832    72 S(61) = EQU(29)*DATS(4)*DATS(5)/D(17)*EQU(78)
   3833       GO TO 78
   3834    74 S(61) = EQU(30)*DATS(4)*DATS(5)/D(17)*EQU(78)
   3835       GO TO 78
   3836    76 S(61) = EQU(31)*DATS(4)*DATS(5)/D(17)*EQU(78)
   3837    78 S(71) = DATS(6)
   3838  C      DUCTS INITIAL ESTIMATE
   3839    80 J = 2
   3840       IF(IGD) 85,85,81
   3841    81 J = 3
   3842       S(81) = SFD(1)/C(17)*EQU(74)*EQU(78)
   3843  C      TEST FOR MAXIMUM DESIGN PRESSURE
   3844    85 S(51) = DSP(8)*D(31)
   3845       S(52) = DSP(9)*D(31)
   3846       S(53) = DSP(10)*D(32)
   3847       S(94) = AMAX1(S(51),S(52),S(53))
   3848       S(95) = EQU(75)*EQU(78)
   3849       IF(EQU(76) - S(94)) 88,90,90
   3850    88 S(95) = (EQU(75) + ((S(94) - EQU(76))/EQU(76))**EQU(77))*EQU(78)
   3851    90 DO 92 I=J,NCD
   3852       S(I+70) = SFD(I-1)/D(17)*S(95)
   3853    92 CONTINUE
```

```
                J = NCD - 1
                DO 94 I=1,J
                S(41) = S(41) + S(I+88)
                S(42) = S(42) + S(I+88)*XSD(I)
             94 CONTINUE
                S(62) = S(41)
                S(72) = S(42)/S(41)
C
C       DO NACELLES IF APPLICABLE
             98 IF(SDMT(8)) 100,100 195
            100 IF(ITP) 150,150,102
            102 J = 2
                IF(ION) 110,110,104
            104 J = 3
            110 DO 114 I=J,NCN
                S(I+88) = SFN(I)/D(17)*EQU(79)
            114 CONTINUE
                DO 116 I=J,NCN
                S(43) = S(43) + S(I+88)
                S(44) = S(44) + S(I+88)*XSN(I )
            116 CONTINUE
                S(63) = S(43)
                S(73) = S(44)/S(43)
C       TEST ON INBD PYLON
                IF(DATS(22)) 150,150,120
            120 S(84) = DATS(22)*DATS(23)/D(17)*EQU(80)
                S(74) = DATS(12) + DATD(NCD+10) + DATS(23)/D(2)*
               1 SIN(DATS(20)*D(18))
                IF(DATS(24)) 150,150,122
C       OUTBD PYLON
            122 S(85) = DATS(24)*DATS(25)/D(17)*EQU(80)
                S(75) = DATS(12) + DATD(NCD+10) + DATS(25)/D(2)*
               1 SIN(DATS(20)*D(18))
C       ENGINE MOUNTS
            150 S(86) = SDMT(10)*EQU(81)
                S(76) = DATS(12) + DATD(NCD+10)
C       SUM ENGINE SECTION
            195 S(45) = S(63)*DATS(1) + S(84)*D(2) + S(85)*D(2) + S(86)
                IF(DATS(1) - D(2)) 160,160,165
            160 S(46) = S(63)*D(2)*(DATS(13) + S(73)) + S(84)*D(2)*
               1 (DATS(13) + S(74)) + S(86)*(DATS(13) + S(76))
                IF(DATS(1)) 162,162,164
            162 S(47) = S(61) + S(62)
                S(48) = S(61)*(DATS(13) + S(71)) + S(62)*(DATS(13) + S(72))
                GO TO 180
            164 S(47) = S(61) + D(2)*S(62)
                S(48) = S(61)*(DATS(13) + S(71)) + S(62)*D(2)*(DATS(13) + S(72))
                GO TO 180
            165 S(48) = S(63)*D(2)*(DATS(13) + DATS(16) + S(73)*D(2)) +
               1 S(84)*D(2)*(DATS(13) + S(74)) + S(85)*D(2)*(DATS(16) + S(75)) +
               2 S(86)*((DATS(13) + DATS(16))/D(2) + S(76))
                S(47) = S(61) + S(62)*DATS(1)
                S(48) = S(61)*((DATS(13) + DATS(16))/D(2) + S(71)) +
               1 S(62)*DATS(1)*((DATS(13) + DATS(16))/D(2) + S(72))
            180 IF(S(45)) 190,190,182
            182 S(107) = S(45)
                S(117) = S(46)/S(45)
            190 IF(S(47)) 200,200,192
            192 S(108) = S(47)
                S(118) = S(48)/S(47)
C
C       TEST ON WING
            200 IF(SDMT(1)) 201,201,300
            201 S(28) = AMAX1(SDD(11),SDD(12),SDD(14))
                S(21) = (SDD(3)*S(25)*D(31))*EQU(82)
                S(22) = DSP(5)*EQU(83)
                S(23) = SDM(1)*EQU(84)
                S(24) = (SDM(2)/(COS(DVM(50)*D(18)))**2))*EQU(85)
                S(25) = (SDM(12)*D(18)*D(18)/COS(DVM(50)*D(18)))*EQU(86)
C       TEST ON TAPER RATIO
                IF(SDM(3)) 222,222,224
```

```
CARD NO  ****                          CONTENTS                          ****
 3005    C       TAPER RATIO = 0.0 SET EQUAL TO 0.001
 3006      302 S(29) = D(1)/D(14)
 3007          GO TO 226
 3008      304 S(29) = GDW(3)
 3009      306 S(28) = S(29)**EQU(87)
 3010          S(27) = D(1)
 3011          IF(GDI(3)) 208,208,204
 3012    C       LANDING GEAR ON WING
 3013      204 S(27) = EQU(88)
 3014      208 S(28) = D(1)
 3015          IF(GDI(2)) 210,210,208
 3016    C       VARIABLE SWEEP WING
 3017      208 S(28) = EQU(89) - EQU(90)*GDW(31)/DWW(45)
 3018      210 S(101) = (S(21)*S(22)*S(23)*S(24)*S(26)/S(25)*S(27)*S(28))/
 3019         1 EQU(91)**EQU(92)
 3020          S(111) = DWW(43) + (EQU(93) - D(23))*DWW(44)
 3021    C
 3022    C       HORIZONTAL TAIL
 3023      300 IF(GDMT(2)) 302,302,400
 3024      302 S(30) = AMAX1(GOD(11),GOD(12),GOD(14))
 3025          S(31) = (GOD(3)*S(30)*D(3)))**EQU(94)
 3026          S(32) = DSP(5)**EQU(95)
 3027          S(33) = DWH(8)**EQU(96)
 3028          S(34) = (DWH(7)/(COS(DWH(23)*D(16)))**2))**EQU(97)
 3029          S(35) = (GDH(12)*D(10)*D(10)/COS(DWH(23)*D(16)))**EQU(98)
 3030    C       TEST ON TAPER RATIO
 3031          IF(DWH(6)) 304,304,306
 3032    C       TAPER RATIO = 0.0 SET EQUAL TO 0.001
 3033      304 S(37) = D(1)/D(14)
 3034          GO TO 308
 3035      306 S(37) = DWH(6)
 3036      308 S(38) = S(37)**EQU(99)
 3037          S(102) = (S(31)*S(32)*S(33)*S(34)/EQU(100)/S(35)/S(36))**EQU(101)
 3038          S(112) = DWH(24) + (EQU(102) - D(23))*DWH(25)
 3039    C
 3040    C       VERTICAL TAIL
 3041      400 IF(GDMT(3)) 402,402,450
 3042      402 S(30) = AMAX1(GOD(11),GOD(12),GOD(14))
 3043          S(31) = (GOD(3)*S(30)*D(3)))**EQU(103)
 3044          S(32) = DSP(5)**EQU(104)
 3045          S(33) = GDV(1)**EQU(105)
 3046          S(34) = (GDV(2)/(COS(DVV(22)*D(16)))**2))**EQU(106)
 3047          S(35) = (GDV(2)*D(10)*D(10)/COS(DVV(22)*D(16)))**EQU(107)
 3048    C       TEST ON TAPER RATIO
 3049          IF(GDV(3)) 404,404,406
 3050    C       TAPER RATIO = 0.0 SET EQUAL TO 0.001
 3051      404 S(37) = D(1)/D(14)
 3052          GO TO 408
 3053      406 S(37) = GDV(3)
 3054      408 S(38) = S(37)**EQU(108)
 3055          S(103) = (S(31)*S(32)*S(33)*S(34)*S(36)/S(35)/EQU(109))**EQU(110)*
 3056         1 DVV(23)
 3057          S(113) = DVV(24) + (EQU(111) - D(23))*DVV(25)
 3058    C       LANDING GEAR
 3059      450 IF(GDMT(5)) 460,460,452
 3060      452 IF(GDMT(6)) 460,460,500
 3061      460 S(30) = D(2)
 3062          S(31) = GOD(20)**2/D(2)/D(26)/GOD(2))*D(12)
 3063          S(32) = S(30)*GOD(1)
 3064          S(33) = S(31)*GOD(5)
 3065          IF(S(32) - S(33)) 462,470,470
 3066    C       LANDING CRITICAL
 3067      462 S(34) = S(31)**EQU(112)
 3068          S(35) = GOD(5)**EQU(113)
 3069          GO TO 480
 3070      470 S(34) = S(30)**EQU(112)
 3071          S(35) = GOD(1)**EQU(113)
 3072      480 S(38) = (GOD(23)/D(12))**EQU(114)
 3073          IF(GDMT(9)) 482,482,490
 3074      482 S(105) = EQU(115)*S(34)*S(35)*S(36)*(GOD(4) - GOD(27))/
 3075         1                                    (GOD(28) - GOD(27))
```

```
CARD NO  ****                          CONTENTS                          ****
 3976          S(115) = GDD(25)
 3977        900 IF(GDAT(6)) 902,902,900
 3978        902 S(108) = EQU(115)*S(34)*S(35)*S(38)*(GDD(26) - GDD(41))/
 3979           1                                    (GDD(26) - GDD(27))
 3980          S(110) = GDD(27)
 3981       C
 3982       C       BODY GROUP
 3983        900 IF(GDAT(9)) 902,902,600
 3984        902 S(38) = AMAX1(GDD(11),GDD(12),GDD(14))
 3985          S(31) = (TOT(11/D(17))**EQU(110)
 3986          S(32) = (S(30)*D(31)*GDD(3))**EQU(119)
 3987          S(33) = (ABS(DWA(24) - DWA(43))/D(12))**EQU(120)
 3988          S(34) = ((X(110) - X(11))/(TOT(10) + TOT(20))*D(12))**EQU(121)
 3989          S(35) = DSP(5)**EQU(122)
 3990          S(124) = S(31)*S(32)/S(33)/S(34)*S(35)*EQU(123)*EQU(124)
 3991          IF(DATS(1)) 510,510,520
 3992        510 S(114) = X(1) + (X(10) - X(1))*EQU(125)
 3993          GO TO 600
 3994        520 S(114) = X(1) + (X(10) - X(1))*EQU(126)
 3995       C
 3996       C       TEST ON WEIGHTS AND C.G. AT FLIGHT DESIGN GROSS WEIGHT
 3997        600 DO 610 I=1,40
 3998          S(1) = S(1) + DWT(I)
 3999          S(2) = S(2) + DWT(I)*DWT(I+50)
 4000        610 CONTINUE
 4001          DO 612 I=41,50
 4002          S(1) = S(1) + DWT(I)
 4003          S(2) = S(2) + DWT(I)*DWT(I+50)
 4004        612 CONTINUE
 4005          DO 614 I=1,8
 4006          S(3) = S(3) + S(I+100)
 4007        614 CONTINUE
 4008          IF(S(3) .NE. 0.0) GO TO 6149
 4009          WRITE (6,6145)
 4010       6145 FORMAT(/// 6240********** ALL DETAIL WEIGHTS AND C.G.S WERE INPUT
 4011          (********** ///)
 4012          GO TO 652
 4013       6149 CONTINUE
 4014          S(1) = (GDD(3) - S(1))/S(3)
 4015          DO 615 I=1,8
 4016          S(I+130) = S(I+100)*S(1)
 4017          S(4) = S(4) + S(I+130)
 4018          S(5) = S(5) + S(I+130)*S(I+110)
 4019        615 CONTINUE
 4020          S(12) = GDD(3)*GDD(4)
 4021       C       ASSUME CG ERROR
 4022       C       TO MATCH REQUIRED CG ERROR, SHIFT CG OF UNKNOWNS
 4023       C       SETUP 10 PERCENT SHIFT
 4024       C       ACCORDING TO CHARACTERISTIC LENGTH
 4025          S(121) = DWA(24)
 4026          S(122) = DWA(25)
 4027          S(123) = DWA(25)
 4028          S(124) = X(110) - X(1)
 4029          S(125) = GDD(23)
 4030          S(126) = GDD(24)
 4031          S(127) = DATS(10)
 4032          S(128) = DATD(NCD+10) - DATD(1)
 4033          DO 620 I=1,8
 4034          S(6) = S(6) + S(I+130)*S(I+120)
 4035        620 CONTINUE
 4036          S(13) = S(12) - S(2) - S(5)
 4037          S(14) = S(13)/S(6)
 4038          DO 622 I=1,8
 4039          IF(S(I+130)) 622,622,621
 4040        621 S(I+140) = S(I+110) + S(I+120)*S(14)
 4041        622 CONTINUE
 4042       C       CORRECT NACELLE PIECES
 4043          DO 624 I=1,8
 4044          S(I+60) = S(I+60)*S(1)
 4045        624 CONTINUE
 4046          S(71) = S(71) + S(128)*S(14)
```

CARD NO **** CONTENTS ****

```
4047              $(782) = $(782) + $(820)*$(14)
4048              $(73) = $(73) + $(127)*$(14)
4049              $(74) = $(74) + $(127)*$(14)
4050              $(75) = $(75) + $(127)*$(14)
4051              $(76) = $(76) + $(127)*$(14)
4052              DO 625 I=1,20
4053              $(I+60) = $(I+80)*$(11)
4054          625 CONTINUE
4055          C         TRANSFER DATA TO STORAGE
4056              IF(DVAT(1)) 632,632,634
4057          632 DVAT(1) = $(131)
4058              DVAT(51) = $(141)
4059          634 IF(DVAT(2)) 638,638,638
4060          638 DVAT(2) = $(132)
4061              DVAT(52) = $(142)
4062          638 IF(DVAT(3)) 642,642,642
4063          642 DVAT(3) = $(133)
4064              DVAT(53) = $(143)
4065          642 IF(DVAT(4)) 644,644,648
4066          644 DVAT(4) = $(134)
4067              DVAT(54) = $(144)
4068          648 IF(DVAT(5)) 648,648,650
4069          648 DVAT(5) = $(135)
4070              DVAT(55) = $(145)
4071          650 IF(DVAT(6)) 652,652,654
4072          652 DVAT(6) = $(136)
4073              DVAT(56) = $(146)
4074          654 IF(DVAT(8)) 656,656,658
4075          656 DVAT(8) = $(137)
4076              DVAT(58) = $(147)
4077          658 IF(DVAT(12)) 660,660,662
4078          660 DVAT(12) = $(138)
4079              DVAT(62) = $(148)
4080              WRITE(6,1620)
4081         1620 FORMAT(1HI,28X,52HSTRUCTURE WEIGHT DATA FROM QUIKIE IN DATA MANAGE
4082             *MENT//32X,18HINITIAL ESTIMATE,4X,14HCHARACTERISTIC,6X,
4083             *18HCORRECTED ESTIMATE/30X,8HWEIGHT,5X,8HHORIZ ARM,6X,8HLENGTH,
4084             * 13X,8HWEIGHT,5X,8HHORIZ ARM)
4085              WRITE(6,1621) ($(I+100),$(I+110),$(I+120),$(I+130),$(I+140),I=1,8)
4086         1621 FORMAT(8X,4HWING,11X,2F12.1,4X,F12.1,4X,2F12.1/
4087             1 8X,15HHORIZONTAL TAIL,2F12.1,4X,F12.1,4X,2F12.1/
4088             2 9X,13HVERTICAL TAIL,2X,2F12.1,4X,F12.1,4X,2F12.1/
4089             3 9X,8HFUSELAGE,7X,2F12.1,4X,F12.1,4X,2F12.1/
4090             4 8X,9HMAIN GEAR,6X,2F12.1,4X,F12.1,4X,2F12.1/
4091             5 8X,9HNOSE GEAR,6X,2F12.1,4X,F12.1,4X,2F12.1/
4092             6 8X,14HENGINE SECTION,1X,2F12.1,4X,F12.1,4X,2F12.1/
4093             7 8X,15HAIR IND. SYSTEM,2F12.1,4X,F12.1,4X,2F12.1)
4094          C
4095          682 CONTINUE
4096          C         TEST FOR 0, 2 OR 4 NACELLES
4097          C         IF 4 NACELLES, PYLONS INBOARD MAY BE DIFFERENT FROM
4098          C         OUTBOARD PYLONS
4099              II = 1
4100              $(51) = D(1)
4101              $(52) = D(24)
4102              IF(D(12) - DATS(1)) 782,770,770
4103          782 II = 2
4104              $(51) = D(18)
4105              $(52) = (DATS(13) - DATS(18))/D(12)
4106          770 $(98) = (DVAT(8) - D(2)*($(84) + $(85)))*$(51)
4107              $(88) = (DVAT(8)*DVAT(98) - D(2)*$(84)*(DATS(13) + $(74)) -
4108             1 D(2)*$(85)*(DATS(18) + $(75)))*$(51)/$(98)
4109              DVAT(281) = $(98) + D(2)*$(84)
4110              DVAT(301) = ($(98)*($(88) + $(52)) + D(2)*$(84)*
4111             1 (DATS(13) + $(74)))/DVAT(281)
4112              IF(I - II) 870,880,880
4113          870 DVAT(301) = $(98) + D(2)*$(85)
4114              DVAT(321) = ($(98)*($(88) - $(52)) + D(2)*$(85)*
4115             1 (DATS(18) + $(75)))/DVAT(301)
4116          880 CONTINUE
4117              IF(IP(49))8001,8001,8002
```

```
01/08/74      INPUT LISTING                    AUTOFLOW CHART SET - SHEEP    DATA MANAGEMENT MODULE

CARD NO       ****                          CONTENTS                          ****

 0110          8001 CONTINUE
 0111         C       BREAKPOINT OUTPUT
 0120               WRITE(6,1000)
 0121          1000 FORMAT(1H1,32X,11H   S-REGION,45X,21H** QUICKIE - IP(44) **)
 0122               DO 1004 N=1,150,5
 0123               K = N + 4
 0124               WRITE(6,1002) N,(S(J),J=N,K,1)
 0125          1002 FORMAT(4X,I4,5F10.4)
 0126          1004 CONTINUE
 0127          8002 CONTINUE
 0128               RETURN
 0129               END
 0130         C
 0131         C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
 0132         C                          SUBROUTINE SPDALT
 0133         C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
 0134         C
 0135               SUBROUTINE SPDALT                                          00020010
 0136         C     WRITTEN 20 MARCH 1972                                      00020020
 0137         C     TO DETERMINE ATMOSPHERIC PROPERTIES FOR 9 POINTS ON V-A DIAGRAM  00020030
 0138         C                                                                00020040
 0139                COMMON TCOM(4320)
 0140               COMMON /IPRINT/ IP(80)
 0141               COMMON /MISC/ XMISC(100)
 0142         C                                                                00020050
 0143               DIMENSION D(700),QD(700),DV(2320),S(400),ND(200)
 0144               DIMENSION QD1(20),QDD(30)
 0145               DIMENSION DATM(40)                                         00020080
 0146               DIMENSION EQU(200)                                         00020090
 0147               DIMENSION TITLE(16)
 0148               DIMENSION ALT(10),TEM(10),PO(10),G(10),CS(10),RHO(10)      00020110
 0149               DIMENSION VH(10),VL(10),QH(10),QL(10),EPH(10),EPL(10),     00020120
 0150              1RATH(10),RATL(10),TEMH(10),TEML(10),PTH(10),PTL(10),PSH(10),  00020130
 0151              2PSL(10)                                                   00020140
 0152               DIMENSION DVB(440),TOT(20)
 0153         C                                                                00020160
 0154               EQUIVALENCE (D(1),TCOM(1)),(QD(1),TCOM(701)),(DV(1),TCOM(1401)),
 0155              1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
 0156               EQUIVALENCE (D(81),EQU(1))                                 00020190
 0157               EQUIVALENCE (QD(1),QD(1)),(QDD(1),QD(21))
 0158               EQUIVALENCE (DATM(1),QD(5))
 0159               EQUIVALENCE (DATM(31),DVL0),(DATM(32),RATO)                00020210
 0160               EQUIVALENCE (TITLE(1),XMISC(85))
 0161               EQUIVALENCE (S(1),TEMALT),(S(2),PRESH),(S(4),ALTT)
 0162               EQUIVALENCE (ALT(1),DV(1)),(TEM(1),DV(11)),(PO(1),DV(21)),
 0163              1 (G(1),DV(31)),(CS(1),DV(41)),(RHO(1),DV(51))
 0164               EQUIVALENCE (VH(1),DV(61)),(VL(1),DV(71)),(QH(1),DV(81)),
 0165              1 (QL(1),DV(91)),(EPH(1),DV(101)),(EPL(1),DV(111)),
 0166              2 (RATH(1),DV(121)),(RATL(1),DV(131)),(TEMH(1),DV(141)),
 0167              3 (TEML(1),DV(151)),(PTH(1),DV(161)),(PTL(1),DV(171)),
 0168              4 (PSH(1),DV(181)),(PSL(1),DV(191))
 0169               EQUIVALENCE (DVB(1),DV(43)),(TOT(1),DVB(35))
 0170               EQUIVALENCE (ND(101),I),(ND(102),J)                        00020330
 0171         C                                                                00020340
 0172               DO 20 I=1,5                                                00020350
 0173              2 IF(DATM(I+10)) 4,6,10                                     00020360
 0174              4 DATM(I+10) = D(I) - DATM(I+10)                            00020370
 0175                GO TO 10                                                 00020380
 0176              6 DATM(I+10) = DVL0                                         00020390
 0177                IF(DVL0) 2,10,10                                         00020400
 0178             10 J = 2*I - 1                                              00020410
 0179                ALT(J) = DATM(I+5)                                       00020420
 0180                VH(J) = DATM(I)                                          00020430
 0181                IF(DATM(I+10) - D(1)) 12,16,16                           00020440
 0182             12 VL(J) = VH(J) + DATM(I+10)                               00020450
 0183                GO TO 20                                                 00020460
 0184             16 VL(J) = VH(J)*DATM(I+10)                                 00020470
 0185             20 CONTINUE                                                 00020480
 0186         C       INTERPOLATE FOR INTERMEDIATE ALTITUDES                   00020490
 0187                DO 40 I=1,4                                              00020500
 0188                J = I+2                                                  00020510
```

```
CARD NO  ****                          CONTENTS                          ****
 4189          ALT(J) = (ALT(J-1) + ALT(J+1))/D(2)                      00020520
 4190       40 CONTINUE                                                 00020530
 4191    C     DEVELOP ATMOSPHERIC TABLES - 9 ALTITUDES                  00020540
 4192          DO 100 I=1,9                                              00020550
 4193          ALTT = ALT(I)                                             
 4194          CALL TEMPRE                                               00020560
 4195          TEM(I) = TEMALT                                           00020570
 4196          PO(I) = PRESH                                             00020580
 4197          G(I) = D(28) - ALT(I)*EQU(21)                             00020590
 4198          RHO(I) = PO(I)/TEM(I)/EQU(22)                             00020600
 4199          CS(I) = (EQU(23)*G(I)*EQU(22)*TEM(I))**.5                 00020610
 4200       100 CONTINUE                                                 00020620
 4201    C     DETERMINE DYNAMIC PRESSURE AT INITIAL POINTS              00020630
 4202          DO 120 I=1,9                                              00020640
 4203          J = 2*I - 1                                               00020650
 4204          QH(J) = RHO(J)/G(J)/D(2)*(VH(J)*CS(J))**2                 00020660
 4205          QL(J) = RHO(J)/G(J)/D(2)*(VL(J)*CS(J))**2                 00020670
 4206       120 CONTINUE                                                 00020680
 4207    C     DETERMINE SPEED AND DYNAMIC PRESSURE AT INTERMEDIATE POINTS 00020690
 4208    C     STRAIGHT LINE INTERPOLATION ON DYNAMIC PRESSURE           00020700
 4209          DO 140 I=1,4                                              00020710
 4210          J= I*2                                                    00020720
 4211          IF(VH(J-1) - VH(J+1)) 132,130,132                         00020730
 4212       130 VH(J) = VH(J-1)                                          00020740
 4213          QH(J) = RHO(J)/G(J)/D(2)*(VH(J)*CS(J))**2                 00020750
 4214          GO TO 134                                                 00020760
 4215       132 QH(J) = (QH(J-1) + QH(J+1))/D(2)                         00020770
 4216          VH(J) = (D(2)*QH(J)*G(J)/RHO(J))** 5/CS(J)                00020780
 4217       134 IF(VL(J-1) - VL(J+1)) 138,136,138                        00020790
 4218       138 VL(J) = VL(J-1)                                          00020800
 4219          QL(J) = RHO(J)/G(J)/D(2)*(VL(J)*CS(J))**2                 00020810
 4220          GO TO 140                                                 00020820
 4221       139 QL(J) = (QL(J-1) + QL(J+1))/D(2)                         00020830
 4222          VL(J) = (D(2)*QL(J)*G(J)/RHO(J))** 5/CS(J)                00020840
 4223       140 CONTINUE                                                 00020850
 4224    C     DETERMINE PRESSURE RECOVERY AND FLOW RATE AT ENGINE FACE  00020860
 4225    C     9 POINTS                                                  00020870
 4226          DO 400 I=1,9                                              00020880
 4227          J = 2*I - 1                                               00020890
 4228          IF(DATM(I+15)) 162,162,190                                00020900
 4229       162 IF(RATO) 184,184,200                                     00020910
 4230       184 IF(VH(J) - D(1)) 186,166,168                             00020920
 4231       166 RATH(J) = D(1)                                           00020930
 4232          GO TO 170                                                 00020940
 4233       168 RATH(J) = D(1) - EQU(24)*(VH(J) - D(1))**EQU(25)         00020950
 4234       170 IF(VL(J) - D(1)) 172,172,174                             00020960
 4235       172 RATL(J) = D(1)                                           00020970
 4236          GO TO 176                                                 00020980
 4237       174 RATL(J) = D(1) - EQU(24)*(VL(J) -D(1))**EQU(25)          00020990
 4238       176 IF(J - 9) 178,300,300                                    00021000
 4239       178 IF(VH(J+1) - D(1)) 180,180,182                           00021010
 4240       180 RATH(J+1) = D(1)                                         00021020
 4241          GO TO 184                                                 00021030
 4242       182 RATH(J+1) = D(1) - EQU(24)*(VH(J+1) - D(1))**EQU(25)     00021040
 4243       184 IF(VL(J+1) - D(1)) 186,186,188                           00021050
 4244       186 RATL(J+1) = D(1)                                         00021060
 4245          GO TO 300                                                 00021070
 4246       188 RATL(J+1) = D(1) - EQU(24)*(VL(J+1) - D(1))**EQU(25)     00021080
 4247          GO TO 300                                                 00021090
 4248       190 RATH(J) = DATM(I+15)                                     00021100
 4249          RATL(J) = DATM(I+20)                                      00021110
 4250       191 IF(I - J) 192,300,300                                    00021120
 4251       192 RATH(J-1) = (RATH(J-2) + RATH(J))/D(2)                   00021130
 4252          RATL(J-1) = (RATL(J-2) + RATL(J))/D(2)                    00021140
 4253          GO TO 300                                                 00021150
 4254       200 RATH(J) = RATO                                           00021160
 4255          RATL(J) = RATO                                            00021170
 4256          GO TO 191                                                 00021180
 4257    C                                                               00021190
 4258       300 IF(DATM(I+25)) 310,310,350                               00021200
 4259       310 EXH(J) = EQU(26)                                         00021210
```

```
CARD NO      ****                          CONTENTS                             ****

 4260              EML(J) = EQU(28)                                             00021220
 4261              IF(VH(J) - D(11)) 312,312,320                                00021230
 4262          312 EMH(J) = EQU(27)                                             00021240
 4263          320 IF(VL(J) - D(11)) 322,322,330                                00021280
 4264          322 EML(J) = EQU(27)                                             00021290
 4265          330 IF(I - J) 332,332,400                                        00021320
 4266          332 EMH(J-1) = (EMH(J-2) + EMH(J))/D(2)                          00021330
 4267              EML(J-1) = (EML(J-2) + EML(J))/D(2)                          00021340
 4268              GO TO 400                                                    00021350
 4269          380 EMH(J) = DATM(I+25)                                          00021360
 4270              EML(J) = DATM(I+25)                                          00021370
 4271              GO TO 330                                                    00021380
 4272          400 CONTINUE                                                     00021390
 4273              S(1) = EQU(23)/(EQU(23) - D(1))                              00021400
 4274              S(2) = (EQU(23) - D(1))/D(2)                                 00021410
 4275         C      DETERMINE RAM TEMPERATURE, TOTAL AND STATIC PRESSURE       00021420
 4276              DO 500 I=1,9                                                 00021430
 4277              TEMH(I) = TEM(I)*(D(1) + S(2)*VH(I)**2)                      00021440
 4278              TEML(I) = TEM(I)*(D(1) + S(2)*VL(I)**2)                      00021450
 4279              PTH(I) = PO(I)*RATH(I)*(D(1) + S(2)*VH(I)**2)**S(1)/D(17)    00021460
 4280              PTL(I) = PO(I)*RATL(I)*(D(1) + S(2)*VL(I)**2)**S(1)/D(17)    00021470
 4281              PSH(I) = PTH(I)/((D(1) + S(2)*EMH(I)**2)**S(1))              00021480
 4282              PSL(I) = PTL(I)/((D(1) + S(2)*EML(I)**2)**S(1))              00021490
 4283          500 CONTINUE                                                     00021500
 4284         C       ADDED CAPABILITY FOR DETERMINING CABIN PRESSURE AND CYCLES
 4285              ALTT = DATM(39)
 4286              CALL TEMPRE
 4287              TOT(18) = (PRESH - PO(9))/D(17)
 4288              S(5) = D(1)
 4289              IF(GD1(1) - D(13)) 503,503,505
 4290          503 S(5) = D(6)
 4291          505 IF(GDD(8)) 507,507,509
 4292          507 GDD(8) = EQU(65)*S(5)
 4293          509 TOT(17) = GDD(8)
 4294              IF(IP(42))5001,5001,5002
 4295         C      BREAKPOINT OUTPUT                                          00021510
 4296         5001 WRITE(6,1111)(TITLE(N),N=1,16)                               00021530
 4297         1111 FORMAT(1H1,88X,21H** SPDALT - IP(42) **/10X,8A10/10X,8A10)
 4298              WRITE(6,80)                                                  00021550
 4299           80 FORMAT(1H0,36X,37H*** SPEED ALTITUDE PROFILE TABLES ***,     00021560
 4300            1          //46X,19HSTANDARD ATMOSPHERE//15X,8HALTITUDE,3X,    00021570
 4301            211HTEMPERATURE,5X,7HDENSITY,8X,8HPRESSURE,8X,1HG,8X,          00021580
 4302            314HSPEED OF SOUND/17X,4HFEET,5X,11HDEG RANKINE,7X,3HPCF,10X,  00021590
 4303            43HPSF,5X,9HFT/SEC SQ,7X,6HFT/SEC)                             00021600
 4304              WRITE(6,82) (ALT(I),TEM(I),RHO(I),PO(I),G(I),CS(I),I=1,9)    00021610
 4305           82 FORMAT(12X,F11.1,F12.3,5X,F10.7, F12.2,F11.3,F14.2)          00021620
 4306              WRITE(6,84)                                                  00021630
 4307           84 FORMAT(///48X,13HPROFILE TABLE//4X,                          00021640
 4308            14HALT.,3X,4HV(H),4X,4HQ(H),3X,2HMH,2X,7HPT2/PT0,2X,           00021650
 4309            25HRAM T,4X,3HPT2,4X,2HP2,4X,4HV(L),4X,4HQ(L),3X,2HML,2X,      00021660
 4310            37HPT2/PT0,2X,5HRAM T,4X,3HPT2,4X,2HP2/4X,                     00021670
 4311            44HFEET,4X,2HKN,5X,3HPSF,4X,2HKN,11X,5HDEG R,4X,3HPSI,4X,3HPSI,  00021680
 4312            54X,2HKN,5X,3HPSF,4X,2HKN,11X,5HDEG R,4X,3HPSI,4X,3HPSI)       00021690
 4313              WRITE(6,86) (ALT(I),VH(I),QH(I),EMH(I),RATH(I),TEMH(I),PTH(I),  00021700
 4314            1PSH(I),VL(I),QL(I),EML(I),RATL(I),TEML(I),PTL(I),PSL(I),I=1,9)   00021710
 4315           86 FORMAT(F10.1,F6.2,F8.2,F6.2,F7.4,F8.2,2F7.2,F7 2,F8.2,F6.2,   00021720
 4316            1F7.4,F8.2,2F7.2)                                              00021730
 4317         5002 CONTINUE
 4318              RETURN                                                       00021740
 4319              END
 4320         C
 4321         C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
 4322         C                           SUBROUTINE TEMPRE
 4323         C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
 4324         C
 4325              SUBROUTINE TEMPRE                                            00030010
 4326         C    WRITTEN 20 MARCH 1972                                        00030020
 4327         C    SUBROUTINE TO DEVELOP TEMPERATURE AND PRESSURE AS WOULD BE   00030030
 4328         C    INDICATED BY AN ALTIMETER CALIBRATED IN GEOPOTENTIAL ALTITUDE  00030040
 4329         C    FOR U.S. STANDARD ATMOSPHERE, 1962.                          00030050
 4330         C                                                                 00030060
```

```
CARD NO  ****                    CONTENTS                              ****
 0331          COMMON TCOM(9328)
 0332    C                                                              00030080
 0333          DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
 0334          DIMENSION EQU(200)                                       00030100
 0335    C                                                              00030120
 0336          EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
 0337         1 (S(1),TCOM(3721)),(ND(1),TCOM(9121))
 0338          EQUIVALENCE (D(81),EQU(1))                               00030160
 0339          EQUIVALENCE (S(1),TEMALT),(S(2),PRESH),(S(3),ALOFT),(S(4),ALTT)  00030170
 0340    C                                                              00030210
 0341          ALOFT = ALTT/D(14)                                       00030220
 0342    C       DEVELOP AMBIENT PRESSURE                               00030230
 0343          IF(ALOFT - EQU(1)) 10,30,20                              00030240
 0344       10 PRESH = EQU(2)*((D(1) - EQU(3)*ALOFT)**EQU(4))           00030250
 0345          GO TO 100                                                00030260
 0346    C       ALTITUDE GREATER THAN 36089.239 FT.                    00030270
 0347       20 IF(ALOFT - EQU(5)) 30,30,40                              00030280
 0348    C       ALTITUDE BETWEEN 36089.239 AND 65616.80 FT.            00030290
 0349       30 PRESH = EQU(7)/(EXP((ALOFT - EQU(1))/EQU(6)))            00030300
 0350          GO TO 100                                                00030310
 0351       40 IF(ALOFT - EQU(8)) 50,50,55                              00030320
 0352    C        ALTITUDE BETWEEN 65616.80 AND 104986.88 FT.           00030330
 0353       50 PRESH = EQU(9)*((D(1) + EQU(12)*(ALOFT - EQU(5)))/EQU(10))**  00030340
 0354         1EQU(11))                                                00030350
 0355          GO TO 100                                                00030360
 0356    C       ALTITUDE BETWEEN 104986.88 AND 154199.48 FT.           00030370
 0357       55 PRESH = EQU(13)*(D(1) + EQU(14)*(ALOFT - EQU(8))/EQU(15))**EQU(16)00030380
 0358          IF(ALOFT - EQU(17)) 100,100,57                           00030390
 0359       57 WRITE(6,60)                                              00030400
 0360       60 FORMAT(1H0,6X,23H*** WARNING MESSAGE ***,16X,            00030410
 0361         1HALTITUDE IS BEYOND VALID RANGE OF PRESSURE)             00030420
 0362      100 CONTINUE                                                 00030440
 0363    C                                                              00030450
 0364    C       DEVELOP AMBIENT TEMPERATURE                            00030460
 0365          IF(ALOFT - EQU(1)) 110,125,120                           00030470
 0366    C       ALTITUDE BETWEEN SEA LEVEL AND 36089.239 FT            00030480
 0367      110 TEMALT = EQU(18) - EQU(19)*ALOFT                         00030490
 0368          GO TO 630                                                00030500
 0369      120 IF(ALOFT - EQU(5)) 125,125,140                           00030510
 0370    C       ALTITUDE BETWEEN 36089.239 AND 65616.80 FT.            00030520
 0371      125 TEMALT = EQU(20)                                         00030530
 0372          GO TO 630                                                00030540
 0373      140 IF(ALOFT - EQU(8)) 145,160,160                           00030550
 0374    C       ALTITUDE BETWEEN 65616.80 AND 104986.88 FT             00030560
 0375      145 TEMALT = EQU(20) + EQU(12)*(ALOFT-EQU(5))                00030570
 0376          GO TO 630                                                00030580
 0377    C       ALTITUDE BETWEEN 104986.88 AND 154199.48 FT.           00030590
 0378      160 TEMALT = EQU(15) + EQU(14)*(ALOFT - EQU(8))              00030600
 0379          IF(ALOFT - EQU(17)) 630,630,610                          00030610
 0380      610 WRITE(6,61)                                              00030620
 0381       61 FORMAT(1H0,6X,23H*** WARNING MESSAGE ***,16 .            00030630
 0382         1HALTITUDE IS BEYOND VALID RANGE OF TEMPERATURE)          00030640
 0383      630 CONTINUE                                                 00030650
 0384          RETURN                                                   00030660
 0385          END                                                      00030670
 0386    C
 0387    C (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
 0388    C                          SUBROUTINE METOST
 0389    C )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
 0390    C
 0391          SUBROUTINE METOST
 0392    C       SUBDIVIDED FROM ORIGINAL QUICIC
 0393          COMMON TCOM(9328)
 0394          COMMON /IPRINT/ IP(80)
 0395          DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
 0396          DIMENSION EQU(200)
 0397          DIMENSION GD1(20),GDMT(100),DATS(40),DVAT(1000)
 0398          EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
 0399         * (S(1),TCOM(3721)),(ND(1),TCOM(9121))
 0400          EQUIVALENCE (GD1(1),GD(1)),(GDMT(1),GD(81)),(DATS(1),GD(141))
 0401         * , (DVAT(1),DV(112))
```

CARD NO **** CONTENTS ****

```
0402          EQUIVALENCE (EQU(1),D(81))
0403          EQUIVALENCE (I,ND(101)),(J,ND(102)),(II,ND(107))
0404    C       LOCATE OPERATIONAL WEIGHT EMPTY ITEMS ACCORDING TO STRUCTURE
0405      710 DWT(181) = DWT(1)
0406          DWT(2  = DWT(51)
0407          DWT(2  = DWT(2)
0408          DWT(231) = DWT(52)
0409          DWT(241) = DWT(3)
0410          DWT(251) = DWT(53)
0411          DWT(101) = DWT(4)
0412          DWT(141) = DWT(54)
0413          IF(OD(3)) 720,720,725
0414      720 DWT(102) = DWT(5)
0415          DWT(142) = DWT(55)
0416          GO TO 730
0417      725 DWT(182) = DWT(5)
0418          DWT(202) = DWT(55)
0419      730 DWT(103) = DWT(6)
0420          DWT(143) = DWT(56)
0421    C       DISTRIBUTE SURFACE CONTROLS BETWEEN COMPONENTS
0422    C       FIXED WING
0423          IF(DWT(7)) 752,752,731
0424      731 S(51) = EQU(133)
0425          S(52) = EQU(134)
0426          S(53) = EQU(135)
0427          S(54) = EQU(131)
0428          S(55) = EQU(132)
0429          IF(OD(2)) 732,732,734
0430      732 S(51) = EQU(128)
0431      734 IF(OD(4) - D(1)) 738,736,738
0432      736 S(52) = EQU(129)
0433      738 IF(OD(5) - D(1)) 742,740,742
0434      740 S(53) = EQU(130)
0435      742 S(56) = D(1)/(S(51) + S(52) + S(53) + S(54) + S(55))
0436          DO 750 I=1,5
0437          S(I+56) = S(I+50)*S(56)
0438      750 CONTINUE
0439          DWT(183) = DWT(7)*S(51)
0440          DWT(203) = DWT(51)
0441          DWT(222) = DWT(7)*S(52)
0442          DWT(232) = DWT(52)
0443          DWT(242) = DWT(7)*S(53)
0444          DWT(252) = DWT(53)
0445          DWT(104) = DWT(7)*S(54)
0446          DWT(144) = DWT(81) - D(10)
0447          DWT(105) = DWT(7)*S(55)
0448          S(57) = DWT(7)*DWT(57)
0449          S(58) = DWT(183)*DWT(203) + DWT(222)*DWT(232) +
0450         1 DWT(242)*DWT(252) + DWT(104)*DWT(144)
0451          DWT(145) = (S(57) - S(58))/DWT(105)
0452    C       NACELLE AND CONTENTS
0453      752 II = 1
0454          S(51) = D(1)
0455          S(52) = D(24)
0456          IF(D(2) - DATS(1)) 760,770,770
0457      760 II = 2
0458          S(51) = D(19)
0459          S(52) = (DATS(13) - DATS(18))/D(2)
0460      770 DO 780 I=1,7
0461          DWT(I+281) = DWT(I+81)*S(51)
0462          DWT(I+281) = DWT(I+58) + S(52)
0463      780 CONTINUE
0464          DWT(200) = DWT(19)*S(51)
0465          DWT(209) = DWT(69) + S(52)
0466          IF(OD(6)) 782,782,784
0467                 70) = DWT(20)*S(51)
0468                     + DWT(70) + S(52)
0469                 .0
0470                (100) = DWT(20)
0471               .( (140) = DWT(70)
0472      80 . DWT(188) = DWT(8)
```

CARD NO **** CONTENTS ****

```
      DVMT(146) = DVMT(58)
C       DISTRIBUTE FUEL SYSTEM ACCORDING TO FUEL DISTRIBUTION
      S(53) = D(24)
      DO 882 I=1,7
      S(53) = S(53) + GDMT(I+83)
  882 CONTINUE
      S(58) = GDMT(84) + GDMT(85)
      S(57) = GDMT(84)*GDMT(84) + GDMT(85)*GDMT(85)
      DVMT(107) = DVMT(17)/S(53)*(S(53) - S(58))
      DVMT(121) = DVMT(32)/S(53)*(S(53) - S(58))
      DVMT(184) = DVMT(17)/S(53)*S(58)
      DVMT(188) = DVMT(32)/S(53)*S(58)
      IF(DVMT(107)) 810,810,812
  810 DVMT(204) = DVMT(67)
      GO TO 820
  812 IF(DVMT(184)) 814,814,816
  814 DVMT(147) = DVMT(67)
      GO TO 820
  816 DVMT(204) = S(57)/S(56)
      DVMT(147) = (DVMT(17)*DVMT(67) - DVMT(184)*DVMT(204))/DVMT(107)
  820 IF(DVMT(121)) 822,822,824
  822 DVMT(206) = DVMT(82)
      GO TO 850
  824 IF(DVMT(188)) 826,826,828
  826 DVMT(161) = DVMT(82)
      GO TO 850
  828 DVMT(206) = S(57)/S(56)
      DVMT(161) = (DVMT(32)*DVMT(82) - DVMT(188)*DVMT(206))/DVMT(121)
  850 DVMT(108) = DVMT(18)
      DVMT(148) = DVMT(68)
C       DISTRIBUTE INSTRUMENTS
      DVMT(185) = DVMT(21)*EQU(137)*S(56)/S(53)
      IF(DVMT(185)) 850,850,852
  852 DVMT(205) = S(57)/S(56)
  856 DVMT(271) = DVMT(21)*EQU(137)*S(51)
      DVMT(291) = DVMT(60) + S(52)
      DVMT(110) = DVMT(21) - DVMT(185) - DVMT(21)*EQU(137)
      IF(DVMT(110)) 862,862,861
  861 DVMT(150) = (DVMT(21)*DVMT(71) - DVMT(185)*DVMT(205) -
     1 DVMT(21)*EQU(137)*DVMT(60))/DVMT(110)
C       HYDRAULICS
  862 IF(DVMT(22)) 864,864,863
  863 DVMT(272) = DVMT(22)*EQU(139)*S(51)
      DVMT(292) = DVMT(60) - DATS(12) -DATS(11)/D(2) + S(52)
      DVMT(111) = DVMT(22)*(D(1) - EQU(139))
      DVMT(151) = (DVMT(22)*DVMT(72) - DVMT(22)*EQU(139)*(DVMT(292) -
     1 S(52)))/DVMT(111)
C     ELECTRICAL
  864 IF(DVMT(23)) 866,866,865
  865 DVMT(273) = DVMT(23)*EQU(140)*S(51)
      DVMT(293) = DVMT(60) - DATS(12) - DATS(11)/D(2) + S(52)
      DVMT(112) = DVMT(23)*(D(1) - EQU(140))
      DVMT(152) = (DVMT(23)*DVMT(73) - DVMT(23)*EQU(140)*(DVMT(293) -
     1 S(52)))/DVMT(112)
C       ELECTRONICS, ARMAMENT, AND FURNISHINGS
  866 DVMT(113) = DVMT(24)
      DVMT(153) = DVMT(74)
      DVMT(114) = DVMT(25)
      DVMT(154) = DVMT(75)
      DVMT(115) = DVMT(26)
      DVMT(155) = DVMT(76)
C       AIR CONDITIONING
      IF(DVMT(27)) 868,868,867
  867 DVMT(274) = DVMT(27)*EQU(141)*S(51)
      DVMT(294) = DVMT(60) + S(52)
      DVMT(116) = DVMT(27)*(D(1) - EQU(141))
      DVMT(156) = (DVMT(27)*DVMT(77) - DVMT(27)*EQU(141)*DVMT(60))/
     1 DVMT(116)
C       PHOTOGRAPHIC, AUXILIARY GEAR, AND OTHERS, AND CREW
  868 DVMT(117) = DVMT(28)
      DVMT(157) = DVMT(78)
```

```
 CARD NO      ****                        CONTENTS                          ****

  4544                DWHT(118) = DWHT(29)
  4545                DWHT(158) = DWHT(79)
  4546                DWHT(119) = DWHT(30)
  4547                DWHT(159) = DWHT(80)
  4548                DWHT(120) = DWHT(31)
  4549                DWHT(160) = DWHT(81)
  4550          C       OIL
  4551                DWHT(275) = DWHT(33)*S(51)
  4552                DWHT(295)  = DWHT(83) + S(52)
  4553          C       LIQUID NITROGEN, MISC, GUNS
  4554                DWHT(122) = DWHT(34)
  4555                DWHT(162) = DWHT(84)
  4556                DWHT(123) = DWHT(35)
  4557                DWHT(163) = DWHT(85)
  4558                DWHT(124) = DWHT(36)
  4559                DWHT(164) = DWHT(86)
  4560          C       WING PYLONS AND TANKS
  4561                DWHT(187) = DWHT(37)
  4562                DWHT(207) = DWHT(87)
  4563                DWHT(188) = DWHT(38)
  4564                DWHT(208) = DWHT(88)
  4565          C       FUSELAGE PYLONS AND TANKS
  4566                DWHT(125) = DWHT(39)
  4567                DWHT(165) = DWHT(89)
  4568                DWHT(126) = DWHT(40)
  4569                DWHT(166) = DWHT(90)
  4570          C       TEST FOR OUTBOARD NACELLES
  4571                IF(I - II) 870,900,900
  4572            870 DO 875 I=1,19
  4573                DWHT(I+301) = DWHT(I+261)
  4574                DWHT(I+321) = DWHT(I+281) - D(2)*S(52)
  4575            875 CONTINUE
  4576            900 CONTINUE
  4577                RETURN
  4578                END
  4579          C
  4580          C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
  4581          C                          SUBROUTINE WHVGEO
  4582          C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
  4583          C
  4584                SUBROUTINE WHVGEO
  4585          C       WRITTEN 25 MAY 1972
  4586          C       TO DEVELOP VEHICLE GEOMETRY DATA IN FORM REQUIRED FOR OTHER MODU
  4587          C     LES
  4588                 COMMON TCOM(4320)
  4589                COMMON /MISC/ XMISC(100)
  4590                DIMENSION D(700),OD(700),DV(2320),S(400),ND(200)
  4591                DIMENSION EQU(200)
  4592                DIMENSION ODI(20),ODB(80),XI(10),ZI(10)
  4593                DIMENSION ODW(50),ODH(40),ODV(40),DVW(50),DVH(30),DVV(30)
  4594                DIMENSION DWHT(1000)
  4595                DIMENSION X(16),Y(16)
  4596                EQUIVALENCE (D(1),TCOM(1)),(OD(1),TCOM(701)),(DV(1),TCOM(1401)),
  4597               1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
  4598                EQUIVALENCE (EQU(1),D(81))
  4599                EQUIVALENCE (ODI(1),OD(1)),(ODB(1),OD(381))
  4600               * , (XI(1),ODB(6)),(ZI(1),ODB(16))
  4601                EQUIVALENCE (ODW(1),OD(251)),(ODH(1),OD(301)),(ODV(1),OD(341)),
  4602               1 (DVW(1),DV(321)),(DVH(1),DV(371)),(DVV(1),DV(401))
  4603                EQUIVALENCE (DWHT(1),DV(1121))
  4604                EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(II,ND(107))
  4605          C       DEVELOP REFERENCE WING DATA
  4606                S(13) = (ODW(1)*ODW(2))**.5*D(12)
  4607                S(11) = D(2)*ODW(1)*D(17)/S(13)/(D(1) + ODW(3))
  4608                S(12) = ODW(3)*S(11)
  4609                S(1) = TAN(ODW(4)*D(18))
  4610                S(2) = D(4)/ODW(2)*(D(1) - ODW(3))/(D(1) + ODW(3))
  4611                S(3) = S(1) + S(2)*ODW(5)
  4612                S(4) = S(1) + S(2)*(ODW(5) - D(1))
  4613                S(5) = S(1) + S(2)*(ODW(5) - ODW(18))
  4614                S(6) = S(1) + S(2)*(ODW(5) - D(19))
```

```
CARD NO  ****                          CONTENTS                          ****
4815            S(7) = D(2)/D(3)*(S(11) + S(12) -S(11)*S(12)/(S(11) + S(12)))
4816            S(14) = ATAN(S(3))/D(18)
4817            S(15) = ATAN(S(4))/D(18)
4818            S(16) = ATAN(S(5))/D(18)
4819            S(17) = ATAN(S(6))/D(18)
4820            S(18) = GDM(8)
4821            IF(GDM(8)) 10,10,20
4822      C       COMPUTE LEADING EDGE LOCATION  FROM MAC LOCATION
4823         10 IF(S(11) - S(12)) 12,12,14
4824         12 S(18) = GDM(9) - S(11)/D(14)
4825            GO TO 20
4826         14 S(18) = GDM(9) -S(7)/D(14) - (S(11) -S(7))/(S(11) - S(12))*
4827           1 S(13)/D(2)*S(3)
4828         20 S(19) = GDM(18)
4829            DVAR(42) = ATAN(S(6))/D(18)
4830            IF(S(11) - S(12)) 22,22,24
4831         22 DVAR(43) = S(18) + S(11)/D(14)
4832            GO TO 26
4833         24 DVAR(43) = S(18) + S(7)/D(14) + (S(11) - S(7))/(S(11) - S(12))*
4834           ( S(13)/D(2)*S(3)
4835         26 DVAR(44) = S(7)
4836            DVAR(45) = S(13)/D(2)
4837            DVAR(46) = ATAN(S(3))/D(18)
4838            DVAR(47) = S(18)
4839            DVAR(48) = S(11)
4840            DVAR(49) = S(12)
4841            S(35) = S(1) + S(2)*(GDM(5) - D(23))
4842            DVAR(50) = ATAN(S(35))/D(18)
4843      C       TEST FOR INPUT CUTS
4844            IF(GDM(18)) 35,35,40
4845      C       COMPUTE Y FOR II CUTS
4846         35 S(8) = (S(13)/D(2) - GDM(7))/D(10)
4847            S(20) = GDM(7)
4848            S(39) = S(20) + EQU(66)*S(8)*D(10)
4849            DO 38 I=1,8
4850            S(I+20) = S(I+19) + S(8)
4851         38 CONTINUE
4852            GO TO 50
4853         40 DO 48 I=1,11
4854            S(I+18) = GDM(I+18)* COS(S(18)*D(16))
4855            IF(GDM(17)) 42,42,48
4856         42 S(I+18) = GDM(I+18)
4857            IF(GDM(I+18) - D(1)) 44,44,48
4858         44 S(I+19) = GDM(7) + (S(13)/D(2) - GDM(7))*GDM(I+18)
4859         48 CONTINUE
4860      C
4861      C       TEST FOR INPUT ERROR LAST CUT OUTBOARD OF TIP
4862         50 IF(S(30) .GT. S(13)/D(2))  S(30) = S(13)/D(2)
4863      C
4864      C    TRANSFER REFERENCE WING CUTS IN S(20-30) TO DV(1881-1891)
4865            DO 52 I=1,11
4866         52 DVAR(I+880) = S(I+19)
4867      C
4868      C       TEST FOR VARIABLE SWEEP WING
4869            IF(GDI(2))100,100,200
4870      C       FIXED WING - TRANSFER DATA TO DV REGION
4871        100 DVAR(1) = GDM(7)
4872            DVAR(2) = S(14)
4873            DVAR(3) = S(16)
4874            DVAR(4) = S(18)
4875            DVAR(5) = S(11)
4876            DVAR(6) = GDM(3)
4877            DVAR(7) = GDM(2)
4878            DVAR(8) = GDM(1)
4879            DVAR(9) = S(13)/D(12)
4880            DO 110 I=1,11
4881            DVAR(I+9) = S(I+19)
4882        110 CONTINUE
4883            DVAR(21) = S(19)
4884      C       USE INPUT OR CALCULATED CUT NO 2 FOR FATIGUE
4885            DVAR(304) = DVAR(11)
```

```
CARD NO  ****                          CONTENTS                          ****
    4686          DVAR(395) = DVAR(4) + DVAR(11)*TAN(DVAR(2)*D(18)) + DVAR(21)*DVAR(5)+
    4687         1 (D(1) - (D(1) - DVAR(8))*DVAR(11)/S(13)*D(2))
    4688          DVAR(396) = DVAR(394)
    4689          DVAR(397) = DVAR(395)
    4690          GO TO 300
    4691    C
    4692    C       II=1 IS AFT WING POSITION  II=2 IS FIN POSITION
    4693    C
    4694      200 II = 1
    4695          DVAR(1) = GOM(7)
    4696    C
    4697      202 J = 20*II - 19
    4698    C
    4699          DANGLE =  YOM(II+32) - GOM(4)) / 57.2958
    4700          CDA = COS(DANGLE)
    4701          SDA = SIN(DANGLE)
    4702    C
    4703          SSPAN = S(13) / 2.0
    4704    C
    4705          X(1) = S(18)
    4706          X(2) = S(18) + S(11)
    4707          X(3) = X(1) + SSPAN * TAN(S(14)/57.2958)
    4708          X(4) = X(2) + SSPAN * TAN(S(15)/57.2958)
    4709    C
    4710          Y(1) = 0.0
    4711          Y(2) = 0.0
    4712          Y(3) = SSPAN
    4713          Y(4) = SSPAN
    4714    C
    4715          VAR = X(1) + GOM(18) * S(11)
    4716          VAR1 = TAN(S(16)/57.2958)
    4717    C
    4718          DO 3000 N=5,15
    4719          X(N) = VAR + VAR1 * S(N+15)
    4720     3000 Y(N) = S(N+15)
    4721    C
    4722          X(16) = VAR + VAR1 * SSPAN
    4723          Y(16) = SSPAN
    4724    C
    4725          DO 3010 N=1,16
    4726          XPNT = X(N)
    4727          YPNT = Y(N)
    4728          X(N) = (XPNT-GOM(32))*CDA + (YPNT-GOM(31))*SDA + GOM(32)
    4729     3010 Y(N) =-(XPNT-GOM(32))*SDA + (YPNT-GOM(31))*CDA + GOM(31)
    4730    C
    4731          XIP = X(1) - (X(1)-X(3)) * (Y(1)/(Y(1)-Y(3)))
    4732          XEP = X(2) - (X(2)-X(4)) * (Y(2)/(Y(2)-Y(4)))
    4733          XOP = X(3) + (X(1)-X(3)) * ((Y(16)-Y(3)) / (Y(1)-Y(3)))
    4734          XWP = X(4) + (X(2)-X(4)) * ((Y(16)-Y(4)) / (Y(2)-Y(4)))
    4735    C
    4736          AREA = (XEP - XIP + XWP - XOP) * Y(16) / 144.0
    4737          ROOTC = X(2) - XIP
    4738          TAPER = (XWP - XOP) / (XEP - XIP)
    4739          SPAN = Y(16) / 6.0
    4740          AR = SPAN**2 / AREA
    4741          XOC = (X(16)-XOP) / (XWP-XOP)
    4742          TANEA = ATAN((X(16)-X(5)) / (Y(16)-Y(5))) * 57.2958
    4743          TANLE = ATAN((XOP-XIP) / Y(16)) * 57.2958
    4744    C
    4745          DVAR(J+20) = XOC
    4746          DVAR(J+1) = TANLE
    4747          DVAR(J+2) = TANEA
    4748          DVAR(J+3) = XIP
    4749          DVAR(J+4) = ROOTC
    4750          DVAR(J+5) = TAPER
    4751          DVAR(J+7) = AREA
    4752          DVAR(J+8) = SPAN
    4753          DVAR(J+6) = AR
    4754    C
    4755          DO 250 I=1,11
    4756          K = J + I
```

CARD NO **** CONTENTS ****

```
0757          250 DVM(K+8) = Y(I+4)
0758        C
0759              I = 2*II
0760              DVM(I+382) = DVM(J+18)
0761              DVM(I+383) = DVM(J+3) + DVM(J+18)*TAN(DVM(J+1))*D(18)) +
0762             1 DVM(J+20)*DVM(J+4)*(D(1) - (D(1) - DVM(J+5))*DVM(J+10)/Y(18))
0763              IF(II - 1) 260,260,300
0764        C       DO FORWARD SWEEP
0765          260 II = 2
0766              GO TO 202
0767        C         VERTICAL STABILIZER
0768          300 S(61) = TAN(GDV(4)*D(18))
0769              S(62) = D(2)/GDV(2)*(D(1) - GDV(3))/(D(1) + GDV(3))
0770              S(63) = S(61) + S(62)*GDV(5)
0771              S(64) = S(61) + S(62)*(GDV(5) - D(1))
0772              S(65) = S(61) + S(62)*(GDV(5) - GDV(18))
0773              S(74) = S(61) + S(62)*(GDV(5) - D(23))
0774              S(88) = (GDV(2)*GDV(1))**.5*D(12)
0775              S(66) = D(2)*GDV(1)*D(17)/S(68)/(D(1) + GDV(3))
0776              S(67) = S(66)*GDV(3)
0777              S(69) = D(21/D(3)*(S(66) + S(67) - S(66)*S(67)/(S(66) + S(67)))
0778        C         HORIZONTAL TAIL DATA
0779              S(91) = TAN(GDH(4)*D(18))
0780              S(92) = D(4)/GDH(2)*(D(1) - GDH(3))/(D(1) + GDH(3))
0781              S(93) = S(91) + S(92)*GDH(5)
0782              S(94) = S(91) + S(92)*(GDH(5) - GDH(18))
0783              S(97) = (GDH(2)*GDH(1))**.5*D(12)
0784              S(95) = D(2)*GDH(1)*D(17)/S(97)/(D(1) + GDH(3))
0785              S(85) = GDH(3)*S(95)
0786              S(98) = D(2)/D(3)*(S(95) + S(96) - S(95)*S(98)/(S(95) + S(96)))
0787              S(99) = GDH(8)
0788              S(101) = S(91) + S(92)*(GDH(5) - D(23))
0789              IF(GDH(8)) 402,402,410
0790          402 IF(S(95) - S(96)) 404,404,406
0791          404 S(99) = GDH(9) - S(95)/D(4)
0792              GO TO 410
0793          406 S(99) = GDH(9) - S(98)/D(4) - (S(95) - S(98))/(S(95) - S(96))*
0794             1 S(97)/D(2)*S(93)
0795          410 DW(1) = D(24)
0796              S(70) = GDV(8)
0797              IF(GDV(8)) 302,302,305
0798          302 IF(S(66) - S(67)) 303,303,304
0799          303 S(70) = GDV(9) - S(66)/D(4)
0800              GO TO 305
0801          304 S(70) = GDV(9) - S(69)/D(4) - (S(66) - S(69))/(S(66) - S(67))*
0802             1 S(68)*S(63)
0803          305 DW(23) = D(1)
0804              DW(25) = S(68)
0805              IF(S(66) - S(67)) 306,306,307
0806          306 DW(24) = S(70) + S(66)/D(4)
0807              GO TO 308
0808          307 DW(24) = S(70) + S(68)/D(4) + (S(66) - S(69))/(S(66) - S(67))*
0809             1 S(68)*S(63)
0810          308 DW(27) = GDV(11) + S(68)
0811              DW(28) = S(70) + S(68)*S(63) + S(67)
0812              DW(29) = S(68)/D(12)
0813              DW(30) = S(66)
0814              IF(GDV(7)) 310,310,309
0815          309 DW(23) = D(2)
0816          310 IF(GDV(30)) 320,320,312
0817        C         VERTICAL IS ON WING OR AFT OFFUSELAGE
0818          312 DW(1) = ATAN(S(63))/D(18)
0819              DW(2) = ATAN(S(65))/D(18)
0820              DW(3) = S(70)
0821              DW(4) = S(66)
0822              DW(5) = GDV(3)
0823              DW(6) = GDV(2)
0824              DW(7) = GDV(1)
0825              DW(8) = S(88)/D(12)
0826              DW(9) = D(24)
0827              DW(28) = GDV(11)
```

```
4829            IF(GDV(18)) 313,313,316
4829          C      COMPUTE CUTS
4830           313 S(71) = S(68)/D(18)
4831               DVV(18) = D(24)
4832           314 DVV(20) = DVV(18) + EQU(68)*S(71)*D(18)
4833               DO 315 I=1,9
4834               DVV(I+18) = DVV(I+8) + S(71)
4835           315 CONTINUE
4836               GO TO 300
4837           316 DO 318 I=1,11
4838               IF(EQV(17)) 3161,3161,317
4839          3161 IF(GDV(I+18) - D(1)) 3163,3163,3165
4840          3163 DVV(I+8) = S(68)*GDV(I+18)
4841               GO TO 318
4842          3165 DVV(I+8) = GDV(I+18)
4843               GO TO 318
4844           317 DVV(I+8) = GDV(I+18)* COS(DVV(2)*D(18))
4845           318 CONTINUE
4846               GO TO 300
4847          C      VERTICAL IS ON FUSELAGE  DETERMINE EFFECTIVE SURFACE
4848           320 S(71) = S(70) + S(68)/D(2)
4849               S(72) = Z1(10)
4850               IF(S(71) - X1(10)) 322,312,312
4851          C      DETERMINE WATER PLANE OF FUSELAGE AT STATION S(71)
4852           322 I = 1
4853           324 IF(X1(I) - S(71)) 326,330,330
4854           326 I = I+1
4855               GO TO 324
4856           330 S(72) = Z1(I-1) + (Z1(I) - Z1(I-1))*(S(71) - X1(I-1))/
4857              * (X1(I) - X1(I-1))
4858          C      TEST ON HORIZONTAL TAIL
4859               IF(GD1(8)) 333,333,339
4860          C      HORIZONTAL TAIL ON FUSELAGE
4861           333 IF(S(70) - S(99) - S(95)) 334,334,339
4862           334 IF(GDH(1) - GDV(1)) 335,335,338
4863           335 S(72) = GDH(1)
4864           338 S(73) = GDV(1) - S(72)
4865               DVV(26) = S(72)
4866               DVV(8) = (S(68) + S(73))/D(12)
4867               DVV(4) = S(67) + (S(86) - S(67))*(S(68) + S(73))/S(68)
4868               DVV(1) = ATAN(S(83))/D(16)
4869               DVV(2) = ATAN(S(85))/D(16)
4870               DVV(3) = S(70) - S(73)*S(83)
4871               DVV(5) = S(67)/DVV(4)
4872               DVV(7) = (DVV(4) + S(67))/D(2)*(S(68) + S(73))/D(17)
4873               DVV(6) = DVV(8)**2/DVV(7)
4874               DVV(9) = S(73)
4875               IF(GDV(18)) 332,332,340
4876           332 DVV(10) = DVV(9)
4877               S(71) = S(68)/D(18)
4878               GO TO 314
4879           340 DO 349 I=1,11
4880               IF(GDV(17)) 341,341,347
4881           341 IF(GDV(I+18)-D(1)) 342,342,344
4882           342 DVV(I+8) = GDV(I+18)* S(68)+(GDV(1)-S(72))
4883               GO TO 349
4884           344 DVV(I+8) = GDV(I+18) + (GDV(1) - S(72))
4885               GO TO 348
4886           347 DVV(I+8) = GDV(I+18)*COS(DVV(2)*D(18)) + (GDV(1)-S(72))
4887           348 CONTINUE
4888          C      TEST FOR INPUT ERROR LAST CUT OUTBOARD OF TIP
4889           350 IF(DVV(20) .GT. DVV(8)*D(12))  DVV(20) = DVV(8)*D(12)
4890               DVV(21) = GDV(18)
4891               DVV(22) = ATAN(S(74))/D(18)
4892               IF(GDV(30)) 414,414,420
4893          C      HORIZONTAL IS ON FUSELAGE OR VERTICAL
4894           414 IF(GD1(8)) 420,420,418
4895           418 DVH(1) = GDH(1) - S(72)
4896           420 DVH(2) = ATAN(S(83))/D(16)
4897               DVH(3) = ATAN(S(84))/D(16)
4898               DVH(10) = GDH(7)
```

```
CARD NO ****                    CONTENTS                    ****
 4800          DVH(11) = GDH(7)
 4800          DVH(22) = GDH(16)
 4801          DVH(23) = ATAN(S(181))/D(16)
 4802          IF(GDH(30)) 421,421,440
 4803    C       GROSS DATA GIVEN
 4804     421  DVH(4) = S(99)
 4805          DVH(5) = S(95)
 4806          DVH(6) = GDH(3)
 4807          DVH(7) = GDH(2)
 4808          DVH(8) = GDH(1)
 4809          DVH(9) = S(97)/D(12)
 4810          IF(GDH(18)) 422,422,430
 4811    C       CALCULATE CUTS
 4812     422  S(100) = (S(97)/D(2) - GDH(7))/D(10)
 4813     423  DVH(21) = DVH(11) + EQU(67)*S(100)*D(10)
 4814          DO 424 I=1,8
 4815          DVH(I+11) = DVH(I+10) + S(100)
 4816     424  CONTINUE
 4817          GO TO 500
 4818    C       CUTS ARE INPUT
 4819     430  DO 439 I=1,11
 4820          IF(GDH(17)) 431,431,437
 4821     431  IF(GDH(I+18) - D(I)) 433,433,435
 4822     433  DVH(I+10) = GDH(I+18)*(S(97)/D(2) - GDH(7)) + GDH(7)
 4823          GO TO 439
 4824     435  DVH(I+10) = GDH(I+18)
 4825          GO TO 439
 4826     437  DVH(I+10) = GDH(I+18)*COS(DVH(3)*D(16))
 4827     439  CONTINUE
 4828          GO TO 500
 4829    C       EXPOSED GEOMETRY GIVEN COMPUTE GROSS GEOMETRY
 4830     440 DVH(5) = S(96) + (S(95) - S(96))*(S(97)/D(2) + GDH(7))/S(97)*D(2)
 4831          DVH(4) = S(99) - GDH(7)*S(93)
          DVH(6) = S(96)/DVH(5)
 4833          DVH(8) = (S(96) + DVH(5))*(S(97)/D(2) + GDH(7))/D(17)
 4834          DVH(9) = (S(97) + D(2)*GDH(7))/D(12)
 4835          DVH(7) = DVH(9)**2/DVH(8)
 4836          IF(GDH(18)) 442,442,450
 4837    C       CALCULATE CUTS
 4838     442  S(100) = S(97)/D(2)/D(10)
 4839          GO TO 423
 4840    C
 4841     450  DO 459 I=1,11
 4842          IF(GDH(17)) 451,451,457
 4843     451  IF(GDH(I+18) - D(I)) 453,453,455
 4844     453  DVH(I+10) = GDH(I+18) + S(97)/D(2) + GDH(7)
 4845          GO TO 459
 4846     455  DVH(I+10) = GDH(I+18) + GDH(7)
 4847          GO TO 459
 4848     457  DVH(I+10) = GDH(I+18)*COS(DVH(3)*D(16)) + GDH(7)
 4849     459  CONTINUE
 4850    C       TEST FOR INPUT ERROR LAST CUT OUTBOARD OF TIP
 4851      500 IF(DVH(21) .GT. DVH(9)*D(6))  DVH(21) = DVH(9)*D(6)
 4852    C       CALCULATE GROSS MAC DATA
 4853          DVH(27) = D(2)/D(3)*DVH(5)*(D(1) + DVH(6) - DVH(6)/
 4854         1 (D(1) + DVH(8)))
 4855          IF(D(1) - DVH(8)) 502,502,504
 4856     502  DVH(26) = DVH(4) + DVH(5)/D(4)
 4857          GO TO 506
          504  DVH(26) = DVH(4) + DVH(27)/D(4) + (DVH(5) - DVH(27))/
 4860         1 (D(1) - DVH(8))*DVH(9)*D(6)/DVH(5)*S(93)
 4860    C       FOR SPINDLE USE EXPOSED MAC FOR BALANCE  GDI(4) = 2
 4861    C       FOR OTHER TYPES USE GROSS DATA  GDI(4) = 0.1
 4862      508 IF(GDI(4) - D(1)) 510,510,520
 4863      510 DVH(25) = DVH(27)
 4864          DVH(24) = DVH(26)
 4865          GO TO 530
 4866      520 IF(D(1) - DVH(8)) 522,522,524
 4867      522 DVH(24) = DVH(4) + DVH(5)/D(4)
 4868          DVH(25) = DVH(5)
 4869          GO TO 530
```

```
  524 S(102) = (D(1) - DVW(8))/DVW(9)/D(8)
      S(103) = DVW(5)*(D(1) -S(102)*DVW(10))
      S(104) = DVW(5)*DVW(6)/S(103)
      DVW(25) = D(2)/D(3)*S(103)*(D(1) + S(104) - S(104)/(D(1)*S(104)))
      DVW(24) = DVW(4) + DVW(25)/D(4) + (DVW(5) - DVW(25))/S(102)/
     * DVW(5)*S(93)
C
C        **** STORE THE AR, SWEEP OF THE .25 CHORD, AND TR IN       ****
C        **** ARRAY XMISC                                           ****
C
C
C        **** WING FIXED OR AFT                                     ****
  530 XMISC(12) = DVW(7)
      VAR = D(4)*(D(1) - DVW(6))/(D(1) + DVW(6))/DVW(7)
      XMISC(13) = ATAN(TAN(DVW(2)*D(16)) - VAR*D(23))/D(16)
      XMISC(14) = DVW(6)
C
C        **** HORIZONTAL TAIL                                       ****
      XMISC(16) = DVH(7)
      XMISC(17) = DVH(23)
      XMISC(18) = DVH(6)
C
C        **** VERTICAL TAIL                                         ****
      XMISC(20) = GDV(2)*D(2)
      XMISC(21) = DVV(22)
      XMISC(22) = GDV(3)
C
      IF(GDI(21)) 999,999,540
C
C        **** STORE THE AR, SWEEP OF THE .25 CHORD, AND TR WITH THE ***
C        **** WING FORWARD IN ARRAY XMISC                           ****
C
  540 XMISC(25) = DVW(27)
      VARF = D(4)*(D(1) - DVW(26))/(D(1) + DVW(26))/DVW(27)
      XMISC(26) = ATAN   N(DVW(22)*D(16)) - VARF*D(23))/D(16)
      XMISC(27) = DVW(26)
  999 CONTINUE
      RETURN
      END
C
C ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
C                            SUBROUTINE WNGDST
C ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
C
      SUBROUTINE WNGDST
C
C       WRITTEN 19 JULY 1972
C       TO DISTRIBUTE WING AND CONTENTSAND DETERMINE INERTIAS
C
       COMMON TCOM(4320)
C
      DIMENSION D(700),GD(700),DV(2320),S(400),ND(200)
C
      DIMENSION GDI(20),GDWT(180),GDW(50)
C
      DIMENSION DVW(50),DWT(1000)
      DIMENSION YW(13),YBW(12),DYW(12),CBW(12),UX(12),UY(12),UZ(12)
C
      DIMENSION XLM(12)
C
      DIMENSION XB10(12),YB10(12),XB20(12),YB20(12)
      DIMENSION WWT(12),XBW(12),WWT1(12),XBW1(12),WWT2(12),XBW2(12),
     1 WWT3(12),XBW3(12),XB11(12),XB21(12),YB11(12),YB21(12),
     2 XB12(12),XB22(12),YB12(12),YB22(12),XB13(12),XB23(12),
     3 YB13(12),YB23(12),YY12(12),YY22(12),YY21(12),YY23(12)
C
      DIMENSION DAW(12),XCW(12),XCW1(12),XCW2(12),YCW1(12),YCW2(12)
C
      EQUIVALENCE (D(1),TCOM(1)),(GD(1),TCOM(701)),(DV(1),TCOM(1401)),
     1 (S(1),TCOM(3721)),(ND(1),TCOM(4121))
C
```

CARD NO **** CONTENTS ****

```
9041          EQUIVALENCE (DVW(1),DV(32)),(DWT(1),DV(112))
9042         1, (QD(1),QD(33)), (QDWT(1),QD(97)), (QDW(1),QD(25))
9043      C
9044          EQUIVALENCE (YW(1),DWT(381)),(YBW(1),DWT(401)),
9045         1 (DYW(1),DWT(413)),(CBW(1),DWT(425)),(UX(1),DWT(437)),
9046         2 (UY(1),DWT(449)),(UZ(1),DWT(461))
9047      C
9048          EQUIVALENCE (XLW(1),S(16))
9049      C
9050          EQUIVALENCE (WT(1),DWT(473)),(XBW(1),DWT(485)),
9051         1 (WT1(1),DWT(497)),(XBW1(1),DWT(509)),(WT2(1),DWT(521)),
9052         2 (XBW2(1),DWT(533)),(WT3(1),DWT(545)),(XBW3(1),DWT(557)),
9053         3 (XB11(1),DWT(569)),(XB21(1),DWT(581)),(YB11(1),DWT(593)),
9054         4 (YB21(1),DWT(605)),(XB12(1),DWT(617)),(XB22(1),DWT(629)),
9055         5 (YB12(1),DWT(641)),(YB22(1),DWT(653)),(XB13(1),DWT(665)),
9056         6 (XB23(1),DWT(677)),(YB13(1),DWT(689)),(YB23(1),DWT(701)),
9057         7 (YY12(1),DWT(713)),(YY22(1),DWT(725)),(YY21(1),DWT(737)),
9058         8 (YY23(1),DWT(749))
9059      C
9060          EQUIVALENCE (YB10(1),DWT(401)),(XB20(1),DWT(413)),
9061         1 (YB20(1),DWT(425)),(XB10(1),DWT(485))
9062      C
9063          EQUIVALENCE (DW(1),DWT(872)),(XDW(1),DWT(884))
9064         *, (XDW1(1),DWT(896)),(XDW2(1),DWT(908))
9065         *, (YDW1(1),DWT(920)),(YDW2(1),DWT(932))
9066      C
9067          EQUIVALENCE (I,ND(101)),(J,ND(102)),(K,ND(103)),(L,ND(104)),
9068         1 (N,ND(106)),(II,ND(107)),(JJ,ND(108))
9069      C
9070      C       DEVELOP SECTION CUTS FOR DEAD WEIGHT DISTRIBUTION AND INERTIA
9071      C         ROOT, 11 SYNTHESIS CUTS, TIP
9072          YW(1) = D(24)
9073          YW(13) = DW(45)
9074          DO 120 I=2,12
9075          YW(I) = DWT(I+859)
9076        120 CONTINUE
9077      C
9078          DO 140 I=1,12
9079          YBW(I) = (YW(I) + YW(I+1))/D(2)
9080          DYW(I) = YW(I+1) - YW(I)
9081          CBW(I) = DW(48) - (DW(48) - DW(49))*YBW(I)/DW(45)
9082          XLW(I) = DW(47) + YBW(I)*TAN(DW(46)*D(16))
9083          S(1) = DYW(I)**2
9084          S(2) = (CBW(I)*D(8)/D(10))**2
9085          S(3) = (CBW(I)*CBW(12)*D(8)/D(10))**2
9086          UX(I) = (S(1) + S(3))/D(12)
9087          UY(I) = (S(2) + S(3))/D(12)
9088          UZ(I) = (S(1) + S(2))/D(12)
9089        140 CONTINUE
9090      C
9091      C
9092      C       DISTRIBUTE WIND ACCORDING TO PARABOLIC DISTRIBUTION
9093          S(1) = DW2**WHT
9094          S(2) = DWT(181)*DWT(201)
9095          S(5) = S(1)/DW(45)**D(20)
9096          WWT(I) = S(1) - S(5)*(DW(45) - YW(2))**D(20)
9097          S(3) = WWT(I)*XLW(I)
9098          S(4) = WWT(I)*CBW(I)
9099      C
9100          DO 150 I=2,12
9101          WWT(I) = S(5)*(DW(45) - YW(I))**D(20) -
9102         1 S(5)*(DW(45) - YW(I+1))**D(20)
9103          S(3) = S(3) + WWT(I)*XLW(I)
9104          S(4) = S(4) + WWT(I)*CBW(I)
9105        150 CONTINUE
9106      C
9107          S(6) = (S(2) - S(3))/S(4)
9108          DO 154 I=1,12
9109          XBW(I) = XLW(I) + S(6)*CBW(I)
9110        154 CONTINUE
9111      C
```

```
CARD NO ****                    CONTENTS                    ****
 0112       C
 0113       C       DISTRIBUTE MISC. WING CONTENTS ACCORDING TO PARABOLIC DIST.
 0114       C         S(I+120) = FIXED WEIGHT
 0115       C         S(I+140) = X-CG
 0116       C
 0117             S(1) = D(24)
 0118             S(2) = D(24)
 0119       C
 0120             DO 100 I=3,10
 0121             S(1) = S(1) +DWAT(I+100)
 0122             S(2) = S(2) + DWAT(I+100)*DWAT(I+200)
 0123         100 CONTINUE
 0124       C
 0125             S(5) = S(1)/DWAT(45)**D(20)
 0126             S(121) = S(1) - S(5)*(DWAT(45) - YW(2))**D(20)
 0127             S(3) = S(121)*XLW(1)
 0128             S(4) = S(121)*CBW(1)
 0129       C
 0130             DO 104 I=2,12
 0131             S(I+120) = S(5)*(DWAT(45) - YW(I))**D(20) -
 0132            1 S(5)*(DWAT(45) - YW(I+1))**D(20)
 0133             S(3) = S(3) + S(I+120)*XLW(I)
 0134             S(4) = S(4) + S(I+120)*CBW(I)
 0135         104 CONTINUE
 0136       C
 0137             S(6) = (S(2) - S(3))/S(4)
 0138       C
 0139             DO 108 I=1,12
 0140             S(I+140) = XLW(I) + S(6)*CBW(I)
 0141         108 CONTINUE
 0142       C
 0143       C
 0144       C       TEST FOR LANDING GEAR ON WING
 0145             IF(DWAT(102)) 108,108,170
 0146       C       SCAN WING CUTS FOR LOCATION
 0147         170 I = 1
 0148         172 IF(YBW(I) - GD(49)) 174,178,178
 0149         174 I = I+1
 0150             IF(12 - I) 176,172,172
 0151         176 I = I-1
 0152         178 J = I-1
 0153             IF(J) 180,180,182
 0154         180 J = 1
 0155             S(J+20) = DWAT(102)
 0156             GO TO 183
 0157         182 S(I+20) = DWAT(102)*(GD(49) - YBW(J))/(YBW(I) - YBW(J))
 0158             S(J+20) = DWAT(102) - S(I+20)
 0159         183 S(14) = D(24)
 0160             S(15) = D(24)
 0161       C
 0162             DO 184 L=J,I
 0163             S(14) = S(14) + S(L+20)*XLW(L)
 0164             S(15) = S(15) + S(L+20)*CBW(L)
 0165         184 CONTINUE
 0166       C
 0167             S(16) = (DWAT(102)*DWAT(202) - S(14))/S(15)
 0168       C
 0169             DO 188 L=J,I
 0170             S(L+40) = XLW(L) + CBW(L)*S(16)
 0171             IF(S(L+120) + S(L+20)) 185,188,185
 0172         185 S(L+140) = (S(L+120)*S(L+140) +  S(L+20)*S(L+40))/
 0173            1 (S(L+120) + S(L+20))
 0174             S(L+120) = S(L+120) + S(L+20)
 0175         188 CONTINUE
 0176       C
 0177         188 CONTINUE
 0178       C
 0179       C
 0180       C       DISTRIBUTE WING PAYLOAD
 0181       C         II = GROSS WEIGHT COUNTER        JJ = PAYLOAD COUNTER
 0182             II = 1
```

```
CARD NO ****                                CONTENTS                                ****
  $183          $(1) = GOMT(122)
  $184          $(2) = GOMT(132)
  $185          $(3) = GOMT(142)
  $186          $(4) = GOMT(124)
  $187          $(5) = GOMT(134)
  $188          $(6) = GOMT(144)
  $189          $(7) = GOMT(125)
  $190          $(8) = GOMT(135)
  $191          $(9) = GOMT(145)
  $192      100 $(11) = D(1)/D(2)
  $193          K = 1
  $194          IF(GOMT(184)) 194,194,200
  $195      194 $(11) = D(1)
  $196      C
  $197      C       CLEAR SCRATCH REGION
  $198      C
  $199      200 DO 205 I=1,20
  $200          $(I+20) = D(24)
  $201          $(I+40) = D(24)
  $202      205 CONTINUE
  $203      C
  $204          IF(GOMT(82)*S(11)) 300,300,220
  $205      C       DISTRIBUTE WING PAY LOAD - TEST ON INBOARD OR OUTBOARD PAYLOAD
  $206      220 S(12) = GOMT(82)*S(11)*S(11)
  $207          S(13) = GOMT(K+102)
  $208          S(17) = GOMT(92)
  $209          IF(S(11) - D(11)) 222,251,251
  $210      222 S(19) = ((GOMT(104) - GOMT(103))*TAN(DVAR(46)*D(16)))/D(2)
  $211          S(17) = GOMT(92) - S(19)
  $212          S(18) = GOMT(92) + S(19)
  $213      C       SCAN WING CUTS FOR LOCATION
  $214      251 I = 1
  $215      252 IF(YBW(I) - S(13)) 254,258,258
  $216      254 I = I + 1
  $217          IF(I2 - I)256,252,252
  $218      256 I = I - 1
  $219      258 J = I - 1
  $220          IF(J) 260,260,262
  $221      260 J = 1
  $222          S(J+60) = S(12)
  $223          GO TO 270
  $224      262 S(I+60) = S(12)*(S(13) - YBW(J))/(YBW(I) - YBW(J))
  $225          S(J+80) = S(12) - S(I+60)
  $226      270 S(14) = D(24)
  $227          S(15) = D(24)
  $228      C
  $229          DO 272 L=J,I
  $230          S(14) = S(14) + S(L+80)*XLW(L)
  $231          S(15) = S(15) + S(L+80)*CBW(L)
  $232      272 CONTINUE
  $233      C
  $234          S(16) = (S(12)*S(K+16) - S(14))/S(15)
  $235      C
  $236          DO 278 L = J,I
  $237          S(L+80) = XLW(L) + CBW(L)*S(16)
  $238          IF(S(L+20) + S(L+60)) 274,278,274
  $239      274 S(19) = (S(L+20)*S(L+40) + S(L+80)*S(L+80))/(S(L+20) + S(L+80))
  $240          S(L+20) = S(L+20) + S(L+80)
  $241          S(L+40) = S(19)
  $242      278 CONTINUE
  $243      C
  $244      C
  $245      C
  $246          IF(K - 1) 280,880,300
  $247      280 IF(S(11) - D(1)) 282,300,300
  $248      282 K = K + 1
  $249          GO TO 220
  $250      C       DISTRIBUTE WING FUEL  K = 1 INBD   K = 2 OUTBD
  $251      300 K = 1
  $252      302 JJ = II + K*3
  $253          IF(GOMT(K+82)*S(JJ)) 400,400,320
```

```
CARD NO ****                                CONTENTS                                ****
      300 S(11) = GDWT(K+83)*S(JJ)
          I = 24
          S(12) = GDWT(I+103)
          S(13) = GDWT(I+104)
          IF(YW(13) - S(12)) 310,314,314
      310 WRITE(6,312)
      312 FORMAT(1H0,20X,39HERROR INBD FUEL RIB IS OUTBD OF TIP)
          I = 12
          J = 13
          GO TO 340
      314 I = 1
      322 IF(YW(I) - S(12)) 324,330,326
      324 I = I + 1
          GO TO 322
      326 IF(YBW(I-1) .GT. S(12))  I = I - 1
      330 J = 1
      332 IF(YW(J) - S(13)) 334,340,336
      334 J = J + 1
          IF(13 - J) 338,332,332
      336 IF(S(13) - YBW(J-1)) 338,340,340
      338 J = J - 1
      340 N = J - 1
          S(14) = D(24)
          DO 342 L=1,N
C         DISTRIBUTE FUEL ACCORDING TO WING VOLUME
          S(L+100) =  DYW(L)*CBW(L)*(D(I) - (D(I) - GDW(3)*GDW(13))*
     1 YBW(L)/DWW(45))
          S(14) = S(14) + S(L+100)
      342 CONTINUE
C
          DO 344 L=1,N
          S(L+60) = S(11)*S(L+100)/S(14)
      344 CONTINUE
C
          S(15) = D(24)
          S(16) = D(24)
C
          DO 350 L=1,N
          S(15) = S(15) + S(L+60)*XLW(L)
          S(16) = S(16) + S(L+60)*CBW(L)
      350 CONTINUE
C
          S(17) = (S(11)*GDWT(K+83) - S(15))/S(16)
          DO 354 L=1,N
          S(L+80) = XLW(L) + CBW(L)*S(17)
          IF(S(L+20) + S(L+60)) 352,354,352
      352 S(19) = (S(L+20)*S(L+40) + S(L+60)*S(L+80))/(S(L+20) + S(L+60))
          S(L+40) = S(19)
          S(L+20) = S(L+20) + S(L+60)
      354 CONTINUE
      400 IF(K - 1) 402,402,450
      402 K = K + 1
          GO TO 302
C
C
C         SETUP DEAD WEIGHT DISTRIBUTION ARRAYS
      450 GO TO(460,470,480),II
C
C         MAXIMUM DESIGN WEIGHT ARRAY
      460 II = 2
C
          DO 465 I=1,12
          WWT(I) = WWT(I) + S(I+120) + S(I+20)
          IF(WWT(I)) 462,465,462
      462 XBW(I) = (WWT(I)*XBW(I) + S(I+120)*S(I+140) + S(I+20)*S(I+40))/
     1 WWT(I)
      465 CONTINUE
C
          GO TO 190
C
C         BASIC FLIGHT DESIGN WEIGHT ARRAY
```

```
CARD NO  ****                          CONTENTS                          ****
   9325          470 II = 3
   9326       C
   9327             DO 475 I=1,12
   9328             WWT2(I) = WWT(I) + S(I+120) + S(I+20)
   9329             IF(WWT2(I)) 472,475,472
   9330         472 XBW2(I) = (WWT(I)*XBW(I) + S(I+120)*S(I+140) + S(I+20)*S(I+40))/
   9331            1 WWT2(I)
   9332         475 CONTINUE
   9333       C
   9334             GO TO 190
   9335       C
   9336       C        LANDING DESIGN WEIGHT ARRAY
   9337       C
   9338         480 DO 485 I=1,12
   9339             WWT3(I) = WWT(I) + S(I+120) + S(I+20)
   9340             IF(WWT3(I)) 482,485,482
   9341         482 XBW3(I) = (WWT(I)*XBW(I) + S(I+120)*S(I+140) + S(I+20)*S(I+40))/
   9342            1 WWT3(I)
   9343         485 CONTINUE
   9344       C
   9345       C        MINIMUM FLYING WEIGHT ARRAY
   9346       C
   9347             DO 495 I=1,12
   9348             DWW(I) = WWT(I) + S(I+120)
   9349             IF(DWW(I)) 492,495,492
   9350         492 XDW(I) = (WWT(I)*XBW(I) + S(I+120)*S(I+140))/DWW(I)
   9351         495 CONTINUE
   9352       C
   9353       C
   9354       C        TRANSFER DATA TO WORKING REGION ROTATE WING IF SWING WING
   9355             DO 650 I=1,12
   9356             IF(GO(I+2)) 602,602,610
   9357       C        FIXED WING OR SEGMENT INBOARD OF PIVOT
   9358         602 XB(1,I) = XBW(I)
   9359             XB2(I) = XBW(I)
   9360             YB(1,I) = YBW(I)
   9361             YB2(I) = YBW(I)
   9362             XB12(I) = XBW2(I)
   9363             XB22(I) = XBW2(I)
   9364             YB12(I) = YBW(I)
   9365             YB22(I) = YBW(I)
   9366             XB13(I) = XBW3(I)
   9367             XB23(I) = XBW3(I)
   9368             YB13(I) = YBW(I)
   9369             YB23(I) = YBW(I)
   9370             YY12(I) = WWT2(I)*UY(I)
   9371             YY22(I) = WWT2(I)*UY(I)
   9372             YY2(I) = WWT(I)*UY(I)
   9373             YY23(I) = WWT3(I)*UY(I)
   9374             XDW1(I) = XDW(I)
   9375             XDW2(I) = XDW(I)
   9376             YDW1(I) = YBW(I)
   9377             YDW2(I) = YBW(I)
   9378             GO TO 650
   9379       C
   9380       C        VARIABLE SWEEP WING
   9381         610 IF(YBW(I) - GDW(3)) 602,602,612
   9382       C        SEGMENT OUTBOARD OF PIVOT
   9383         612 S(1) = ATAN((XBW(I) - GDW(32))/(YBW(I) - GDW(3)))
   9384             S(2) = ((XBW(I) - GDW(32))**2 + (YBW(I) - GDW(3))**2)**.5
   9385             S(3) = GDW(33) - GDW(4)
   9386             S(4) = GDW(34) - GDW(4)
   9387             XB1(I) = GDW(32) + S(2)*SIN(S(1) + S(3)*D(18))
   9388             YB1(I) = GDW(31) + S(2)*COS(S(1) + S(3)*D(18))
   9389             XB21(I) = GDW(32) + S(2)*SIN(S(1) + S(4)*D(18))
   9390             YB21(I) = GDW(31) + S(2)*COS(S(1) + S(4)*D(18))
   9391             S(1) = ATAN((XBW2(I) - GDW(32))/(YBW(I) - GDW(3)))
   9392             S(2) = ((XBW2(I) - GDW(32))**2 + (YBW(I) - GDW(3))**2)**.5
   9393             XB12(I) = GDW(32) + S(2)*SIN(S(1) + S(3)*D(18))
   9394             YB12(I) = GDW(31) + S(2)*COS(S(1) + S(3)*D(18))
   9395             XB22(I) = GDW(32) + S(2)*SIN(S(1) + S(4)*D(18))
```

```
CARD NO  ****                    CONTENTS                    ****
9388          YBB2(I) = GDM(31) + S(2)*COS(S(I) + S(4)*D(18))
9387          S(I) = ATAN((XBW3(I) - GDM(32))/(YBW(I) - GDM(31)))
9388          S(2) = ((XBW3(I) - GDM(32))**2 + (YBW(I) - GDM(31))**2)**.5
9389          XB13(I) = GDM(32) + S(2)*SIN(S(I) + S(3)*D(18))
9400          YB13(I) = GDM(31) + S(2)*COS(S(I) + S(3)*D(18))
9401          XB23(I) = GDM(32) + S(2)*SIN(S(I) + S(4)*D(18))
9402          YB23(I) = GDM(31) + S(2)*COS(S(I) + S(4)*D(18))
9403          S(I) = ATAN((XBM(I) - GDM(32))/(YBM(I) - GDM(31)))
9404          S(2) = ((XBM(I) - GDM(32))**2 + (YBM(I) - GDM(31))**2)**.5
9405          XBM1(I) = GDM(32) + S(2)*SIN(S(I) + S(3)*D(18))
9406          YBM1(I) = GDM(31) + S(2)*COS(S(I) + S(3)*D(18))
9407          XBM2(I) = GDM(32) + S(2)*SIN(S(I) + S(4)*D(18))
9408          YBM2(I) = GDM(31) + S(2)*COS(S(I) + S(4)*D(18))
9409          YY12(I) = WHT2(I)*(UY(I)*COS(S(3)*D(18))**2 +
9410         1 UX(I)*SIN(S(3)*D(18))**2)
9411          YY22(I) = WHT2(I)*(UY(I)*COS(S(4)*D(18))**2 +
9412         1 UX(I)*SIN(S(4)*D(18))**2)
9413          YY21(I) = WHT1(I)*(UY(I)*COS(S(4)*D(18))**2 +
9414         1 UX(I)*SIN(S(4)*D(18))**2)
9415          YY23(I) = WHT3(I)*(UY(I)*COS(S(4)*D(18))**2 +
9416         1 UX(I)*SIN(S(4)*D(18))**2)
9417       850 CONTINUE
9418       C
9419       C       NOTE  YB10 AND YBW, AND XB10 AND XBW ARE COMMON - SEE EQUIVAL
9420       C       TRANSFER WING WEIGHT DATA - ROTATE IF SWING WING
9421            DO 750 I=1,12
9422            IF(QD1(2)) 710,710,720
9423       C
9424       C      FIXED WING
9425         710 XB20(I) = XB10(I)
9426            YB20(I) = YB10(I)
9427            GO TO 750
9428       C
9429       C      SWING WING
9430         720 IF(YBM(I) - GDM(31)) 710,710,730
9431       C      WEIGHT OUTBOARD OF PIVOT
9432         730 S(I) = ATAN((XBM(I) - GDM(32))/(YBM(I) - GDM(31)))
9433            S(2) = ((XBM(I) - GDM(32))**2 + (YBM(I) - GDM(31))**2)**.5
9434            S(3) = GDM(33) - GDM(4)
9435            S(4) = GDM(34) - GDM(4)
9436            XB10(I) = GDM(32) + S(2)*SIN(S(I) + S(3)*D(18))
9437            YB10(I) = GDM(31) + S(2)*COS(S(I) + S(3)*D(18))
9438            XB20(I) = GDM(32) + S(2)*SIN(S(I) + S(4)*D(18))
9439            YB20(I) = GDM(31) + S(2)*COS(S(I) + S(4)*D(18))
9440         750 CONTINUE
9441       C
9442            RETURN
9443            END
```